

# Backtracking linesearch for conditional gradient sliding

Hamid Nazari · Yuyuan Ouyang

Received: date / Accepted: date

**Abstract** We present a modification of the conditional gradient sliding (CGS) method that was originally developed in [9]. While the CGS method is a theoretical breakthrough in the theory of projection-free first-order methods since it is the first that reaches the theoretical performance limit, in implementation it requires the knowledge of the Lipschitz constant of the gradient of the objective function  $L$  and the number of total gradient evaluations  $N$ . Such requirements imposes difficulties in the actual implementation, not only because that it can be difficult to choose proper values of  $L$  and  $N$  that satisfies the conditions for convergence, but also since conservative choices of  $L$  and  $N$  can deteriorate the practical numerical performance of the CGS method. Our proposed method, called the conditional gradient sliding method with linesearch (CGS-ls), does not require the knowledge of either  $L$  and  $N$ , and is able to terminate early before the theoretically required number of iterations. While more practical in numerical implementation, the theoretical performance of our proposed CGS-ls method is still as good as that of the CGS method. We present numerical experiments to show the efficiency of our proposed method in practice.

**Keywords** Convex optimization · Conditional gradient sliding · Backtracking linesearch

## 1 Introduction

The problem of interest of this paper is the convex optimization problem

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

---

Hamid Nazari  
School of Mathematical and Statistical Sciences  
Clemson University  
E-mail: snazari@clemson.edu

Yuyuan Ouyang  
School of Mathematical and Statistical Sciences  
Clemson University  
E-mail: yuyuan@clemson.edu

where  $\mathcal{X} \in \mathbb{R}^n$  is a convex compact set and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a convex differentiable function such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{X}. \quad (2)$$

Here  $\|\cdot\|$  is the Euclidean norm. Our goal is to compute an  $\varepsilon$ -approximate solution  $y$  such that  $f(y) - f^* \leq \varepsilon$  using first-order information, namely, the function and gradient values  $f$  and  $\nabla f$ .

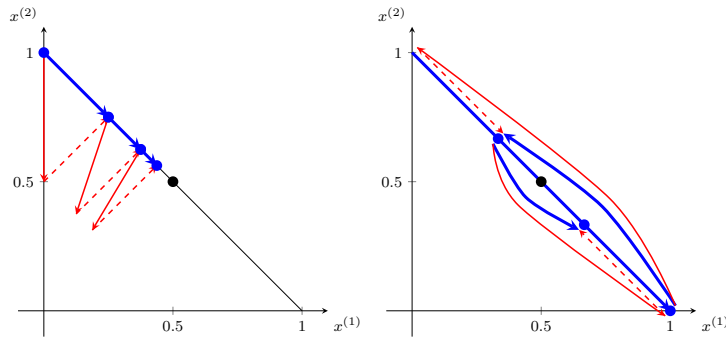
When designing numerical methods that uses first-order information, the structures of the objective function and the compact feasible set  $\mathcal{X}$  have significant impact on the theoretical convergence theory and practical computing performance. As an example, consider the performance of two classical iterative numerical methods on a simple instance of problem (1) in which  $\mathcal{X} := \{(x^{(1)}, x^{(2)}) \in \mathbb{R}^2 \mid x^{(1)} + x^{(2)} = 1, x^{(1)}, x^{(2)} \geq 0\}$  is a line segment in  $\mathbb{R}^2$  and  $f(x) = \|x\|^2/2$ . The first numerical method is the projected gradient method with iterates

$$x_k = y_{k-1} - (1/\beta_k)\nabla f(y_{k-1}), \quad y_k = \text{Proj}_{\mathcal{X}}(x_k).$$

In each iteration, the projected gradient method moves along the negative gradient direction with a pre-specified stepsize  $(1/\beta_k)$  to obtain an updated iterate  $x_k$ . Such update  $x_k$  may fall outside of the feasible region, so we maintain feasibility by projecting  $x_k$  onto the feasible set to obtain a new approximate solution  $y_k$ . The second numerical method is the conditional gradient method [2]

$$x_k = \min_{x \in \mathcal{X}} \langle \nabla f(x_{k-1}), x \rangle, \quad y_k = \sum_{i=1}^k \lambda_k^i x_i. \quad (3)$$

Here we compute an update  $x_k$  from a linear optimization over the line segment  $\mathcal{X}$ . Note that  $x_k$  will always be selected from the two extreme points  $(1, 0)^\top$  and  $(0, 1)^\top$ . To avoid oscillating between the extreme points, we select the new approximate solution  $y_k$  to a convex combination of all previous updates  $x_i$ 's. Both the aforementioned methods have been extensively studied in the literature. See, e.g., the monograph [1] or the book [7] for the survey of both methods.



