# CONDITIONAL GRADIENT SLIDING FOR CONVEX OPTIMIZATION [*]

GUANGHUI LAN [†] AND YI ZHOU [‡]

**Abstract.** In this paper, we present a new conditional gradient type method for convex optimization by calling a linear optimization (LO) oracle to minimize a series of linear functions over the feasible set. Different from the classic conditional gradient method, the conditional gradient sliding (CGS) algorithm developed herein can skip the computation of gradients from time to time, and as a result, can achieve the optimal complexity bounds in terms of not only the number of calls to the LO oracle, but also the number of gradient evaluations. More specifically, we show that the CGS method requires $\mathcal{O}(1/\sqrt{\epsilon})$ and $\mathcal{O}(\log(1/\epsilon))$ gradient evaluations, respectively, for solving smooth and strongly convex problems, while still maintaining the optimal $\mathcal{O}(1/\epsilon)$ bound on the number of calls to LO oracle. We also develop variants of the CGS method which can achieve the optimal complexity bounds for solving stochastic optimization problems and an important class of saddle point optimization problems. To the best of our knowledge, this is the first time that these types of projection-free optimal first-order methods have been developed in the literature. Some preliminary numerical results have also been provided to demonstrate the advantages of the CGS method.

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

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

## 1. Introduction.

The conditional gradient (CndG) method, which was initially developed by Frank and Wolfe in 1956 [13] (see also [11, 12]), has been considered one of the earliest first-order methods for solving general convex programming (CP) problems. Consider the basic 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 smooth convex function such that

$$\|f'(x) - f'(y)\|_* \le L\|x - y\|, \forall x, y \in X. \tag{1.2}$$

The CndG method solves (1.1) iteratively by minimizing a series of linear approximations of $f$ over the feasible set $X$. More specifically, given $x_{k-1} \in X$ at the $k$-th iteration, it updates $x_k$ according to the following steps.

1) Call the first-order (FO) oracle to compute $(f(x_{k-1}), f'(x_{k-1}))$ and set $p_k = f'(x_{k-1})$.

2) Call the linear optimization (LO) oracle to compute

$$y_k \in \text{Argmin}_{x \in X} \langle p_k, x \rangle. \tag{1.3}$$

3) Set $x_k = (1 - \alpha_k)x_{k-1} + \alpha_k y_k$ for some $\alpha_k \in [0, 1]$.

In addition to the computation of first-order information, each iteration of the CndG method requires only the solution of a linear optimization subproblem (1.3), while most other first-order methods require the projection over $X$. Since in some cases it is computationally cheaper to solve (1.3) than to perform projection over $X$, the CndG method has gained much interests recently from both the machine learning and optimization community (see, e.g.,[1, 2, 3, 10, 8, 14, 21, 20, 22, 23, 24, 29, 34, 36]). In particular, much recent research effort has been devoted to the complexity analysis of the CndG method. For example, it has been shown that if $\alpha_k$ in step 3) of the CndG method are properly chosen, then this algorithm can find an $\epsilon$-solution of (1.1) (i.e., a point $\bar{x} \in X$ s.t. $f(\bar{x}) - f^* \le \epsilon$) in at most $\mathcal{O}(1/\epsilon)$ iterations. In fact, such a complexity result has been established for the CndG method under a stronger termination criterion based on the first-order optimality condition of (1.1) and referred to as the Wolfe Gap (see [23, 24, 14, 20]).

Observe that the aforementioned $\mathcal{O}(1/\epsilon)$ bound on gradient evaluations is significantly worse than the optimal $\mathcal{O}(1/\sqrt{\epsilon})$ bound for smooth convex optimization [30, 32]. Hence, a natural question is whether one can further improve the $\mathcal{O}(1/\epsilon)$ complexity bound associated with the CndG method. The research results along this direction, however, are mostly pessimistic. For example, by generalizing an interesting observation

[*] The author of this paper was partially supported by NSF grant CMMI-1000347, DMS-1319050, 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).

[‡] Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 32611. (email: yizhou@ufl.edu).

made by Jaggi in [23], Lan [27] considered a general class of linear-optimization-based convex programming (LCP) methods which consist of the following steps.

1) Define the linear function $\langle p_k, \cdot \rangle$.
2) Call the LO oracle to compute $y_k \in \text{Argmin}_{x \in X} \langle p_k, x \rangle$.
3) Output $x_k \in \text{Conv}\{y_0, \ldots, y_k\}$.

The LCP methods cover the CndG algorithm and also a few of its variants in [27] as certain special cases. It is shown in [23, 27] that total number of iterations for the LCP methods cannot be smaller than $\mathcal{O}(1/\epsilon)$, even if the objective function $f$ is strongly convex. By generalizing the black-box oracle complexity of large-scale smooth convex optimization in [30], Guzman and Nemirovski [19] showed that the aforementioned $\mathcal{O}(1/\epsilon)$ bound is tight (up to a logarithmic in the design dimension) for some particular classes of problems, e.g., $X$ is an $l_\infty$ ball. Improved complexity results can only be obtained under stronger assumptions on the LO oracle or the feasible set (see, e.g., [15, 27]).

Our main goal in this paper is to show that, although the number of calls to the LO oracle cannot be improved for the LCP methods in general, we can substantially improve their complexity bounds in terms of the number of gradient evaluations. To this end, we present a new LCP algorithm, referred to as the conditional gradient sliding (CGS) method, which can skip the computation for the gradient of $f$ from time to time while still maintaining the optimal bound on the number of calls to the LO oracle. Our development has been leveraged on the basic idea of applying the CndG method to the subproblems of Nesterov's accelerated gradient method [31, 32], rather than to the original CP problem in (1.1) itself. As a result, the same first-order information of $f$ will be used throughout a large number of CndG iterations. Moreover, the accuracy of the approximate solutions to these subproblems is measured by the aforementioned Wolfe gap, which allows us to establish the convergence of an inexact version of Nesterov's accelerated gradient method. It should be noted that several previous studies (e.g., [35, 38]) had also considered inexact solutions of these subproblems. These studies focused on the rate of convergence of the inexact accelerated gradient method, rather than the computation of the solutions to these subproblems. As a consequence, they did not reveal how to achieve the overall optimal complexity in terms of both the computation of first-order information and the computation of the solutions to these subproblems. On the other hand, our work has been inspired by the gradient sliding method recently developed by Lan [28] for solving a class of composite problems whose objective function is given by the summation of a general smooth and nonsmooth convex function. It is shown in [28] that one can obtain optimal complexity bounds on the number of gradient evaluations for the smooth component and subgradient evaluations for the nonsmooth component separately. Note however that this algorithm does not apply to our problem setting as it requires exact solutions to certain projection-type subproblems.

Our main theoretical contributions are briefly summarized as follows. Firstly, we show that if $f$ is a smooth function satisfying (1.2), then the number of calls to the FO and LO oracles, respectively, can be bounded by $\mathcal{O}(1/\sqrt{\epsilon})$ and $\mathcal{O}(1/\epsilon)$. Moreover, if $f$ is smooth and strongly convex, then the number of calls to the FO oracle can be significantly reduced to $\mathcal{O}(\log 1/\epsilon)$ while the number of calls to the LO oracle remains the same. It should be noted that these improved complexity bounds were obtained without enforcing any stronger assumptions on the LO oracle or the feasible set $X$.

Secondly, we consider the stochastic case where one can only have access to a stochastic first-order oracle (SFO) of $f$, which upon requests, return unbiased estimators for the gradient of $f$. By developing a stochastic counterpart of the CGS method, i.e., the SCGS algorithm, we show that the number of calls to the SFO and LO oracles, respectively, can be optimally bounded by $\mathcal{O}(1/\epsilon^2)$ and $\mathcal{O}(1/\epsilon)$ when $f$ is smooth. In addition, if $f$ is smooth and strongly convex, then the former bound can be significantly reduced to $\mathcal{O}(1/\epsilon)$.

Thirdly, we generalize the CGS and SCGS algorithms to solve an important class of nonsmooth CP problems that can be closely approximated by a class of smooth functions. By incorporating an adaptive smoothing technique into the conditional gradient sliding algorithms, we show that the number of gradient evaluations and calls to the LO oracle can bounded optimally by $\mathcal{O}(1/\epsilon)$ and $\mathcal{O}(1/\epsilon^2)$, respectively.

To the best of our knowledge, all these theoretical developments seem to be new in the literature. Some promising numerical results have also been provided to demonstrate the advantages of the CGS algorithm over the classic CndG method applied directly to problem (1.1).

This paper is organized as follows. In Section 2 we present the basic scheme for the CGS method, and establish its general convergence properties to solve problem (1.1). Moreover, we develop a variant of CGS to solve strongly convex problems in this section. Section 3 is devoted to the stochastic conditional gradient sliding algorithm for solving a class of stochastic programming problems and its variant to solve strongly convex stochastic problems. In Section 4, we generalize the CGS algorithm for solving a special class of nonsmooth CP problems possessing a saddle point structure. Finally, we present some promising numerical results for the basic CGS algorithms in Section 5.

**1.1. Notation and terminology.** Let $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^m$ be given convex compact sets. Also let $\|\cdot\|_X$ and $\|\cdot\|_Y$ be the norms associated with inner product in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively (see Remark 1 for more discussions). 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

$$D_X \equiv D_{X,\|\cdot\|} := \max_{x,y\in X} \|x-y\| \tag{1.4}$$

and

$$D_Y \equiv D_{Y,\|\cdot\|} := \max_{x,y\in Y} \|x-y\|. \tag{1.5}$$

For a given norm $\|\cdot\|$, we denote its conjugate by $\|s\|_* = \max_{\|x\|\le 1}\langle s, x\rangle$. 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

$$l_f(x;y) := f(x) + \langle f'(x), y-x\rangle. \tag{1.6}$$

Clearly, if $f$ satisfies (1.2), then

$$f(y) \le l_f(x;y) + \frac{L}{2}\|y-x\|^2, \quad \forall\, x,y \in X. \tag{1.7}$$

Notice that the constant $L$ in (1.2) and (1.7) depends on $\|\cdot\|$.

## 2. The conditional gradient sliding method.

Our goal in this section is to present a new LCP method, namely the conditional gradient sliding (CGS) method, which can skip the computation for the gradient of $f$ from time to time when performing linear optimization over the feasible region $X$. More specifically, we introduce the CGS method for smooth convex problems in Subsection 2.1 and generalize it for smooth and strongly convex problems in Subsection 2.2.

### 2.1. Smooth convex optimization.

The basic scheme of the CGS method is obtained by applying the classic conditional gradient (CndG) method to solve the projection subproblems existing in the accelerated gradient (AG) method approximately. By properly specifying the accuracy for solving these subproblems, we will show that the resulting CGS method can achieve the optimal bounds on the number of calls to the FO and LO oracles for solving problem (1.1). The development of the CGS method, in spirit, is similar to the gradient sliding algorithm developed by Lan in [28] for solving a class of composite optimization problems. It should be noted, however, that the gradient sliding algorithm in [28] requires to perform projections over $X$ and targets to solve CP problems with a general nonsmooth term in the objective function.

The CGS method is formally described as follows.

**Algorithm 1** The conditional gradient sliding (CGS) method

```
Input: Initial point $x_0 \in X$ and iteration limit $N$.
Let $\beta_k \in \mathbb{R}_{++}, \gamma_k \in [0,1]$, and $\eta_k \in \mathbb{R}_+$, $k = 1, 2, \ldots$, be given and set $y_0 = x_0$.
for $k = 1, 2, \ldots, N$ do
        $z_k = (1 - \gamma_k) y_{k-1} + \gamma_k x_{k-1},$    (2.1)
        $x_k = \mathrm{CndG}(f'(z_k), x_{k-1}, \beta_k, \eta_k),$    (2.2)
        $y_k = (1 - \gamma_k) y_{k-1} + \gamma_k x_k.$    (2.3)
end for
Output: $y_N$.

procedure $u^+ = \mathrm{CndG}(g,\ u,\ \beta,\ \eta)$
    1. Set $u_1 = u$ and $t = 1$.
    2. Let $v_t$ be an optimal solution for the subproblem of
        $V_{g,u,\beta}(u_t) := \max_{x \in X} \langle g + \beta(u_t - u), u_t - x \rangle.$    (2.4)
    3. If $V_{g,u,\beta}(u_t) \le \eta$, set $u^+ = u_t$ and terminate the procedure.
    4. Set $u_{t+1} = (1 - \alpha_t) u_t + \alpha_t v_t$ with
        $\alpha_t = \min \left\{ 1, \frac{\langle \beta(u - u_t) - g, v_t - u_t \rangle}{\beta \|v_t - u_t\|^2} \right\}.$    (2.5)
    5 Set $t \leftarrow t + 1$ and go to step 2.
end procedure
```

Clearly, the most crucial step of the CGS method is to update the search point $x_k$ by calling the CndG procedure in (2.2). Denoting $\phi(x) := \langle g, x \rangle + \beta \|x - u\|^2/2$, the CndG procedure can be viewed as a specialized version of the classical conditional gradient method applied to $\min_{x \in X} \phi(x)$. In particular, it can be easily seen that $V_{g,u,\beta}(u_t)$ in (2.4) is equivalent to $\max_{x \in X} \langle \phi'(u_t), u_t - x \rangle$, which is often called the Wolfe gap, and the CndG procedure terminates whenever $V_{g,u,\beta}(u_t)$ is smaller than the pre-specified tolerance $\eta$. In fact, this procedure is slightly simpler than the generic conditional gradient method in that the selection of $\alpha_t$ in (2.5) explicitly solves

$$\alpha_t = \mathrm{argmin}_{\alpha \in [0,1]} \phi((1 - \alpha) u_t + \alpha v_t). \tag{2.6}$$

It should be pointed out that (2.5) has been initially suggested by Frank and Wolfe to specify the stepsizes for the CndG method through the minimization of an upper quadratic approximation of $f(\cdot)$ at $x_k$ (see (6.6) in [13], and some more recent developments in [4, 5]). In view of the above discussion, we can easily see that $x_k$ obtained in (2.2) is an approximate solution for the projection subproblem

$$\min_{x \in X} \left\{ \phi_k(x) := \langle f'(z_k), x \rangle + \frac{\beta_k}{2} \|x - x_{k-1}\|^2 \right\} \tag{2.7}$$

such that

$$\langle \phi_k'(x_k), x_k - x \rangle = \langle f'(z_k) + \beta_k (x_k - x_{k-1}), x_k - x \rangle \le \eta_k, \quad \forall x \in X, \tag{2.8}$$

for some $\eta_k \ge 0$.

Clearly, problem (2.7) is equivalent to $\min_{x \in X} \beta_k/2 \|x - x_{k-1} + f'(z_k)/\beta_k\|^2$ after completing the square, and it admits explicit solutions in some special cases, e.g., when $X$ is a standard Euclidean ball. However, this paper focuses on the case where (2.7) is solved iteratively by calling the LO oracle.

We now add a few comments about the main CGS method. Firstly, similarly to the accelerated gradient method, the above CGS method maintains the updating of three intertwined sequences, namely $\{x_k\}$, $\{y_k\}$, and $\{z_k\}$, in each iteration. The main difference between CGS and the original AG exists in the computation of $x_k$. More specifically, $x_k$ in the original AG method is set to the exact solution of (2.7) (i.e., $\eta_k = 0$ in (2.8)), while the subproblem in (2.7) is only solved approximately for the CGS method (i.e., $\eta_k > 0$ in (2.8)).

Secondly, we say that an inner iteration of the CGS method occurs whenever the index $t$ in the CndG procedure increments by 1. Accordingly, an outer iteration of CGS occurs whenever $k$ increases by 1. While we need to call the FO oracle to compute the gradient $f'(z_k)$ in each outer iteration, the gradient $\phi'_k(p_t)$ used in the CndG subroutine is given explicitly by $f'(z_k) + \beta_k(p - x_{k-1})$. Hence, the main cost per each inner iteration of the CGS method is to call the LO oracle to solve linear optimization problem in (2.4). As a result, the total number of outer and inner iterations performed by the CGS algorithm are equivalent to the total number of calls to the FO and LO oracles, respectively.

Thirdly, observe that the above CGS method is conceptual only since we have not specified a few parameters, including $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$, used in this algorithm yet. We will come back to this issue after establishing some important convergence properties for the above generic CGS algorithm.

We first state a simple technical result that will be used in the analysis of the CGS algorithm. This result slightly generalizes Lemma 3 of [27].

Lemma 2.1. *Let $w_t \in (0,1]$, $t = 1, 2, \ldots$, be given. Also let us denote*

$$W_t := \begin{cases} 1 & t = 1 \\ (1 - w_t)W_{t-1} & t \geq 2. \end{cases} \tag{2.9}$$

*Suppose that $W_t > 0$ for all $t \geq 2$ and that the sequence $\{\delta_t\}_{t \geq 0}$ satisfies*

$$\delta_t \leq (1 - w_t)\delta_{t-1} + B_t, \quad t = 1, 2, \ldots. \tag{2.10}$$

*Then for any $1 \leq l \leq k$, we have*

$$\delta_k \leq W_k \left( \frac{1 - w_l}{W_l} \delta_{l-1} + \sum_{i=l}^{k} \frac{B_i}{W_i} \right). \tag{2.11}$$

*Proof.* Dividing both sides of (2.10) by $W_t$, we obtain

$$\frac{\delta_1}{W_1} \leq \frac{(1 - w_1)\delta_0}{W_1} + \frac{B_1}{W_1}$$

and

$$\frac{\delta_i}{W_i} \leq \frac{(1 - w_i)\delta_{i-1}}{W_i} + \frac{B_i}{W_i} = \frac{\delta_{i-1}}{W_{i-1}} + \frac{B_i}{W_i}, \quad \forall i \geq 2.$$

The result then immediately follows by summing up the above inequalities for $i = l, \ldots, k$ and rearranging the terms. ■

Theorem 2.2 describes the main convergence properties of the above CGS method. More specifically, both Theorem 2.2.a) and b) show the convergence of the AG method when the projection subproblem is approximately solved according to (2.8), while Theorem 2.2.c) states the convergence of the CndG procedure by using the Wolfe gap as the termination criterion. To the best of our knowledge, the analysis of the AG method under the inexact projection condition in (2.8) has not been studied before in the literature (see [28] for the analysis of a different inexact AG method), while the convergence of the CndG method using the Wolfe gap as the termination criterion has been well-understood in the literature (see, e.g., [23, 14]). Hence, part c) is included here mainly for the sake of completeness. It should be noted, however, that the analysis we provided in part c) is more specialized to problem (2.7), and seems to be slightly simpler than those given in [23, 14].

Observe that the following quantity will be used in the convergence analysis of the CGS algorithm:

$$\Gamma_k := \begin{cases} 1 & k = 1 \\ \Gamma_{k-1}(1-\gamma_k) & k \geq 2. \end{cases} \tag{2.12}$$

THEOREM 2.2. *Let $\Gamma_k$ be defined in (2.12). Suppose that $\{\beta_k\}$ and $\{\gamma_k\}$ in the CGS algorithm satisfy*

$$\gamma_1 = 1 \quad \text{and} \quad L\gamma_k \leq \beta_k, \quad k \geq 1. \tag{2.13}$$

*a) If*

$$\frac{\beta_k \gamma_k}{\Gamma_k} \geq \frac{\beta_{k-1}\gamma_{k-1}}{\Gamma_{k-1}}, \quad k \geq 2, \tag{2.14}$$

*then for any $x \in X$ and $k \geq 1$,*

$$f(y_k) - f(x^*) \leq \frac{\beta_k \gamma_k}{2} D_X^2 + \Gamma_k \sum_{i=1}^k \frac{\eta_i \gamma_i}{\Gamma_i}. \tag{2.15}$$

*where $x^*$ is an arbitrary optimal solution of (1.1) and $D_X$ is defined in (1.4).*

*b) If*

$$\frac{\beta_k \gamma_k}{\Gamma_k} \leq \frac{\beta_{k-1}\gamma_{k-1}}{\Gamma_{k-1}}, \quad k \geq 2, \tag{2.16}$$

*then for any $x \in X$ and $k \geq 1$,*

$$f(y_k) - f(x^*) \leq \frac{\beta_1 \Gamma_k}{2} \|x_0 - x^*\|^2 + \Gamma_k \sum_{i=1}^k \frac{\eta_i \gamma_i}{\Gamma_i}. \tag{2.17}$$

*c) Under the assumptions in either part a) or b), the number of inner iterations performed at the k-th outer iteration can be bounded by*

$$T_k := \left\lceil \frac{6\beta_k D_X^2}{\eta_k} \right\rceil, \quad \forall k \geq 1. \tag{2.18}$$

*Proof.* We first show part a). Note that by (2.1) and (2.3), we have $y_k - z_k = \gamma_k(x_k - x_{k-1})$. By using this observation, (1.7) and (2.3) we have

$$\begin{aligned} f(y_k) &\leq l_f(z_k; y_k) + \frac{L}{2}\|y_k - z_k\|^2 \\ &= (1-\gamma_k) l_f(z_k; y_{k-1}) + \gamma_k l_f(z_k; x_k) + \frac{L\gamma_k^2}{2}\|x_k - x_{k-1}\|^2 \\ &= (1-\gamma_k) l_f(z_k; y_{k-1}) + \gamma_k l_f(z_k; x_k) + \frac{\beta_k\gamma_k}{2}\|x_k - x_{k-1}\|^2 - \frac{\gamma_k}{2}\left(\beta_k - L\gamma_k\right)\|x_k - x_{k-1}\|^2 \\ &\leq (1-\gamma_k) f(y_{k-1}) + \gamma_k l_f(z_k; x_k) + \frac{\beta_k\gamma_k}{2}\|x_k - x_{k-1}\|^2, \end{aligned} \tag{2.19}$$

where the last inequality follows from the convexity of $f(\cdot)$ and (2.13). Also observe that by (2.8), we have

$$\langle f'(z_k) + \beta_k(x_k - x_{k-1}), x_k - x \rangle \leq \eta_k, \quad \forall x \in X,$$

which implies that

$$\begin{aligned}\frac{1}{2}\|x_k - x_{k-1}\|^2 &= \frac{1}{2}\|x_{k-1} - x\|^2 - \langle x_{k-1} - x_k, x_k - x\rangle - \frac{1}{2}\|x_k - x\|^2 \\ &\le \frac{1}{2}\|x_{k-1} - x\|^2 + \frac{1}{\beta_k}\langle f'(z_k), x - x_k\rangle - \frac{1}{2}\|x_k - x\|^2 + \frac{\eta_k}{\beta_k}. \qquad (2.20)\end{aligned}$$

Combining (2.19) and (2.20), we obtain

$$\begin{aligned}f(y_k) &\le (1-\gamma_k)f(y_{k-1}) + \gamma_k l_f(z_k; x) + \frac{\beta_k\gamma_k}{2}\left(\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right) + \eta_k\gamma_k \\ &\le (1-\gamma_k)f(y_{k-1}) + \gamma_k f(x) + \frac{\beta_k\gamma_k}{2}\left(\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right) + \eta_k\gamma_k, \quad \forall x \in X, \qquad (2.21)\end{aligned}$$

where the last inequality follows from the convexity of $f(\cdot)$. Subtracting $f(x)$ from both sides of the above inequality, we have

$$f(y_k) - f(x) \le (1-\gamma_k)[f(y_{k-1}) - f(x)] + \frac{\beta_k\gamma_k}{2}\left(\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right) + \eta_k\gamma_k, \quad \forall x \in X.$$

which, in view of Lemma 2.1, then implies that

$$\begin{aligned}f(y_k) - f(x) &\le \frac{\Gamma_k(1-\gamma_1)}{\Gamma_1}[f(y_0) - f(x)] \\ &\quad + \Gamma_k \sum_{i=1}^{k} \frac{\beta_i\gamma_i}{2\Gamma_i}(\|x_{i-1} - x\|^2 - \|x_i - x\|^2) + \Gamma_k \sum_{i=1}^{k} \frac{\eta_i\gamma_i}{\Gamma_i}. \qquad (2.22)\end{aligned}$$

Our result in part a) then immediately follows from the above inequality, the assumption that $\gamma_1 = 1$, and the fact that

$$\begin{aligned}&\sum_{i=1}^{k} \frac{\beta_i\gamma_i}{\Gamma_i}(\|x_{i-1} - x\|^2 - \|x_i - x\|^2) \\ &= \frac{\beta_1\gamma_1}{\Gamma_1}\|x_0 - x\|^2 + \sum_{i=2}^{k}\left(\frac{\beta_i\gamma_i}{\Gamma_i} - \frac{\beta_{i-1}\gamma_{i-1}}{\Gamma_{i-1}}\right)\|x_{i-1} - x\|^2 - \frac{\beta_k\gamma_k}{\Gamma_k}\|x_k - x\|^2 \\ &\le \frac{\beta_1\gamma_1}{\Gamma_1}D_X^2 + \sum_{i=2}^{k}\left(\frac{\beta_i\gamma_i}{\Gamma_i} - \frac{\beta_{i-1}\gamma_{i-1}}{\Gamma_{i-1}}\right)D_X^2 = \frac{\beta_k\gamma_k}{\Gamma_k}D_X^2, \qquad (2.23)\end{aligned}$$

where the inequality follows from the third assumption in (2.14) and the definition of $D_X$ in (1.4).

Similarly, Part b) follows from (2.22), the assumption that $\gamma_1 = 1$, and the fact that

$$\sum_{i=1}^{k} \frac{\beta_i\gamma_i}{\Gamma_i}(\|x_{i-1} - x\|^2 - \|x_i - x\|^2) \le \frac{\beta_1\gamma_1}{\Gamma_1}\|x_0 - x\|^2 - \frac{\beta_k\gamma_k}{\Gamma_k}\|x_k - x\|^2 \le \beta_1\|x_0 - x\|^2, \qquad (2.24)$$

due to the assumptions in (2.13) and (2.16).