**Fig. 1** Left: Project gradient method iterations. Blue arrows describe update of approximate solutions, red solid arrows describe gradient descent moves, and red dashed arrows describe projections onto  $\mathcal{X}$ . Right: Conditional gradient method iterations. Blue arrows describe update of approximate solutions, red solid arrows describe the updates  $x_k$  (always extreme points), and red dashed arrows describe the convex combination weighting process.

The performance of the projected gradient and conditional gradient methods are illustrated in Figure 1. Both methods start at point  $(0, 1)^\top$ . In the projected gradient method we select the constant stepsize

$1/\beta_k \equiv 0.5$  and in the conditional gradient method we select the weights  $\lambda_i = 2i/(k(k+1))$ . From Figure 1 we can observe that approximate solutions of both methods are gradually approaching the optimal solution. The ones produced by the projected gradient method converge faster comparing to the “back and forth” behavior (see the blue arrows in the right plot) from the conditional gradient method; this is because the projected gradient method benefits from the structure of the objective function (i.e., smoothness and strong convexity). However, in the conditional gradient method iterations we only need to solve a linear program, while the projected gradient method needs to compute projections onto  $\mathcal{X}$ . If we extend the aforementioned simple example to the general case in where  $\mathcal{X}$  is a high dimensional general convex polytope, then solving linear programs would likely be more preferable than solving projections onto general convex polytopes.

In fact, the above simple example reflects the difference between the projected gradient and conditional gradient methods in terms of their theoretical convergence theory. For our problem of interest (1), the accelerated version of projected gradient method (also known as Nesterov’s optimal method; see [13,14]) is able to compute an  $\varepsilon$ -approximate solution with  $\mathcal{O}(\sqrt{L/\varepsilon})$  gradient evaluations and projections, and the conditional gradient method will need  $\mathcal{O}(L/\varepsilon)$  gradient evaluations and linear objective optimizations [4,5]. Here  $L$  is the Lipschitz constant defined in (2). It should be noted that both convergence properties are not improvable in the following sense: for any numerical method that uses only first-order information for solving problem (1) with large number of decision variables  $n$ , it requires at least  $\mathcal{O}(\sqrt{L/\varepsilon})$  gradient computation to compute an  $\varepsilon$ -solution [12] (see also [10,14]); for any numerical method that defines a linear objective function and solve a linear objective optimization in each iteration, it requires at least  $\mathcal{O}(L/\varepsilon)$  iterations to compute an  $\varepsilon$ -solution [6] to problem (1) with large  $n$ .

The conditional gradient methods are still preferred in many practices due to its advantage of not requiring the projection computation, even though its requirement of  $\mathcal{O}(L/\varepsilon)$  gradient computations is not as good as the theoretical limit  $\mathcal{O}(\sqrt{L/\varepsilon})$ . For problems with sophisticated feasible set  $\mathcal{X}$ , the possibly expensive computational time of projection operator can significantly outweigh the theoretical advantage of the smaller  $\mathcal{O}(\sqrt{L/\varepsilon})$  gradient evaluation of any projection-based methods.

Recently, there has been a breakthrough in the theory of conditional gradient methods. A condition gradient sliding (CGS) method is proposed in [9] that is able to solve an  $\varepsilon$ -approximate solution of problem (1) with  $\mathcal{O}(\sqrt{L/\varepsilon})$  gradient evaluations and  $\mathcal{O}(L/\varepsilon)$  linear objective optimization subproblems. It should be pointed out that the CGS method still obeys the aforementioned theoretical performance limits of numerical methods that uses first-order information. In fact, it is the first conditional gradient-type method that reaches the theoretical performance limit. The key concept of the CGS method is to use the conditional gradient method to approximately solve the projection subproblem in the accelerated projected gradient method. With proper precision on solving the projection subproblem, the CGS method is able to reduce the number of gradient evaluations from the required  $\mathcal{O}(L/\varepsilon)$  of the original conditional gradient method to a improve order of  $\mathcal{O}(\sqrt{L/\varepsilon})$ .