Now we show that part c) holds. Let us denote $\phi \equiv \phi_k$ and $\phi^* \equiv \min_{x\in X}\phi(x)$. Also let us denote

$$\lambda_t := \frac{2}{t} \quad \text{and} \quad \Lambda_t = \frac{2}{t(t-1)}. \qquad (2.25)$$

It then follows from the above definitions that

$$\Lambda_{t+1} = \Lambda_t(1 - \lambda_{t+1}), \quad \forall\, t \ge 2. \qquad (2.26)$$

Let us define $\bar{u}_{t+1} := (1-\lambda_{t+1})u_t + \lambda_{t+1}v_t$. Clearly we have $\bar{u}_{t+1} - u_t = \lambda_{t+1}(v_t - u_t)$. Observe that $u_{t+1} = (1-\alpha_t)u_t + \alpha_t v_t$ and $\alpha_t$ is an optimal solution of (2.6), and hence that $\phi(u_{t+1}) \le \phi(\bar{u}_{t+1})$. Using this observation, (1.7) and the fact that $\phi$ has Lipschitz continuous gradients, we have

$$\begin{aligned}\phi(u_{t+1}) \le \phi(\bar{u}_{t+1}) &\le l_\phi(u_t, \bar{u}_{t+1}) + \frac{\beta}{2}\|\bar{u}_{t+1} - u_t\|^2 \\ &\le (1-\lambda_{t+1})\phi(u_t) + \lambda_{t+1} l_\phi(u_t, v_t) + \frac{\beta\lambda_{t+1}^2}{2}\|v_t - u_t\|^2. \end{aligned} \tag{2.27}$$

Also observe that by (1.6) and the fact that $v_t$ solves (2.4), we have

$$l_\phi(u_t, v_t) = \phi(u_t) + \langle \phi'(u_t), v_t - u_t \rangle \le \phi(u_t) + \langle \phi'(u_t), x - u_t \rangle \le \phi(x)$$

for any $x \in X$, where the last inequality follows from the convexity of $\phi(\cdot)$. Combining the above two inequalities and re-arranging the terms, we obtain

$$\phi(u_{t+1}) - \phi(x) \le (1-\lambda_{t+1})[\phi(u_t) - \phi(x)] + \frac{\beta\lambda_{t+1}^2}{2}\|v_t - u_t\|^2, \quad \forall x \in X,$$

which, in view of Lemma 2.1, then implies that, for any $x \in X$ and $t \ge 1$,

$$\phi(u_{t+1}) - \phi(x) \le \Lambda_{t+1}(1-\lambda_2)[\phi(u_1) - \phi(x)] + \Lambda_{t+1}\beta \sum_{j=1}^{t} \frac{\lambda_{j+1}^2}{2\Lambda_{j+1}}\|v_j - u_j\|^2 \le \frac{2\beta D_X^2}{t+1}, \tag{2.28}$$

where the last inequality easily follows from (2.25) and the definition of $D_X$ in (1.4). Now, let the gap function $V_{g,u,\beta}$ be defined in (2.4). Also let us denote $\Delta_j = \phi(u_j) - \phi^*$. It then follows from (1.6), (2.4), and (2.27) that that for any $j = 1, \ldots, t$,

$$\begin{aligned}\lambda_{j+1} V_{g,u,\beta}(u_j) &\le \phi(u_j) - \phi(u_{j+1}) + \frac{\beta\lambda_{j+1}^2}{2}\|v_j - u_j\|^2 \\ &= \Delta_j - \Delta_{j+1} + \frac{\beta\lambda_{j+1}^2}{2}\|v_j - u_j\|^2.\end{aligned}$$

Dividing both sides of the above inequality by $\Lambda_{j+1}$ and summing up the resulting inequalities, we obtain

$$\begin{aligned}\sum_{j=1}^{t} \frac{\lambda_{j+1}}{\Lambda_{j+1}} V_{g,u,\beta}(u_j) &\le -\frac{1}{\Lambda_{t+1}}\Delta_{t+1} + \sum_{j=2}^{t}\left(\frac{1}{\Lambda_{j+1}} - \frac{1}{\Lambda_j}\right)\Delta_j + \Delta_1 + \sum_{j=1}^{t}\frac{\beta\lambda_{j+1}^2}{2\Lambda_{j+1}}\|v_j - u_j\|^2 \\ &\le \sum_{j=2}^{t}\left(\frac{1}{\Lambda_{j+1}} - \frac{1}{\Lambda_j}\right)\Delta_j + \Delta_1 + \sum_{j=1}^{t}\frac{\beta\lambda_{j+1}^2}{2\Lambda_{j+1}}D_X^2 \le \sum_{j=1}^{t} j\Delta_j + t\beta D_X^2,\end{aligned}$$

where the last inequality follows from the definitions of $\lambda_t$ and $\Lambda_t$ in (2.25). Using the above inequality and the bound on $\Delta_j$ given in (2.28), we conclude that

$$\min_{j=1,\ldots,t} V_{g,u,\beta}(u_j) \sum_{j=1}^{t}\frac{\lambda_{j+1}}{\Lambda_{j+1}} \le \sum_{j=1}^{t}\frac{\lambda_{j+1}}{\Lambda_{j+1}} V_{g,u,\beta}(u_j) \le 3t\beta D_X^2,$$

which, in view of the fact that $\sum_{j=1}^{t} \lambda_{j+1}/\Lambda_{j+1} = t(t+1)/2$, then clearly implies that

$$\min_{j=1,\ldots,t} V_{g,u,\beta}(u_j) \le \frac{6\beta D_X^2}{t+1}, \quad \forall t \ge 1, \tag{2.29}$$

from which part c) immediately follows. ∎

Clearly, there exist various options to specify the parameters $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$ so as to guarantee the convergence of the CGS method. In the following corollaries, we provide two different parameter settings for $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$, which lead to optimal complexity bounds on the total number of calls to the FO and LO oracles for smooth convex optimization.

Corollary 2.3. *If $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$ in the CGS method are set to*

$$\beta_k = \frac{3L}{k+1}, \quad \gamma_k = \frac{3}{k+2}, \quad \text{and} \quad \eta_k = \frac{LD_X^2}{k(k+1)}, \quad \forall k \geq 1, \tag{2.30}$$

*then for any $k \geq 1$,*

$$f(y_k) - f(x^*) \leq \frac{15LD_X^2}{2(k+1)(k+2)}. \tag{2.31}$$

*As a consequence, the total number of calls to the* FO *and* LO *oracles performed by the CGS method for finding an $\epsilon$-solution of (1.1) can be bounded by $\mathcal{O}\left(\sqrt{LD_X^2/\epsilon}\right)$ and $\mathcal{O}\left(LD_X^2/\epsilon\right)$, respectively.*

*Proof.* We first show Part a). It can be easily seen from (2.30) that (2.13) holds. Also note that by (2.30), we have

$$\Gamma_k = \frac{6}{k(k+1)(k+2)}, \tag{2.32}$$

and

$$\frac{\beta_k \gamma_k}{\Gamma_k} = \frac{9L}{(k+1)(k+2)} \frac{k(k+1)(k+2)}{6} = \frac{3Lk}{2},$$

which implies that (2.14) is satisfied. It then follows from Theorem 2.2.a), (2.30), and (2.32) that

$$f(y_k) - f(x^*) \leq \frac{9LD_X^2}{2(k+1)(k+2)} + \frac{6}{k(k+1)(k+2)} \sum_{i=1}^{k} \frac{\eta_i \gamma_i}{\Gamma_i} = \frac{15LD_X^2}{2(k+1)(k+2)},$$

which implies that the total number of outer iterations performed by the CGS method for finding an $\epsilon$-solution can be bounded by $N = \sqrt{15LD_X^2/(2\epsilon)}$. Moreover, it follows from the bound in (2.18) and (2.30) that the total number of inner iterations can be bounded by

$$\sum_{k=1}^{N} T_k \leq \sum_{k=1}^{N} \left( \frac{6\beta_k D_X^2}{\eta_k} + 1 \right) = 18 \sum_{k=1}^{N} k + N = 9N^2 + 10N,$$

which implies that the total number of inner iterations is bounded by $\mathcal{O}(LD_X^2/\epsilon)$. ∎

Observe that in the above result, the number of calls to the LO oracle is not improvable in terms of their dependence on $\epsilon$, $L$, and $D_X$ for LCP methods [27]. Similarly, the number of calls to the FO oracle is also optimal in terms of its dependence on $\epsilon$ and $L$ [30, 32]. It should be noted, however, that we can potentially improve the latter bound in terms of its dependence on $D_X$. Indeed, by using a different parameter setting, we show in Corollary 2.4 a slightly improved bound on the number of calls to the FO oracle which only depends on the distance from the initial point to the set of optimal solutions, rather than the diameter $D_X$. This result will play an important role for the analysis of the CGS method for solving strongly convex problems. The disadvantage of using this parameter setting is that we need to fix the number of iterations $N$ in advance.

Corollary 2.4. *Suppose that there exists an estimate $D_0 \geq \|x_0 - x^*\|$ and that the outer iteration limit $N \geq 1$ is given. If*

$$\beta_k = \frac{2L}{k}, \quad \gamma_k = \frac{2}{k+1}, \quad \eta_k = \frac{2LD_0^2}{Nk}, \tag{2.33}$$

*for any $k \ge 1$, then*

$$f(y_N) - f(x^*) \le \frac{6LD_0^2}{N(N+1)}. \tag{2.34}$$

*As a consequence, the total number of calls to the* FO *and* LO *oracles performed by the CGS method for finding an $\epsilon$-solution of (1.1), respectively, can be bound by*

$$\mathcal{O}\left(D_0\sqrt{\frac{L}{\epsilon}}\right) \tag{2.35}$$

*and*

$$\mathcal{O}\left(\frac{LD_X^2}{\epsilon} + D_0\sqrt{\frac{L}{\epsilon}}\right). \tag{2.36}$$

*Proof.* It can be easily seen from the definition of $\gamma_k$ in (2.33) and $\Gamma_k$ in (2.12) that

$$\Gamma_k = \frac{2}{k(k+1)}. \tag{2.37}$$

Using the previous identity and (2.33), we have $\beta_k\gamma_k/\Gamma_k = 2L$, which implies that (2.16) holds. It then follows from (2.17), (2.33), and (2.37) that

$$f(y_N) - f(x^*) \le \Gamma_N \left(LD_0^2 + \sum_{i=1}^N \frac{\eta_i\gamma_i}{\Gamma_i}\right) = \Gamma_N \left(LD_0^2 + \sum_{i=1}^N i\eta_i\right) = \frac{6LD_0^2}{N(N+1)}.$$

Moreover, it follows from the bound in (2.18) and (2.33) that the total number of inner iterations can be bounded by

$$\sum_{k=1}^N T_k \le \sum_{k=1}^N \left(\frac{6\beta_k D_X^2}{\eta_k} + 1\right) = \frac{6N^2D_X^2}{D_0^2} + N.$$

The complexity bounds in (2.35) and (2.36) then immediately follow from the previous two inequalities. ■

In view of the classic complexity theory for convex optimization, the bound on the total number of calls to FO oracle in (2.35) is optimal for smooth convex optimization. Moreover, in view of the complexity results established in [27], the total number of calls to the LO oracle in (2.36) is not improvable for a wide class of LCP methods. To the best of our knowledge, the CGS method is the first algorithm in the literature that can achieve these two optimal bounds at the same time.

**Remark 1.** Observe that in this section, we have assumed that the Euclidean distance function $\|x - x_{k-1}\|^2$ has been used in the subproblem (2.7). However, one can also replace it with the more general Bregman distance

$$V(x, x_{k-1}) := \omega(x) - [\omega(x_{k-1}) + \langle \omega'(x_{k-1}), x - x_{k-1}\rangle]$$

and relax the assumption that the norms are associated with the inner product, where $\omega$ is a strongly convex function. We can show similar complexity results as those in Corollaries 2.3 and 2.4 under the following assumptions: i) $\omega$ is a smooth convex function with Lipschitz continuous gradients; and ii) in the CndG subroutine, the objective function in (2.4) and the stepsizes $\alpha_t$ in (2.5) are replaced by $g + \beta[\omega'(u_t) - \omega'(u)]$ and $2/(t+1)$, respectively. However, if $\omega$ is nonsmooth (e.g., the entropy function), then we cannot obtain these results since the CndG subroutine cannot be directly applied to the modified subproblem. One possible remedy to this issue is to incorporate the randomized smoothing technique into the CndG subroutine (see [27]).

**2.2. Strongly convex optimization.** In this subsection, we assume that the objective function $f$ is not only smooth (i.e., (1.7) holds), but also strongly convex, that is, $\exists \mu > 0$ s.t.

$$f(y) - f(x) - \langle f'(x), y - x\rangle \ge \frac{\mu}{2}\|y - x\|^2, \quad \forall x, y \in X. \tag{2.38}$$

Our goal is to show that a linear rate of convergence, in terms of the number of calls to the FO oracle, can be obtained by only performing linear optimization over the feasible region $X$. In contrast with the shrinking conditional gradient method in [27], here we do not need to enforce any additional assumptions on the LO oracle. We also show that the total number of calls to the LO oracle is bounded by $\mathcal{O}(LD_X^2/\epsilon)$, which has been shown to be optimal for strongly convex optimization (see, e.g., [23, 27]).

We are now ready to formally describe the CGS method for solving strongly convex problems, which is obtained by properly restarting the CGS method in Algorithm 1.

**Algorithm 2** The CGS method for strongly convex problems

```
Input: Initial point $p_0 \in X$ and an estimate $\delta_0 > 0$ satisfying $f(p_0) - f(x^*) \le \delta_0$.
for $s = 1, 2, \ldots$
    Call the CGS method in Algorithm 1 with input
        $x_0 = p_{s-1} \quad \text{and} \quad N = \left\lceil 2\sqrt{\frac{6L}{\mu}}\right\rceil,$          (2.39)
    and parameters
        $\beta_k = \frac{2L}{k}, \quad \gamma_k = \frac{2}{k+1}, \quad \text{and} \quad \eta_k = \eta_{s,k} := \frac{8L\delta_0 2^{-s}}{\mu N k},$          (2.40)
    and let $p_s$ be its output solution.
end for
```

In Algorithm 2, we restart the CGS method for smooth optimization (i.e., Algorithm 1) every $\lceil 2\sqrt{6L/\mu}\rceil$ iterations. We say that a phase of the above CGS algorithm occurs whenever $s$ increases by 1. Observe that $\{\eta_k\}$ decrease by a factor of 2 as $s$ increments by 1, while $\{\beta_k\}$ and $\{\gamma_k\}$ remain the same. The following theorem shows the convergence of the above variant of the CGS method.