In the original CGS method in [9], the knowledge of the Lipschitz constant  $L$  and the number of gradient evaluations  $N$  are required for implementation<sup>1</sup>. Such requirements lead to two disadvantages in practice. First, in order to make sure that a constant  $L$  satisfies the Lipschitz condition (2), we will need to choose a constant  $L$  that satisfies the Lipschitz condition (2) for all pairs  $x$  and  $y$  in  $\mathcal{X}$ . Computing such  $L$  can be difficult; the computed  $L$  can also be too conservative and lead to worse practical performance. Second, in

<sup>1</sup> Some parameter settings of the CGS method does not require  $N$ ; see, e.g., Corollary 2.3 in [9]. However, since no termination criterion is proposed in [9], to terminate we still need to specify the total number of iterations  $N$ . It should be noted that we can use some termination criterion for the CGS method, e.g., the Wolfe gap, which we will use for CGS in the numerical experiments of our paper; however, the theoretical convergence property in terms of the number of iterations needed to achieve small Wolfe gap is different from the properties of the CGS method, and may deteriorate its practical performance significantly.

order to compute an  $\varepsilon$ -approximate solution, we need to either tune the number of gradient evaluations  $N$  in practice or follow its theoretical property and specify a possibly conservative  $N = \mathcal{O}(L/\sqrt{\varepsilon})$ . While the CGS method reaches the theoretical performance limits, such disadvantages may deteriorate its practical performance significantly.

In this paper, we propose a modification of the CGS method that allows for its practical implementation. Our proposed method, called the CGS with linesearch (CGS-ls), performs a backtracking linesearch strategy to gradually increase the initial guess of Lipschitz constant  $L_0$  to values that satisfy the convergence condition. The initial guess  $L_0$  does not need to satisfy the Lipschitz condition (2) and can be significantly smaller than the actual Lipschitz constant. We also maintain the estimate of a lower bound of  $f^*$  that certifies the achievement of an  $\varepsilon$ -approximate solution. Consequently, our proposed method does not require the knowledge of either  $L$  and  $N$  and is able to stop before the theoretical  $\mathcal{O}(\sqrt{L}/\varepsilon)$  bound of required gradient evaluations or the  $\mathcal{O}(L/\varepsilon)$  bound of linear objective optimization subproblems. It should be noted that our theoretical analysis of the backtracking linesearch is non-trivial. In order to improve the practical implementation, add proper termination criterion, and maintain the same theoretical convergence properties as the CGS method, we need to modify some theoretical analysis in the original CGS results in [9]. We demonstrate through numerical experiments the advantages of our proposed CGS-ls method in implementation.

## 2 Proposed Algorithm

In this section, we describe our proposed conditional gradient sliding method with linesearch (CGS-ls) in Algorithm 1.

A few remarks are in place for the proposed CGS-ls algorithm. First, the CndG procedure is exactly the same as the one described in the CGS method in [9]. Noting the termination criterion of the CndG procedure, we can observe that the update  $x_k$  computed by the CndG procedure satisfies

$$\langle \nabla f(z_k) + \beta_k(x_k - x_{k-1}), x_k - x \rangle \leq \eta_k, \quad \forall x \in \mathcal{X}, \quad (12)$$

where  $\eta_k$  is an accuracy parameter and the  $\beta_k$  is a stepsize parameter whose values will be described in the sequel. Note that when the accuracy  $\eta_k \equiv 0$  (this is only the ideal case; in practice the CndG procedure will never terminate when  $\eta_k$  is set to 0), then  $x_k$  is the exact optimal solution to the problem

$$\min_{x \in \mathcal{X}} \langle \nabla f(z_k), x \rangle + \frac{\beta_k}{2} \|x - x_{k-1}\|^2. \quad (13)$$

In such case, the iterates (7), (8), and (9) becomes the iterates for the accelerated gradient method (see, e.g., [14]). Consequently, CGS-ls reduces to the accelerated gradient method with backtracking linesearch.

Second, when  $L_k \equiv L$  where  $L$  satisfies condition (2), then CGS-ls reduces to a version of CGS method in [9] with  $\Gamma_k = L\gamma_k^3$ . The concept behind the CGS method is to use a version of conditional gradient method to solve the possibly sophisticated projection subproblem described in (13). The theoretical performance limit is achieved through proper choice of the accuracy parameter  $\eta_k$ . Note that the convergence analysis of the choice of  $\Gamma_k$  and  $\gamma_k$  in our proposed CGS-ls method is not discussed in [9]. In the sequel, we will show that the choice  $\Gamma_k = L_k\gamma_k^3$  with a backtracking strategy for  $L_k$  yields our desired convergence result.