THEOREM 2.5. *Assume (2.38) holds and let $\{p_s\}$ be generated by Algorithm 2. Then,*

$$f(p_s) - f(x^*) \le \delta_0 2^{-s}, \quad s \ge 0. \tag{2.41}$$

*As a consequence, the total number of calls to the* FO *and* LO *oracles performed by this algorithm for finding an $\epsilon$-solution of problem (1.1) can be bounded by*

$$\mathcal{O}\left\{\sqrt{\frac{L}{\mu}}\left\lceil \log_2 \max\left(1, \frac{\delta_0}{\epsilon}\right)\right\rceil\right\} \tag{2.42}$$

*and*

$$\mathcal{O}\left\{\frac{LD_X^2}{\epsilon} + \sqrt{\frac{L}{\mu}}\left\lceil \log_2 \max\left(1, \frac{\delta_0}{\epsilon}\right)\right\rceil\right\}, \tag{2.43}$$

*respectively.*

*Proof.* We prove (2.41) by using induction. This inequality holds obviously when $s = 0$ due to our assumption on $\delta_0$. Now suppose that (2.41) holds before the $s$-th phase starts, i.e.,

$$f(p_{s-1}) - f(x^*) \le \delta_0 2^{-s+1}.$$

Using the above relation and the strong convexity of $f$, we have

$$\|p_{s-1} - x^*\|^2 \le \frac{2}{\mu}\left[f(p_{s-1}) - f(x^*)\right] \le \frac{4\delta_0 2^{-s}}{\mu}.$$

Hence, by comparing the parameter settings in (2.40) with those in (2.33), we can easily see that Corollary 2.4 holds with $x_0 = p_{s-1}$, $y_N = p_s$, and $D_0^2 = 4\delta_0 2^{-s}/\mu$, which implies that

$$f(y_s) - f(x^*) \le \frac{6LD_0^2}{N(N+1)} = \frac{24L\delta_0 2^{-s}}{\mu N(N+1)} \le \delta_0 2^{-s},$$

where the last inequality follows from the definition of $N$ in (2.39). In order to show the bounds in (2.42) and (2.43), it suffices to consider the case when $\delta_0 > \epsilon$ (otherwise, the results are obvious). Let us denote

$$S := \left\lceil \log_2 \max\left(\frac{\delta_0}{\epsilon}, 1\right)\right\rceil. \tag{2.44}$$

By (2.41), an $\epsilon$-solution of (1.1) can be found at the $s$-th phase for some $1 \le s \le S$. Since the number of calls to the FO in each phase is bounded by $N$, the total number of calls to the FO performed by Algorithm 2 is clearly bounded by $NS$, which is bounded by (2.42). Now, let $T_{s,k}$ denote the number of calls to LO required at the the $k$-th outer iteration in $s$-th phase. It follows from Theorem 2.2.c) that

$$T_{s,k} \le \frac{6\beta_k D_X^2}{\eta_{k,s}} + 1 \le \frac{3\mu D_X^2 2^s N}{2\delta_0} + 1.$$

Therefore, the total number of calls to the LO can be bounded by

$$\begin{aligned}\sum_{s=1}^{S}\sum_{k=1}^{N} T_{s,k} &\le \sum_{s=1}^{S}\sum_{k=1}^{N} \frac{3\mu D_X^2 2^s N}{2\delta_0} + NS = \frac{3\mu D_X^2 N^2}{2\delta_0}\sum_{s=1}^{S} 2^s + NS \\ &\le \frac{3\mu D_X^2 N^2}{2\delta_0} 2^{S+1} + NS \\ &\le \frac{6}{\epsilon}\mu D_X^2 N^2 + NS, \end{aligned} \tag{2.45}$$

which is bounded by (2.43) due to the definitions of $N$ and $S$ in (2.39) and (2.44), respectively. ■

In view of the classic complexity theory for convex optimization, the bound on the total number of calls to FO oracle in (2.42) is optimal for strongly convex optimization. Moreover, in view of the complexity results established in [27], the bound on the total number of calls to the LO oracle in (2.43) is also not improvable for a wide class of linear-optimization based convex programming methods. To the best of our knowledge, this is the first time that these two bounds were achieved simultaneously by a single optimization algorithm.

## 3. The stochastic conditional gradient sliding method.

**3.1. The algorithm and the main convergence results.** In this section, we still consider smooth convex optimization problems satisfying (1.2). However, here we only have access to the stochastic first-order information about $f$. More specifically, we assume that $f$ is represented by a stochastic first-order (SFO) oracle, which, for a given search point $z_k \in X$, outputs a vector $G(z_k, \xi_k)$ s.t.

$$\mathbb{E}\left[G(z_k, \xi_k)\right] = f'(z_k), \tag{3.1}$$

$$\mathbb{E}\left[\|G(z_k, \xi_k) - f'(z_k)\|_*^2\right] \le \sigma^2. \tag{3.2}$$

Our goal in this section is to present a stochastic conditional gradient type algorithm that can achieve the optimal bounds on the number of calls to SFO and LO oracles, while no such algorithms have been developed before in the literature.

The stochastic CGS (SCGS) method is obtained by simply replacing the exact gradients in Algorithm 1 with an unbiased estimator computed by the SFO oracle. The algorithm is formally described as follows.

**Algorithm 3** The stochastic conditional gradient sliding method

This algorithm is the same as Algorithm 1 except that (2.2) is replaced by

$$x_k = \mathrm{CndG}(g_k, x_{k-1}, \beta_k, \eta_k). \tag{3.3}$$

Here,

$$g_k := \frac{1}{B_k}\sum_{j=1}^{B_k} G(z_k, \xi_{k,j}) \tag{3.4}$$

and $G(z_k, \xi_{k,j})$, $j = 1, \ldots, B_k$, are stochastic gradients computed by the SFO at $z_k$.

In the above stochastic CGS method, the parameters $\{B_k\}$ denote the batch sizes used to compute $g_k$. It can be easily seen from (3.1), (3.2), and (3.4) that

$$\mathbb{E}[g_k - f'(z_k)] = 0 \ \text{ and } \ \mathbb{E}[\|g_k - f'(z_k)\|_*^2] \le \frac{\sigma^2}{B_k} \tag{3.5}$$

and hence $g_k$ is an unbiased estimator of $f'(z_k)$. In fact, letting $S_{B_k} = \sum_{j=1}^{B_k}(G(z_k, \xi_{k,j}) - f'(z_k))$, from (3.1) and (3.2), we have

$$\begin{aligned}
\mathbb{E}\left[\|S_{B_k}\|_*^2\right] &= \mathbb{E}\left[\|S_{B_k-1} + G(z_k, \xi_{k,B_k}) - f'(z_k)\|_*^2\right] \\
&= \mathbb{E}\left[\|S_{B_k-1}\|_*^2 + 2\langle S_{B_k-1}, G(z_k, \xi_{k,B_k}) - f'(z_k)\rangle + \|G(z_k, \xi_{k,B_k}) - f'(z_k)\|_*^2\right] \\
&= \mathbb{E}\left[\|S_{B_k-1}\|_*^2\right] + \mathbb{E}\left[\|G(z_k, \xi_{k,B_k}) - f'(z_k)\|_*^2\right] = \ldots = \sum_{j=1}^{B_k}\mathbb{E}\left[\|G(z_k, \xi_{k,j}) - f'(z_k)\|_*^2\right] \le B_k\sigma^2.
\end{aligned}$$

Note that by (3.4), we have

$$g_k - f'(z_k) = \frac{1}{B_k}\sum_{j=1}^{B_k} G(z_k, \xi_{k,j}) - f'(z_k) = \frac{1}{B_k}\sum_{j=1}^{B_k}[G(z_k, \xi_{k,j}) - f'(z_k)] = \frac{1}{B_k}S_{B_k}.$$

Therefore, the second relationship in (3.5) immediately follows. Since the algorithm is stochastic, we will establish the complexity for finding a stochastic $\epsilon$-solution, i.e., a point $\bar{x} \in X$ s.t. $\mathbb{E}[f(\bar{x}) - f(x^*)] \le \epsilon$, as well as a stochastic $(\epsilon, \Lambda)$-solution, i.e., a point $\bar{x} \in X$ s.t. $\mathrm{Prob}\{f(\bar{x}) - f(x^*) \le \epsilon\} \ge 1 - \Lambda$ for some $\epsilon > 0$ and $\Lambda \in (0, 1)$.

Observe that the above SCGS method is conceptual only as we have not yet specified the parameters $\{B_k\}$, $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$. We will come back to this issue after establishing the main convergence properties for this algorithm.

Theorem 3.1. *Let $\Gamma_k$ and $D_X$ be defined in (2.12) and (1.4), respectively. Also assume that $\{\beta_k\}$ and $\{\gamma_k\}$ satisfy (2.13) and (2.14).*

*a) Under assumptions (3.1) and (3.2), we have*

$$\mathbb{E}\left[f(y_k) - f(x^*)\right] \le \mathcal{C}_e := \frac{\beta_k\gamma_k}{2}D_X^2 + \Gamma_k\sum_{i=1}^{k}\left[\frac{\eta_i\gamma_i}{\Gamma_i} + \frac{\gamma_i\sigma^2}{2\Gamma_i B_i(\beta_i - L\gamma_i)}\right], \quad \forall k \ge 1, \tag{3.6}$$

*where $x^*$ is an arbitrary optimal solution of (1.1).*

*b) If (2.16) (rather than (2.14)) is satisfied, then the results in part a) still hold by replacing $\beta_k\gamma_k D_X^2$ with $\beta_1\Gamma_k\|x_0 - x^*\|^2$ in the first term of $\mathcal{C}_e$ in (3.6).*

*c) Under the assumptions in part a) or b), the number of inner iterations performed at the k-th outer iterations is bounded by (2.18).*

*Proof.* Let us denote $\delta_{k,j} = G(z_k, \xi_{k,j}) - f'(z_k)$ and $\delta_k \equiv g_k - f'(z_k) = \sum_{j=1}^{B_k} \delta_{k,j}/B_k$ . Note that by (2.19) and (2.20) (with $f'(z_k)$ replaced by $g_k$), we have

$$
\begin{aligned}
f(y_k) &\le (1-\gamma_k) f(y_{k-1}) + \gamma_k l_f(z_k, x_k) + \gamma_k\langle g_k, x - x_k\rangle + \frac{\beta_k\gamma_k}{2}\left[\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right] \\
&\quad + \eta_k\gamma_k - \frac{\gamma_k}{2}\left(\beta_k - L\gamma_k\right)\|x_k - x_{k-1}\|^2 \\
&= (1-\gamma_k) f(y_{k-1}) + \gamma_k l_f(z_k, x) + \gamma_k\langle \delta_k, x - x_k\rangle + \frac{\beta_k\gamma_k}{2}\left[\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right] \\
&\quad + \eta_k\gamma_k - \frac{\gamma_k}{2}\left(\beta_k - L\gamma_k\right)\|x_k - x_{k-1}\|^2.
\end{aligned}
$$

Using the above inequality and the fact that

$$
\begin{aligned}
&\langle \delta_k, x - x_k\rangle - \frac{1}{2}\left(\beta_k - L\gamma_k\right)\|x_k - x_{k-1}\|^2 \\
&\quad = \langle \delta_k, x - x_{k-1}\rangle + \langle \delta_k, x_{k-1} - x_k\rangle - \frac{1}{2}\left(\beta_k - L\gamma_k\right)\|x_k - x_{k-1}\|^2 \\
&\quad \le \langle \delta_k, x - x_{k-1}\rangle + \frac{\|\delta_k\|_*^2}{2(\beta_k - L\gamma_k)},
\end{aligned}
$$

we obtain

$$
\begin{aligned}
f(y_k) \le (1-\gamma_k) f(y_{k-1}) + \gamma_k f(x) + \frac{\beta_k\gamma_k}{2}\left[\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right] + \eta_k\gamma_k \\
+ \gamma_k\langle \delta_k, x - x_{k-1}\rangle + \frac{\gamma_k\|\delta_k\|_*^2}{2(\beta_k - L\gamma_k)}, \quad \forall x \in X. \qquad (3.7)
\end{aligned}
$$

Subtracting $f(x)$ from both sides of (3.7) and using Lemma 2.1, we have

$$
\begin{aligned}
f(y_k) - f(x) &\le \Gamma_k(1-\gamma_1)\left[f(y_0) - f(x)\right] + \Gamma_k\sum_{i=1}^{k}\left\{\frac{\beta_i\gamma_i}{2\Gamma_i}\left[\|x_{k-1} - x\|^2 - \|x_k - x\|^2\right] + \frac{\eta_i\gamma_i}{\Gamma_i}\right\} \\
&\quad + \Gamma_k\sum_{i=1}^{k}\frac{\gamma_i}{\Gamma_i}\left[\langle \delta_i, x - x_{i-1}\rangle + \frac{\|\delta_i\|_*^2}{2(\beta_i - L\gamma_i)}\right] \\
&\le \frac{\beta_k\gamma_k}{2}D_X^2 + \Gamma_k\sum_{i=1}^{k}\frac{\eta_i\gamma_i}{\Gamma_i} + \Gamma_k\sum_{i=1}^{k}\frac{\gamma_i}{\Gamma_i}\left[\sum_{j=1}^{B_i} B_i^{-1}\langle \delta_{i,j}, x - x_{i-1}\rangle + \frac{\|\delta_i\|_*^2}{2(\beta_i - L\gamma_i)}\right], \qquad (3.8)
\end{aligned}
$$

where the last inequality follows from (2.23) and the fact that $\gamma_1 = 1$. Note that by our assumptions on the SFO, the random variables $\delta_{i,j}$ are independent of the search point $x_{i-1}$ and hence $\mathbb{E}[\langle \delta_{i,j}, x^* - x_{i-1}\rangle] = 0$. In addition, relation (3.5) implies that $\mathbb{E}[\|\delta_i\|_*^2] \le \sigma^2/B_i$. Using the previous two observations and taking expectation on both sides of (3.8) (with $x = x^*$), we obtain (3.6).

Part b) follows similarly from the bound in (2.24) and (3.8), and the proof of part c) is exactly the same as that of Theorem 2.2.c). ■

Now we provide a set of parameters $\{\beta_k\}, \{\gamma_k\}, \{\eta_k\}$, and $\{B_k\}$ which lead to optimal bounds on the number of calls to the SFO and LO oracles.

Corollary 3.2. *Suppose that $\{\beta_k\}, \{\gamma_k\}$, $\{\eta_k\}$, and $\{B_k\}$ in the SCGS method are set to*

$$\beta_k = \frac{4L}{k+2}, \quad \gamma_k = \frac{3}{k+2}, \quad \eta_k = \frac{LD_X^2}{k(k+1)}, \quad \text{and} \quad B_k = \left\lceil \frac{\sigma^2(k+2)^3}{L^2 D_X^2} \right\rceil, \quad k \geq 1. \tag{3.9}$$

*Under assumptions (3.1) and (3.2), we have*

$$\mathbb{E}\left[f(y_k) - f(x^*)\right] \leq \frac{6LD_X^2}{(k+2)^2} + \frac{9LD_X^2}{2(k+1)(k+2)}, \quad \forall k \geq 1. \tag{3.10}$$

*As a consequence, the total number of calls to the* SFO *and* LO *oracles performed by the SCGS method for finding a stochastic $\epsilon$-solution of (1.1), respectively, can be bounded by*

$$\mathcal{O}\left\{\sqrt{\frac{LD_X^2}{\epsilon}} + \frac{\sigma^2 D_X^2}{\epsilon^2}\right\} \quad \text{and} \quad \mathcal{O}\left\{\frac{LD_X^2}{\epsilon}\right\}. \tag{3.11}$$

*Proof.* It can be easily seen from (3.9) that (2.13) holds. Also by (3.9), $\Gamma_k$ is given by (2.32) and hence

$$\frac{\beta_k \gamma_k}{\Gamma_k} = \frac{2Lk(k+1)}{k+2},$$

which implies that (2.14) holds. It can also be easily checked from (2.32) and (3.9) that

$$\sum_{i=1}^{k} \frac{\eta_i \gamma_i}{\Gamma_i} \leq \frac{kLD_X^2}{2}, \quad \sum_{i=1}^{k} \frac{\gamma_i}{\Gamma_i B_i (\beta_i - L\gamma_i)} \leq \frac{kLD_X^2}{2\sigma^2}.$$

Using the bound in (3.6), we obtain (3.10), which implies that the total number of outer iterations can be bounded by

$$\mathcal{O}\left(\sqrt{\frac{LD_X^2}{\epsilon}}\right)$$

under the assumptions (3.1) and (3.2). The bounds in (3.11) then immediately follow from this observation and the fact that the number of calls to the SFO and LO oracles are bounded by

$$\sum_{k=1}^{N} B_k \leq \sum_{k=1}^{N} \frac{\sigma^2 (k+2)^3}{L^2 D_X^2} + N \leq \frac{\sigma^2 (N+3)^4}{4L^2 D_X^2} + N,$$
$$\sum_{k=1}^{N} T_k \leq \sum_{k=1}^{N} \left(\frac{6\beta_k D_X^2}{\eta_k} + 1\right) \leq 12N^2 + 13N.$$

■

Now we give a different set of parameters $\{\beta_k\}, \{\gamma_k\}, \{\eta_k\}$, and $\{B_k\}$, which can slightly improve the bounds on the number of calls to the SFO in terms of its dependence on $D_X$.

Corollary 3.3. *Suppose that there exists an estimate $D_0$ s.t. $\|x_0 - x^*\| \leq D_0 \leq D_X$. Also assume that the outer iteration limit $N \geq 1$ is given. If*

$$\beta_k = \frac{3L}{k}, \quad \gamma_k = \frac{2}{k+1}, \quad \eta_k = \frac{2LD_0^2}{Nk}, \quad \text{and} \quad B_k = \left\lceil \frac{\sigma^2 N (k+1)^2}{L^2 D_0^2} \right\rceil, \quad k \geq 1. \tag{3.12}$$

*Under assumptions (3.1) and (3.2),*

$$\mathbb{E}\left[f(y_N) - f(x^*)\right] \leq \frac{8LD_0^2}{N(N+1)}, \quad \forall N \geq 1. \tag{3.13}$$

*As a consequence, the total number of calls to the* SFO *and* LO *oracles performed by the SCGS method for finding a stochastic $\epsilon$-solution of (1.1), respectively, can be bounded by*

$$\mathcal{O}\left\{\sqrt{\frac{LD_0^2}{\epsilon}} + \frac{\sigma^2 D_0^2}{\epsilon^2}\right\} \quad \textit{and} \quad \mathcal{O}\left\{\frac{LD_X^2}{\epsilon}\right\}. \tag{3.14}$$

*Proof.* It can be easily seen from (3.12) that (2.13) holds. Also by (3.12), $\Gamma_k$ is given by (2.37) and hence

$$\frac{\beta_k \gamma_k}{\Gamma_k} = 3L,$$

which implies that (2.16) holds. It can also be easily checked from (2.37) and (3.12) that

$$\sum_{i=1}^{N} \frac{\eta_i \gamma_i}{\Gamma_i} \le 2LD_0^2, \quad \sum_{i=1}^{N} \frac{\gamma_i}{\Gamma_i B_i(\beta_i - L\gamma_i)} \le \sum_{i=1}^{N} \frac{i(i+1)}{LB_i} \le \frac{LD_0^2}{\sigma^2}.$$

Using the bound in (3.6) (with $\beta_k \gamma_k D_X^2$ replaced by $\beta_1 \Gamma_k D_0^2$ in the definition of $\mathcal{C}_e$), we obtain (3.13), which implies that the total number of outer iterations can be bounded by

$$\mathcal{O}\left(\sqrt{\frac{LD_0^2}{\epsilon}}\right)$$

under the assumptions (3.1) and (3.2). The bounds in (3.14) then immediately follow from this observation and the fact that the total number calls to the SFO and LO are bounded by

$$\sum_{k=1}^{N} B_k \le N \sum_{k=1}^{N} \frac{\sigma^2 (k+1)^2}{L^2 D_0^2} + N \le \frac{\sigma^2 N (N+1)^3}{3L^2 D_0^2} + N,$$
$$\sum_{k=1}^{N} T_k \le \sum_{k=1}^{N} \frac{6\beta_k D_X^2}{\eta_k} + N \le \frac{9N^2 D_X^2}{D_0^2} + N.$$

■

According to the complexity bounds in Corollaries 3.2 and 3.3, the total number of calls to the SFO oracle can be bounded by $\mathcal{O}(1/\epsilon^2)$, which is optimal in view of the classic complexity theory for stochastic convex optimization (see [26]). Moreover, the total number of calls to the LO oracle can be bounded by $\mathcal{O}(1/\epsilon)$, which is the same as the CGS method for deterministic smooth convex optimization and hence not improvable for a wide class of LCP methods.

In view of the results in Corollary 3.3, we can present an optimal algorithm for solving stochastic strongly convex problems, similarly to the deterministic case.

**Algorithm 4** The stochastic CGS method for solving strongly convex problems

```
Input: Initial point $p_0 \in X$ and an estimate $\delta_0 > 0$ satisfying $f(p_0) - f(x^*) \le \delta_0$.
for $s = 1, 2, \ldots$
    Call the stochastic CGS method in Algorithm 3 with input
        $x_0 = p_{s-1} \quad \text{and} \quad N = \left\lceil 4\sqrt{\frac{2L}{\mu}} \right\rceil,$    (3.15)
    and parameters
        $\beta_k = \frac{3L}{k},\ \gamma_k = \frac{2}{k+1},\ \eta_k = \eta_{s,k} := \frac{8L\delta_0 2^{-s}}{\mu N k},\ \text{and}\ B_k = B_{s,k} := \left\lceil \frac{\mu\sigma^2 N(k+1)^2}{4L^2\delta_0 2^{-s}} \right\rceil,$    (3.16)
    and let $p_s$ be its output solution.
end for
```

The main convergence properties of Algorithm 4 are described as follows.

Theorem 3.4. *Assume that (2.38) holds and let $\{p_s\}$ be generated by Algorithm 4. Then,*

$$\mathbb{E}[f(p_s) - f(x^*)] \le \delta_0 2^{-s}, \quad s \ge 0. \tag{3.17}$$

*As a consequence, the total number of calls to the* SFO *and* LO *oracles performed by this algorithm for finding a stochastic $\epsilon$-solution of problem (1.1) can be bounded by*

$$\mathcal{O}\left\{ \frac{\sigma^2}{\mu\epsilon} + \sqrt{\frac{L}{\mu}} \left\lceil \log_2 \max\left(1, \frac{\delta_0}{\epsilon}\right) \right\rceil \right\} \tag{3.18}$$

*and*

$$\mathcal{O}\left\{ \frac{LD_X^2}{\epsilon} + \sqrt{\frac{L}{\mu}} \left\lceil \log_2 \max\left(1, \frac{\delta_0}{\epsilon}\right) \right\rceil \right\}, \tag{3.19}$$

*respectively.*

*Proof.* In view of Corollary 3.3, (3.17) can be proved in a way similar to (2.41). It now remains to show the bounds in (3.18) and (3.19), respectively, for the total number of calls to the SFO and LO oracles. It suffices to consider the case when $\delta_0 > \epsilon$, since otherwise the results are obvious. Let us denote

$$S := \left\lceil \log_2 \max\left(\frac{\delta_0}{\epsilon}, 1\right) \right\rceil. \tag{3.20}$$

By (3.17), a stochastic $\epsilon$-solution of (1.1) can be found at the $s$-th phase for some $1 \le s \le S$. Since the number of calls to the SFO oracle in each phase is bounded by $N$, the total number of calls to the SFO oracle can be bounded by

$$\sum_{s=1}^{S}\sum_{k=1}^{N} B_k \le \sum_{s=1}^{S}\sum_{k=1}^{N} \left( \frac{\mu\sigma^2 N(k+1)^2}{4L^2\delta_0 2^{-s}} + 1 \right) \le \frac{\mu\sigma^2 N(N+1)^3}{12L^2\delta_0} \sum_{s=1}^{S} 2^s + SN \le \frac{\mu\sigma^2 N(N+1)^3}{3L^2\epsilon} + SN.$$

Moreover, let $T_{s,k}$ denote the number of calls to LO oracle required at the the $k$-th outer iteration in $s$-th phase of the stochastic CGS method. It follows from Theorem 2.2.c) that

$$T_{s,k} \le \frac{6\beta_k D_X^2}{\eta_{k,s}} + 1 \le \frac{9\mu D_X^2 2^s N}{4\delta_0} + 1.$$

Therefore, the total number of calls to the LO oracle can be bounded by

$$\begin{aligned}\sum_{s=1}^{S}\sum_{k=1}^{N} T_{s,k} &\leq \sum_{s=1}^{S}\sum_{k=1}^{N} \frac{9\mu D_X^2 2^s N}{4\delta_0} + NS = \frac{9}{4}\mu D_X^2 N^2 \delta_0^{-1} \sum_{s=1}^{S} 2^s + NS \\ &\leq \frac{9}{\epsilon}\mu D_X^2 N^2 + NS\end{aligned}$$

which is bounded by (2.43) due to the definitions of $N$ and $S$ in (3.15) and (3.20), respectively. ∎

According to Theorem 3.4, the total number of calls to the SFO oracle can be bounded by $\mathcal{O}(1/\epsilon)$, which is optimal in view of the classic complexity theory for strongly convex optimization (see [16, 17]). Moreover, the total number of calls to the LO oracle can be bounded by $\mathcal{O}(1/\epsilon)$, which is the same as the deterministic CGS method for strongly convex optimization and not improvable for a wide class of LCP methods discussed in [27].

**3.2. The large deviation results.** For the sake of simplicity, in this subsection we only consider smooth convex optimization problems rather than strongly convex problems. In order to develop some large deviation results associated with the aforementioned optimal complexity bounds, we need to make some assumptions about the objective function values and its estimator, $F(x, \xi)$, given by the SFO. More specifically, we assume that

$$\mathbb{E}\left[F(x,\xi)\right] = f(x), \quad \text{and} \quad \mathbb{E}\left[\exp\left\{\left(F(x,\xi) - f(x)\right)^2/M^2\right\}\right] \leq \exp\{1\} \tag{3.21}$$

for some $M \geq 0$.

We now propose a variant of the SCGS method which has some desirable large deviation properties. Similar to the 2-RSPG algorithm in [18], this method consists of two phases: an optimization phase and a post-optimization phase. In the optimization phase, we restart the SCGS algorithm for a certain number of times to generate a list of candidate solutions, and in the post-optimization phase, we choose a solution $\hat{x}$ from this list according to a certain rule.

**Algorithm 5** A two phase SCGS (2-SCGS) algorithm

```
Input: Initial point $x_0 \in X$, number of restart times $S$, iteration limit $N$, and the sample size $K$ in the post-optimization phase.
Optimizatoin phase:
for $s = 1, \ldots, S$ do
    Call the SCGS algorithm with iteration limit $N$, and the initial point $x_{s-1}$, where $x_s = x_{N_s}$, $s = 1, \ldots, S$, are the outputs of the $s$-th run of the SCGS algorithm.
end for
Let $\{\bar{x}_s = x_{N_s},\ s = 1, \ldots, S\}$, be the output list of candidate solutions.
Post-optimization phase:
Choose a solution $\hat{x}$ from the candidate list $\{\bar{x}_1, \ldots, \bar{x}_S\}$ such that
    $\hat{x} = \text{argmin}_{s=1,\ldots,S}\{\hat{f}(\bar{x}_s)\}$,        (3.22)
where $\hat{f}(x) = \frac{1}{K}\sum_{j=1}^{K} F(x, \xi_j)$.
```

Now we are ready to state the large deviation results obtained for the above 2-SCGS algorithm.

Theorem 3.5. *Assuming that $\{\beta_k\}$ and $\{\gamma_k\}$ satisfy (2.13) and (2.14), under assumption (3.21), we have*

$$\text{Prob}\left\{f(\hat{x}) - f(x^*) \geq \frac{2\sqrt{2}(1+\lambda)M}{\sqrt{K}} + 2\mathcal{C}_e\right\} \leq S\text{exp}\left\{-\lambda^2/3\right\} + 2^{-S}, \tag{3.23}$$

*where $\hat{x}$ is the output of the 2-SCGS algorithm, $x^*$ is an arbitrary optimal solution of (1.1), and $\mathcal{C}_e$ is defined in (3.6).*

*Proof.* It follows from the definition of $\hat{x}$ in (3.22) that

$$\begin{aligned}\hat{f}(\hat{x}) - f(x^*) &= \min_{s=1,\ldots,S} \hat{f}(\bar{x}_s) - f(x^*) \\ &= \min_{s=1,\ldots,S} \left\{\hat{f}(\bar{x}_s) - f(\bar{x}_s) + f(\bar{x}_s) - f(x^*)\right\} \\ &\leq \min_{s=1,\ldots,S} \left\{|\hat{f}(\bar{x}_s) - f(\bar{x}_s)| + f(\bar{x}_s) - f(x^*)\right\} \\ &\leq \max_{s=1,\ldots,S} |\hat{f}(\bar{x}_s) - f(\bar{x}_s)| + \min_{s=1,\ldots,S} \{f(\bar{x}_s) - f(x^*)\},\end{aligned}$$

which implies that

$$\begin{aligned}f(\hat{x}) - f(x^*) &= f(\hat{x}) - \hat{f}(\hat{x}) + \hat{f}(\hat{x}) - f(x^*) \\ &\leq f(\hat{x}) - \hat{f}(\hat{x}) + \max_{s=1,\ldots,S} |\hat{f}(\bar{x}_s) - f(\bar{x}_s)| + \min_{s=1,\ldots,S} \{f(\bar{x}_s) - f(x^*)\} \\ &\leq 2 \max_{s=1,\ldots,S} |\hat{f}(\bar{x}_s) - f(\bar{x}_s)| + \min_{s=1,\ldots,S} \{f(\bar{x}_s) - f(x^*)\}. \end{aligned} \tag{3.24}$$

Note that by the Markov's inequality and (3.6), we obtain

$$\text{Prob}\{f(\bar{x}_s) - f(x^*) \geq 2\mathcal{C}_e\} \leq \frac{\mathbb{E}[f(\bar{x}_s) - f(x^*)]}{2\mathcal{C}_e} \leq \frac{1}{2}, \quad \forall s = 1, \ldots, S. \tag{3.25}$$

Let $E_s$ be the event that $f(\bar{x}_s) - f(x^*) \geq 2\mathcal{C}_e$, note that due to the boundedness of $X$, and the above observation, we have

$$\text{Prob}\left\{E_s | \bigcap_{j=1}^{s-1} E_j\right\} \leq \frac{1}{2}, s = 1, \ldots, S,$$

which then implies that

$$\begin{aligned}&\text{Prob}\left\{\min_{s=1,\ldots,S} [f(\bar{x}_s) - f(x^*)] \geq 2\mathcal{C}_e\right\} \\ &= \text{Prob}\left\{\bigcap_{s=1}^{S} E_s\right\} = \prod_{s=1}^{S} \text{Prob}\left\{E_s | \bigcap_{j=1}^{s-1} E_j\right\} \leq 2^{-S}.\end{aligned} \tag{3.26}$$

By assumption (3.21) and part b) of Lemma 4 in [18] (see [25] for a general result), it is clear that

$$\text{Prob}\left\{|\sum_{j=1}^{K} [F(\bar{x}_s, \xi_j) - f(\bar{x}_s)]| \geq \sqrt{2}(1+\lambda)\sqrt{KM^2}\right\} \leq \exp\left\{-\lambda^2/3\right\}, \quad s = 1, \ldots, S,$$

which implies

$$\text{Prob}\left\{|\hat{f}(\bar{x}_s) - f(\bar{x}_s)| \geq \frac{\sqrt{2}(1+\lambda)M}{\sqrt{K}}\right\} \leq \exp\left\{-\lambda^2/3\right\}, \quad s = 1, \ldots, S.$$

Therefore, we obtain

$$\text{Prob}\left\{\max_{s=1,\ldots,S} |\hat{f}(\bar{x}_s) - f(\bar{x}_s)| \geq \frac{\sqrt{2}(1+\lambda)M}{\sqrt{K}}\right\} \leq S\exp\left\{-\lambda^2/3\right\}. \tag{3.27}$$

Our result in (3.23) directly follows from (3.24), (3.26) and (3.27). ■

Now we state a set of parameters $S$, $N$, and $K$, and the associated bounds on the number of calls to the SFO and LO oracles.

Corollary 3.6. *Suppose that parameters $\{\beta_k\}, \{\gamma_k\}$, $\{\eta_k\}$, and $\{B_k\}$ in the 2-SCGS method are set as in (3.9) for each run of SCGS algorithm. Let $\epsilon > 0$ and $\Lambda \in (0,1)$ be given, parameters $S$, $N$, and $K$ are set to*

$$S(\Lambda) := \lceil \log_2(2/\Lambda) \rceil\,, \quad N(\epsilon) := \left\lceil \sqrt{\frac{42LD_X^2}{\epsilon}} \right\rceil, \quad \textit{and} \quad K(\epsilon,\Lambda) := \left\lceil \frac{32(1+\lambda)^2M^2}{\epsilon^2} \right\rceil, \tag{3.28}$$

*where $\lambda = \sqrt{3\ln(2S/\Lambda)}$, then the total number of calls to the* SFO *and* LO *oracles performed by the 2-SCGS method in the optimization phase to compute a stochastic $(\epsilon,\Lambda)$-solution of the problem (1.1), respectively, can be bounded by*

$$\mathcal{O}\left\{\sqrt{\frac{LD_X^2}{\epsilon}}\log_2\frac{2}{\Lambda} + \frac{\sigma^2D_X^2}{\epsilon^2}\log_2\frac{2}{\Lambda}\right\} \quad \textit{and} \quad \mathcal{O}\left\{\frac{LD_X^2}{\epsilon}\log_2\frac{2}{\Lambda}\right\}. \tag{3.29}$$

*Proof.* By Corollary 3.2, we have

$$\mathcal{C}_e \le \frac{21LD_X^2}{2(N+1)^2},$$

together with the definition of $S$, $N$ and $K$ in (3.28), (3.23), and $\lambda = \sqrt{3\ln(2S/\Lambda)}$, we have

$$\text{Prob}\left\{f(\hat{x}) - f(x^*) \ge \epsilon\right\} \le \Lambda,$$

i.e. $\hat{x}$ is a stochastic $(\epsilon,\Lambda)$-solution of problem (1.1). Moreover, we obtain from Corollary 3.2 that the bounds for the number of calls to the SFO and LO oracles for each run of SCGS algorithm as (3.11), which immediately implies the bounds in (3.29), as we restart the SCGS algorithm in 2-SCGS method $S$ times. ■

**4. Generalization to saddle point problems.** In this section, we consider an important class of saddle point problems with $f$ given in the form of:

$$f(x) = \max_{y\in Y}\left\{\langle Ax, y\rangle - \hat{f}(y)\right\}, \tag{4.1}$$

where $A : \mathbb{R}^n \to \mathbb{R}^m$ denotes a linear operator, $Y \in \mathbb{R}^m$ is a convex compact set, and $\hat{f} : Y \to \mathbb{R}$ is a simple convex function. Since the objective function $f$ given in (4.1) is nonsmooth, we cannot directly apply the CGS method presented in the previous section. However, as shown by Nesterov [33], the function $f(\cdot)$ in (4.1) can be closely approximated by a class of smooth convex functions. More specifically, let $v : Y \to \mathbb{R}$ be a given strongly convex function such that

$$v(y) \ge v(x) + \langle v'(x), y - x\rangle + \frac{\sigma_v}{2}\|y - x\|^2, \forall x, y \in Y, \tag{4.2}$$

for some $\sigma_v > 0$, and let us denote $c_v := \text{argmin}_{y\in Y} v(y)$, $V(y) := v(y) - v(c_v) - \langle \nabla v(c_v), y - c_v\rangle$ and

$$\mathcal{D}_{Y,V}^2 := \max_{y\in Y} V(y). \tag{4.3}$$

It can be easily seen that

$$\|y - c_v\|^2 \le \frac{2}{\sigma_v}V(y) \le \frac{2}{\sigma_v}\mathcal{D}_{Y,V}^2, \forall y \in Y$$

and hence that

$$\|y_1 - y_2\|^2 \le \frac{4}{\sigma_v}\mathcal{D}_{Y,V}^2, \forall y_1, y_2 \in Y.$$

In view of these relations, the function $f(\cdot)$ in (4.1) can be closely approximated by

$$f_\tau(x) := \max_y \left\{ \langle Ax, y\rangle - \hat{f}(y) - \tau\,[V(y) - \mathcal{D}_{Y,V}^2] : \; y \in Y \right\}. \tag{4.4}$$

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

$$f(x) \le f_\tau(x) \le f(x) + \tau\,\mathcal{D}_{Y,V}^2, \quad \forall x \in X. \tag{4.5}$$

Moreover, Nesterov [33] shows that $f_\tau(\cdot)$ is differentiable and its gradients are Lipschitz continuous with the Lipschitz constant given by

$$\mathcal{L}_\tau := \frac{\|A\|^2}{\tau \sigma_v}. \tag{4.6}$$

In this subsection, we assume that the feasible region $Y$ and the function $\hat{f}$ are simple enough, so that the subproblem in (4.4) is easy to solve, and as a result, the major computational cost for computing the gradient of $f_\tau$ exists in the evaluation of the linear operator $A$ and its adjoint operator $A^T$. Our goal is to present a variant of the CGS method, which can achieve the optimal bounds on the number of calls to the LO oracle and the number of evaluations for the linear operator $A$ and $A^T$.

**Algorithm 6** The CGS method for solving saddle point problems

This algorithm is the same as Algorithm 1 except that (2.2) is replaces by

$$x_k = \text{CndG}(f'_{\tau_k}(z_k), x_{k-1}, \beta_k, \eta_k), \tag{4.7}$$

for some $\tau_k \ge 0$.

We now ready to describe the main convergence properties of this modified CGS method to solve the saddle point problem in (1.1)-(4.1).

Theorem 4.1. *Suppose that $\tau_1 \ge \tau_2 \ge \ldots \ge 0$. Also assume that $\{\beta_k\}$ and $\{\gamma_k\}$ satisfy (2.13) (with $L$ replaced by $L_{\tau_k}$) and (2.14). Then,*

$$f(y_k) - f(x^*) \le \frac{\beta_k \gamma_k}{2} D_X^2 + \Gamma_k \sum_{i=1}^k \frac{\gamma_i}{\Gamma_i}\left(\eta_i + \tau_i \mathcal{D}_{Y,V}^2\right), \quad \forall k \ge 1, \tag{4.8}$$

*where $x^*$ is an arbitrary optimal solution of (1.1)-(4.1). Moreover, the number of inner iterations performed at the $k$-th outer iteration can be bounded by (2.18).*

*Proof.* First, observe that by the definition of $f_\tau(\cdot)$ in (4.4), and the facts that $V(y) - \mathcal{D}_{Y,V}^2 \le 0$ and $\tau_{k-1} \ge \tau_k$, we have

$$f_{\tau_{k-1}}(x) \ge f_{\tau_k}(x) \;\; \forall x \in X, \; \forall k \ge 1. \tag{4.9}$$

Applying relation (2.21) to $f_{\tau_k}$ and using (4.9), we obtain

$$\begin{aligned} f_{\tau_k}(y_k) &\le (1-\gamma_k) f_{\tau_k}(y_{k-1}) + \gamma_k f_{\tau_k}(x) + \frac{\beta_k\gamma_k}{2}(\|x_{k-1} - x\|^2 - \|x_k - x\|^2) + \eta_k\gamma_k \\ &\le (1-\gamma_k) f_{\tau_{k-1}}(y_{k-1}) + \gamma_k\left[f(x) + \tau_k \mathcal{D}_{Y,V}^2\right] + \frac{\beta_k\gamma_k}{2}(\|x_{k-1} - x\|^2 - \|x_k - x\|^2) + \eta_k\gamma_k \end{aligned}$$

for any $x \in X$, where the second inequality follows from (4.5) and (4.9). Subtracting $f(x)$ from the both sides of the above inequality, we have

$$f_{\tau_k}(y_k) - f(x) \le (1-\gamma_k)\left[f_{\tau_{k-1}}(y_{k-1}) - f(x)\right] + \frac{\beta_k\gamma_k}{2}(\|x_{k-1}-x\|^2 - \|x_k - x\|^2) + \eta_k\gamma_k + \gamma_k\tau_k\mathcal{D}^2_{Y,V}$$

for any $x \in X$, which, in view of Lemma 2.1 and (2.23), then implies that

$$\begin{aligned} f_{\tau_k}(y_k) - f(x) &\le \Gamma_k \sum_{i=1}^{k} \frac{\beta_i\gamma_i}{2\Gamma_i}(\|x_{i-1}-x\|^2 - \|x_i - x\|^2) + \Gamma_k \sum_{i=1}^{k}\frac{\gamma_i}{\Gamma_i}\left(\eta_i + \tau_i\mathcal{D}^2_{Y,V}\right) \\ &\le \frac{\beta_k\gamma_k}{2}D_X^2 + \Gamma_k \sum_{i=1}^{k}\frac{\gamma_i}{\Gamma_i}\left(\eta_i + \tau_i\mathcal{D}^2_{Y,V}\right). \end{aligned} \tag{4.10}$$

Our result in (4.8) then immediately follows from the above relation and the fact that $f_{\tau_k}(y_k) \ge f(y_k)$ due to (4.5). The last part of our claim easily follows from Theorem 2.2.c). ■

We now provide two sets of parameters for $\{\beta_k\}, \{\gamma_k\}, \{\eta_k\}$, and $\{\tau_k\}$ which can guarantee the optimal convergence of the above variant of CGS method for saddle point optimization.

Corollary 4.2. *Assume the outer iteration limit $N \ge 1$ is given. If*

$$\tau_k \equiv \tau = \frac{2\|A\|D_X}{\mathcal{D}_{Y,V}\sqrt{\sigma_\nu}N}, \quad k \ge 1, \tag{4.11}$$

*and $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$ used in Algorithm 6 are set to*

$$\beta_k = \frac{3\mathcal{L}_{\tau_k}}{k+1}, \quad \gamma_k = \frac{3}{k+2}, \textit{and} \ \ \eta_k = \frac{\mathcal{L}_{\tau_k}D_X^2}{k^2}, \quad k \ge 1, \tag{4.12}$$

*then the number of linear operator evaluations (for $A$ and $A^T$) and the number of calls to the* LO *oracle performed by Algorithm 6 for finding an $\epsilon$-solution of problem (1.1)-(4.1), respectively, can be bounded by*

$$\mathcal{O}\left\{\frac{\|A\|D_X\mathcal{D}_{Y,V}}{\sqrt{\sigma_v}\epsilon}\right\} \quad \textit{and} \quad \mathcal{O}\left\{\frac{\|A\|^2D_X^2\mathcal{D}^2_{Y,V}}{\sigma_v\epsilon^2}\right\}. \tag{4.13}$$

*Proof.* Observe that $\Gamma_k$ is given by (2.32) due to the definition of $\gamma_k$ in (4.12). By (2.32) and (4.12), we have

$$\frac{\beta_k}{\gamma_k} = \frac{\mathcal{L}_\tau(k+2)}{k+1} \ge \mathcal{L}_\tau,$$

and

$$\frac{\beta_k\gamma_k}{\Gamma_k} = \frac{3\mathcal{L}_\tau k}{2} \ge \frac{\beta_{k-1}\gamma_{k-1}}{\Gamma_{k-1}}.$$

The above results indicate that the assumptions in Theorem 4.1 are satisfied. It then follows from Theorem 4.1, (4.11), and (4.12) that

$$\begin{aligned} f(y_N) - f(x^*) &\le \frac{9\mathcal{L}_\tau D_X^2}{2(N+1)(N+2)} + \frac{6}{N(N+1)(N+2)}\sum_{i=1}^{N}\left[\frac{\mathcal{L}_\tau D_X^2}{i^2} + \frac{2\|A\|D_X\mathcal{D}_{Y,V}}{\sqrt{\sigma_\nu}N}\right]\frac{i(i+1)}{2} \\ &\le \frac{9\|A\|D_X\mathcal{D}_{Y,V}}{4\sqrt{\sigma_\nu}(N+2)} + \frac{15\|A\|D_X\mathcal{D}_{Y,V}}{\sqrt{\sigma_\nu}N(N+1)(N+2)}\sum_{i=1}^{N} N \le \frac{69\|A\|D_X\mathcal{D}_{Y,V}}{4\sqrt{\sigma_\nu}(N+2)}, \end{aligned}$$

where the second inequality follows from the definition of $\mathcal{L}_\tau$ in (4.6). Moreover, it follows from (2.18) and (4.12) that the total number of calls to the LO oracle can be bounded by

$$\sum_{k=1}^{N} T_k \le \sum_{k=1}^{N} \left( \frac{18\mathcal{L}_{\tau_k} D_X^2}{k+1} \frac{k^2}{\mathcal{L}_{\tau_k} D_X^2} + 1 \right) \le \frac{18(N+1)N}{2} + N \le 9N^2 + 10N.$$

The bounds in (4.13) then immediately follow from the previous two conclusions. ∎

In the above result, we used a static smoothing technique as described in Nesterov [33], in which we need to fix the number of outer iterations $N$ in advance for obtaining a constant $\tau_k$ in (4.11). We now state a dynamic parameter setting for $\tau_k$ so that the number of outer iterations $N$ need not to be given a priori.

Corollary 4.3. *Suppose that parameter $\{\tau_k\}$ is now set to*

$$\tau_k = \frac{2\|A\| D_X}{\mathcal{D}_{Y,V}\sqrt{\sigma_v} k}, \quad k \ge 1, \tag{4.14}$$

*and the parameters $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$ used in Algorithm 6 are set as in (4.12). Then, the number of linear operator evaluations (for $A$ and $A^T$) and the number of calls to the* LO *oracle performed by Algorithm 6 for finding an $\epsilon$-solution of problem (1.1)-(4.1), respectively, can also be bounded by (4.13).*

*Proof.* Note that $\gamma_k$ is defined in (4.12), and hence that $\Gamma_k$ is given by (2.32). We have

$$\frac{\beta_k}{\gamma_k} \ge \mathcal{L}_{\tau_k},$$

and

$$\frac{\beta_k \gamma_k}{\Gamma_k} = \frac{3\mathcal{L}_{\tau_k} k}{2} = \frac{3\|A\|\mathcal{D}_{Y,V} k^2}{4\sqrt{\sigma_v} D_X} \ge \frac{\beta_{k-1}\gamma_{k-1}}{\Gamma_{k-1}}.$$

Therefore, the assumptions in Theorem 4.1 are satisfied. It then follows from Theorem 4.1, (4.12), and (4.14) that

$$\begin{aligned} f(y_k) - f(x^*) &\le \frac{9\mathcal{L}_{\tau_k} D_X^2}{2(k+1)(k+2)} + \frac{6}{k(k+1)(k+2)} \sum_{i=1}^{k} \left[ \frac{\mathcal{L}_{\tau_i} D_X^2}{i^2} + \frac{2\|A\| D_X \mathcal{D}_{Y,V}}{\sqrt{\sigma_v} i} \right] \frac{i(i+1)}{2} \\ &\le \frac{9\|A\| D_X \mathcal{D}_{Y,V} k}{4\sqrt{\sigma_v}(k+1)(k+2)} + \frac{15\|A\| D_X \mathcal{D}_{Y,V}}{\sqrt{\sigma_v} k(k+1)(k+2)} \sum_{i=1}^{k} i \le \frac{39\|A\| D_X \mathcal{D}_{Y,V}}{4(k+2)\sqrt{\sigma_v}}, \end{aligned}$$

where the second inequality follows from the definition of $\mathcal{L}_{\tau_k}$ in (4.6). Similarly to the proof in Corollary 4.2, we can show that the total number of calls to the LO oracle in $N$ outer iterations can be bounded by $\mathcal{O}(N^2)$. The bounds in (4.13) then immediately follow. ∎

In view of the discussions in [7], the $\mathcal{O}(1/\epsilon)$ bound on the total number of operator evaluations is not improvable for solving the saddle point problems in (1.1)-(4.1). Moreover, according to [27], the $\mathcal{O}(1/\epsilon^2)$ bound on the total number of calls to the LO is also optimal for the LCP methods for solving the saddle point problems in (1.1)-(4.1).

We now turn our attention to stochastic saddle point problems for which only stochastic gradients of $f_\tau$ are available. In particular, we consider the situation when the original objective function $f$ in (1.1) is given by

$$f(x) = \mathbb{E}\left[ \max_{y \in Y} \langle A_\xi x, y \rangle - \hat{f}(y, \xi) \right], \tag{4.15}$$

where $\hat{f}(\cdot,\xi)$ is simple concave function for all $\xi \in \Xi$ and $A_\xi$ is a random linear operator such that

$$\mathbb{E}\left[\|A_\xi\|^2\right] \leq L_A^2 \tag{4.16}$$

We can solve this stochastic saddle point problem by replacing (4.7) with

$$x_k = \mathrm{CndG}(g_k, x_{k-1}, \beta_k, \eta_k) \;\text{ where }\; g_k = \frac{1}{B_k}\sum_{j=1}^{B_k} F'(z_k, \xi_j) \tag{4.17}$$

for some $\tau_k \geq 0$ and $B_k \geq 1$. By properly specifying $\{\beta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$, and $\{B_k\}$, we can show that the number of linear operator evaluations (for $A_\xi$ and $A_\xi^T$) and the number of calls to the LO performed by this variant of CGS method for finding a stochastic $\epsilon$-solution of problem (1.1)-(4.15) can be bounded by

$$\mathcal{O}\left\{\frac{L_A^2 D_X^2 \mathcal{D}_{Y,V}^2}{\sigma_v \epsilon^2}\right\}. \tag{4.18}$$

This result can be proved by combining the techniques in Section 3 and those in Theorem 4.1. However, we skip the details of these developments for the sake of simplicity.

## 5. Numerical experiments.

Our goal in this section is to present the results from our preliminary numerical experiments. In particular, we will demonstrate the potential advantages of the basic CGS method over the original CG method, and compare CGS to CG with exact line-search (referred to as CG_LS) through two numerical experiments detailed in Subsection 5.1 and 5.2.

### 5.1. Quadratic programming problems over standard spectrahedrons.

In this experiment, we consider quadratic programming (QP) problems over a standard spectrahedron. Let $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$ be given, the QP over a standard spectrahedron is defined by

$$\min_{x\in S_n} \|\mathcal{A}x - b\|_2^2, \;\text{ where }\; S_n := \left\{x \in \mathbb{R}^{n\times n} : \mathrm{Tr}(x) = 1, x \succeq 0\right\}. \tag{5.1}$$

Table 5.1
*Randomly generated instances for QP*

| Inst. | Domain | $n$ | $m$ | $d$ | Inst. | Domain | $n$ | $m$ | $d$ |
|---|---|---|---|---|---|---|---|---|---|
| SPE11 | $S_n$ | 100 | 500 | 0.6 | SPE12 | $S_n$ | 100 | $1,000$ | 0.6 |
| SPE21 | $S_n$ | 200 | 500 | 0.4 | SPE22 | $S_n$ | 200 | $1,000$ | 0.4 |
| SPE31 | $S_n$ | 400 | 500 | 0.2 | SPE32 | $S_n$ | 400 | $1,000$ | 0.2 |

In our experiment, we use the same instances as those generated in [27]. More specifically, the linear operators $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ are sparse with entries uniformly distributed over $[0, 1]$, and the total number of nonzero entries is specified by the density parameter $d$. Because of the way the instances are generated, the optimal values of these problems are given by 0. Totally 6 instances have been generated, see Table 5.1 for more details. All of the CG and CGS algorithms in the experiment are implemented in MATLAB R2014b with initial point $y_0$ randomly generated and remaining the same for different algorithms. In our implementation, CG and CG_LS use stepsizes $\alpha_k = 2/(k + 1)$ and

$$\alpha_k = \mathrm{argmin}_{\alpha\in[0,1]}\langle f'(x_{k-1}), (1-\alpha)x_{k-1} + \alpha y_k\rangle,$$

respectively, and the latter one-dimensional optimization problem w.r.t. $\alpha$ is solved by a simple bisection method. The parameters $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$ used in the CGS method are set according to (2.33), where $N = 200$, $D_0^2 = D_X^2/2$, and $L$ is estimated using the power iteration method.

Now for each problem instance, we report in Table 5.2, the target accuracy $(f(\bar{x}) - f^*)$, the number of iterations (or gradient evaluations for CG and CGS, or calls to LO oracle for CG_LS algorithm), and the CPU

Table 5.2
*Comparisons of CG, CG_LS and CGS for QP minimization over spectrahedrons*

|  |  |  | CG |  | CG_LS |  |  | CGS |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $f(y_0)$ | Accuracy | Iterations | Time | Iterations | Time | Calls to FO | Iterations | Time | Total inner |
| SPE11 | 4.70e+1 | 1e-3 | 1257 | 23.22 | 91 | 28.47 | 1911 | 77 | 4.99 | 617 |
| SPE12 | 9.33e+1 | 1e-3 | 2491 | 81.67 | 556 | 335.23 | 11676 | 109 | 8.94 | 1024 |
| SPE21 | 1.59e+1 | 1e-3 | 894 | 51.26 | 22 | 19.79 | 462 | 105 | 11.98 | 307 |
| SPE22 | 3.29e+1 | 1e-3 | 1451 | 141.60 | 48 | 84.07 | 1008 | 127 | 23.37 | 459 |
| SPE31 | 3.26e+0 | 1e-3 | 621 | 106.99 | 10 | 21.46 | 210 | 139 | 37.95 | 215 |
| SPE32 | 6.20e+0 | 1e-3 | 896 | 232.22 | 17 | 67.48 | 357 | 200 | 81.35 | 309 |

time (in seconds, Intel Core 2 2.67GHz) required for performing these algorithms. We also record the total number of calls to the FO oracle for CG_LS, the total number of inner iterations (i.e. the total number of calls to the LO oracle) for CGS.

Let us make a few observations about the results in Table 5.2. Firstly, it is clear that CGS is more advantageous over the original CG method in terms of both CPU time and the number of iterations (gradient evaluations) to obtain the target accuracy. For example, for SPE11, CG requires $1,257$ gradient evaluations while the CGS method only requires 77 gradient evaluations. Secondly, for 4 out of 6 instances, CGS outperform CG_LS by saving up to 97% of CPU time (instance SPE 12) , while the latter algorithm is better than CGS for the last two instances. It should be noted that to obtain the results shown in Table 5.2, we did not fine tune the estimation of the Lipschitz constant used in the CGS method. Better ways (e.g., certain line-search procedures) of estimating the Lipschitz constant can further improve the numerical performance of this algorithm. It has also been observed from our numerical experiments that certain heuristic ideas, e.g., adding an lower bound on the total number of inner iterations, say $T_k \geq 3$, can also improve the practical performance of the CGS method. Using these ideas together, we were able to show that the CGS algorithm can outperform CG_LS for all these test instances.

**5.2. Matrix completion problems.** In this subsection, we consider the following matrix completion problem to recover a lower rank matrix:

$$\min \sum_{(i,j)\in\Omega} \|X_{i,j} - a_{i,j}\|^2 \quad s.t. \quad \|X\|_* \leq R. \tag{5.2}$$

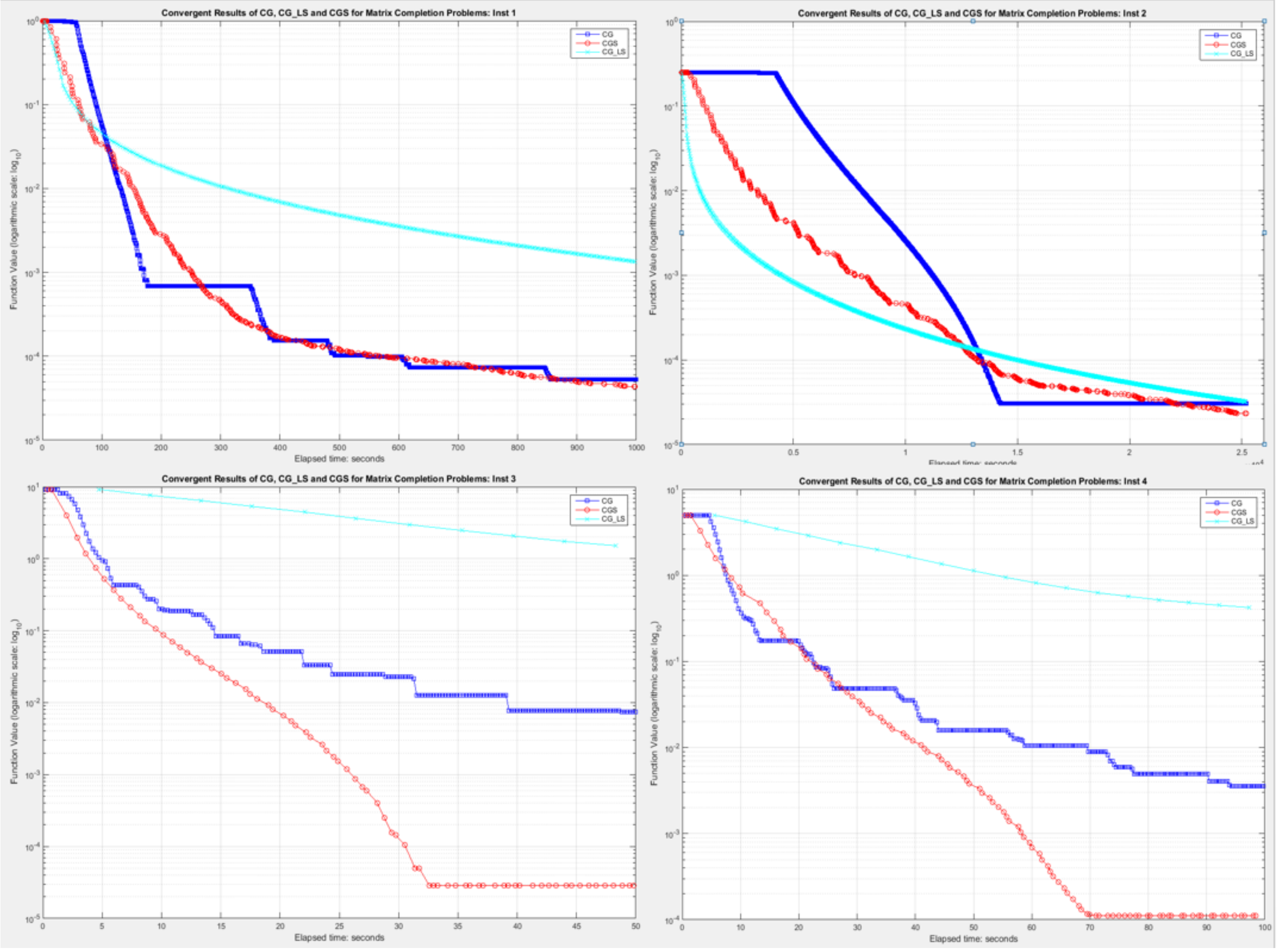
Here, $\|\cdot\|_*$ represents the nuclear norm, that is, $\|A\|_* = trace(\sqrt{A^T A}) = \sum_{i=1}^{min\{m,n\}} \sigma_i$ and $\sigma_i$ denote the singular values of $A$, $\Omega$ is a subset of entries of $A$, and $a_{i,j}$, $i,j \in \Omega$ are given, and and $R$ is a given constant. As in [6], we generate the original matrix $A$, an $m \times n$ matrix of rank $r$, by first generating an $m \times r$ factor $A_L$ and an $r \times n$ factor $A_R$ with Gaussian entries and then setting $A = A_L * A_R$. We normalize the data by dividing each entry of $A$ by $\sqrt{mn}$ and choose $R = \lceil \|A\|_* \rceil$ in our experiments. We also assume that $|\Omega| = s = \min(5r(m+n-r), \lceil 0.99mn \rceil)$ and choose these $s$ entries uniformly. Totally 4 instances have been generated in this manner (see Table 5.3). CG, CG_LS and CGS are implemented in MATLAB R2014b with the initial point $y_0$ randomly generated and remaining the same for different algorithms. Parameter settings for CG and CG_LS are the same as in Subsection 5.1, and for CGS algorithm, they are set as in (2.30). Observe that instead of the MATLAB built-in function "svds.m", we used a faster maximum singular value decomposition code by Vijayan[37].

Our numerical results are shown in Table 5.4 and Figure 5.1. In Table 5.4, we report the total iteration counts (or gradient evaluations) for all algorithms in Figure 5.1. We also record the total number of inner iterations, i.e., the total number of calls to the LO oracles for CGS, and the total number of calls to the FO oracles for CG_LS. Figure 5.1 shows how the objective function values change w.r.t. the CPU time, where the $x$-axis represents the elapsed CPU time (in seconds, Intel Core 2 2.67GHz), and the $y$-axis represents the objective function values in logarithmic scale ($log_{10}$). We use the blue, light blue, and red lines to represent CG, CG_LS, and CGS, respectively. As can be seen from Figure 5.1, the CGS method converges much faster than CG for all the tested instances. In particular, for a given CPU time, the solution accuracy obtained

Table 5.3
*Randomly generated instances for matrix completion problems*

| Inst. | $m$ | $n$ | $r$ | $R$ |
|---|---|---|---|---|
| Inst1 | $3,000$ | $1,000$ | 10 | 11 |
| Inst2 | $5,000$ | $4,000$ | 10 | 11 |
| Inst3 | $10,000$ | 100 | 10 | 10 |
| Inst4 | 100 | $20,000$ | 10 | 11 |

Fig. 5.1. *Convergence results of CGS and CG for matrix completion problems*



by the CGS method can be better than the one by CG by up to 3 orders of magnitude (see, e.g., Inst 3 at 29 seconds). Compared to CG_LS, CGS may fall behind at the beginning, but it can obtain better accuracy given enough time(see Inst 1 and 2). It should be mentioned that sometimes CG converges so slowly that the objective function values remain almost the same, which explains why in Figure 5.1 CG remains flat for Inst 2 at the beginning.

**6. Concluding remarks.** In this paper, we present a new conditional gradient type method, referred to as the CGS method for convex optimization. We show that this method can achieve the optimal complexity bounds in terms of not only the number of calls to the LO oracle, but also the number of gradient evaluations. We generalize the CGS method for solving stochastic optimization problems and show that they also exhibit

Table 5.4
*Iteration counts for Figure 5.1*

| | CG | CGS | | CG_LS | |
|---|---|---|---|---|---|
| Inst | Iterations | Iterations | Total inner | Iterations | Calls to FO |
| Inst 1 | 1711 | 384 | 1308 | 323 | 6783 |
| Inst 2 | 2188 | 579 | 1941 | 1108 | 23268 |
| Inst 3 | 206 | 77 | 249 | 12 | 252 |
| Inst 4 | 302 | 105 | 290 | 19 | 399 |

the optimal rate of convergence in terms of the number of calls to the stochastic oracle. Generalization to a special class of saddle point problems have also been presented in this paper. Some promising preliminary numerical results have also been reported for this algorithm. Extensions to the composite case, where the objective function of (1.1) contains a relatively simple nonsmooth component, can also be considered in the CGS algorithm. We leave this as an interesting topic for the future research.

It should be noted that in this paper we focus on the rate of convergence in terms of the objective values obtained by the CGS methods during a finite number of iterations. Another desirable convergence property of first-order methods is the convergence of the iterates. While it is known that one cannot establish the rate of convergence of the iterates generated by first-order methods for general smooth optimization (see Theorem 2.1.7 of [32]) unless strong convexity is assumed, we can show that the iterates generated by the standard projected-gradient method do converge asymptotically to an optimal solution for general smooth problems (see [9]). On the other hand, in the Frank-Wolfe or CGS method, we can only establish the rate of convergence of the iterates under the strong convexity assumption, or show that there exists an asymptotically convergent subsequence of the iterates to an optimal solution when such a strong convexity assumption is removed. Therefore, it will be interesting to study whether the iterates of the CGS methods (or its certain variants) converge to an optimal solution in case the objective function is not strongly convex. One possible approach would be to enforce the monotonic decreasing of the objective values generated by the CGS methods, and an alternative approach would be to add a decreasing strongly convex perturbation term to the objective function. We leave such possible extensions of the Frank-Wolfe type methods, as well as the comparison with those methods that do guarantee asymptotic iterate convergence for future research.

## 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] K. Bredies, D. Lorenz, and P. Maass. Iterated hard shrinkage for minimization problems with sparsity constraints. *SIAM Journal on Scientific Computing*, pages 657–683, 2008.

[5] K. Bredies, D. Lorenz, and P. Maass. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, pages 173–193, 2009.

[6] J. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[7] Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814, 2014.

[8] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6(4):63:1–63:30, September 2010.

[9] P. L. Combettes and W. R. Valérie. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[10] 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.

[11] J. C. Dunn. Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals. *SIAM Journal on Control and Optimization*, 17(2):674–701, 1979.

[12] J. C. Dunn. Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM Journal on Control and Optimization*, 18(5):473487, 1980.

[13] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[14] R. M. Freund and P. Grigas. New Analysis and Results for the Frank-Wolfe Method. *ArXiv e-prints*, July 2013.

[15] D. Garber and E. Hazan. A Linearly Convergent Conditional Gradient Algorithm with Applications to Online and Stochastic Optimization. *ArXiv e-prints*, Jan 2013.

[16] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, I: a generic algorithmic framework. *SIAM Journal on Optimization*, 22:1469–1492, 2012.

[17] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23:2061–2089, 2013.

[18] S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for constrained nonconvex stochastic programming. *Mathematical Programming*, 2014. to appear.

[19] Cristóbal Guzmana and A. Nemirovski. On Lower Complexity Bounds for Large-Scale Smooth Convex Optimization. *ArXiv e-prints*, January 2014.

[20] Z. Harchaoui, A. Juditsky, and A. S. Nemirovski. Conditional gradient algorithms for machine learning. NIPS OPT workshop, 2012.

[21] Elad Hazan. Sparse approximate solutions to semidefinite programs. In EduardoSany Laber, Claudson Bornstein, LoanaTito Nogueira, and Luerbio Faria, editors, *LATIN 2008: Theoretical Informatics*, volume 4957 of *Lecture Notes in Computer Science*, pages 306–316. Springer Berlin Heidelberg, 2008.

[22] M. Jaggi. *Sparse Convex Optimization Methods for Machine Learning.* PhD thesis, ETH Zürich, 2011. http://dx.doi.org/10.3929/ethz-a-007050453.

[23] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *the 30th International Conference on Machine Learning*, 2013.

[24] M. Jaggi and M. Sulovský. A simple algorithm for nuclear norm regularized problems. In *the 27th International Conference on Machine Learning*, 2010.

[25] A. Juditsky and A. S. Nemirovski. Large deviations of vector-valued martingales in 2-smooth normed spaces. Manuscript, Georgia Institute of Technology, Atlanta, GA, 2008. E-print: www2.isye.gatech.edu/∼ nemirovs/LargeDevSubmitted.pdf.

[26] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

[27] G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. Manuscript, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, June 2013. Available on http://www.optimization-online.org/.

[28] G. Lan. Gradient sliding for composite optimization. Manuscript, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, June 2014.

[29] R. Luss and M. Teboulle. Conditional gradient algorithms for rank one matrix approximations with a sparsity constraint. *SIAM Review*, 55:65–98, 2013.

[30] A. S. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization.* Wiley-Interscience Series in Discrete Mathematics. John Wiley, XV, 1983.

[31] 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.

[32] Y. E. Nesterov. *Introductory Lectures on Convex Optimization: a basic course.* Kluwer Academic Publishers, Massachusetts, 2004.

[33] Y. E. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.

[34] 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.

[35] M. Schmidt, N. L. Roux, and F. R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *Advances in Neural Information Processing Systems*, 24:1458–1466, 2011.

[36] 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.

[37] V. Vijayan. Faster svdsecon. `http://www.mathworks.com/matlabcentral`.

[38] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 3:1607–1633, 2013.