Third, the termination criterion (10) of the CGS-ls method is based on the linear lower bound function  $\xi_k(x)$ . In the sequel, we will prove that  $\xi_k(x) \leq f(x)$  for all  $x \in \mathcal{X}$ . Consequently, whenever the termination criterion is satisfied, we have

$$f(y_k) - f^* = f(y_k) - f(x^*) \leq f(y_k) - \xi_k(x^*) \leq f(y_k) - \min_{x \in \mathcal{X}} \xi_k(x) \leq \varepsilon \quad (14)$$

**Algorithm 1** A conditional gradient sliding algorithm with backtracking linesearch (CGS-ls)

Choose initial point  $y_0 \in \mathcal{X}$  and initial guess of Lipschitz constant  $L_0 > 0$ . Set  $x_0 = y_0$ . Define function  $\xi_0(x) \equiv 0$ .  
**for**  $k = 1, 2, \dots, N$  **do**

Find the smallest integer  $j \geq 0$  such that the estimated local Lipschitz constant  $L_k = L_{k-1} \cdot 2^j$  satisfies

$$f(y_k) \leq f(z_k) + \langle \nabla f(z_k), y_k - z_k \rangle + \frac{L_k}{2} \|y_k - z_k\|^2 + \frac{\varepsilon}{2} \gamma_k, \text{ where} \quad (4)$$

$$\gamma_k = \begin{cases} 1 & \text{if } k = 1 \\ \text{Positive solution to } L_k \gamma_k^3 = L_{k-1} (1 - \gamma_k) & \text{if } k \geq 2 \end{cases} \quad (5)$$

$$\Gamma_k = L_k \gamma_k^3 \quad (6)$$

$$z_k = (1 - \gamma_k) y_{k-1} + \gamma_k x_{k-1} \quad (7)$$

$$x_k = \text{CndG}(\nabla f(z_k), x_{k-1}, \beta_k, \eta_k) \quad (8)$$

$$y_k = (1 - \gamma_k) y_{k-1} + \gamma_k x_k \quad (9)$$

Terminate the loop if

$$f(y_k) - \min_{x \in \mathcal{X}} \xi_k(x) \leq \varepsilon \text{ where function } \xi_k(\cdot) \text{ is defined by } \xi_k(x) := (1 - \gamma_k) \xi_{k-1}(x) + \gamma_k (f(z_k) + \langle \nabla f(z_k), x - z_k \rangle). \quad (10)$$

**end for**

Output approximate solution  $y_k$  at the termination of the above for-loop.

**procedure**  $u^+ = \text{CndG}(g, u, \beta, \eta)$

1. Set  $u_1 = u$  and  $t = 1$ .

2. Let  $v_t$  be the optimal solution for the subproblem of

$$V_{g,u,\beta}(u_t) := \max_{x \in \mathcal{X}} \langle g + \beta(u_t - u), u_t - x \rangle \quad (11)$$

3. If  $V_{g,u,\beta}(u_t) \leq \eta$ , set  $u^+ = u_t$  and terminate the procedure.

4. Set  $u_{t+1} = (1 - \alpha_t) u_t + \alpha_t v_t$  where  $\alpha_t = \min \{1, \langle \beta(u - u_t) - g, v_t - u_t \rangle / (\beta \|v_t - u_t\|^2)\}$

5. Set  $t \leftarrow t + 1$  and go to step 2.

**end procedure**

where  $x^*$  is an optimal solution to problem (1). The above relation certifies that  $y_k$  is an approximate solution to problem (1). Such certification strategy for approximate solutions is previously discussed in [11] for convex optimization problems and implemented in accelerated gradient methods (see, e.g., [15]). In the sequel, we will prove that the termination criterion is satisfied with at most  $\mathcal{O}(\sqrt{L/\varepsilon})$  gradient evaluations. One alternative termination criterion is the Wolfe gap

$$\max_{x \in \mathcal{X}} \langle \nabla f(y_k), y_k - x \rangle \leq \varepsilon$$

which is widely employed in the literature of conditional gradient methods. When the Wolfe gap termination criterion is satisfied, we also have  $f(y_k) - f^* \leq \varepsilon$  due to the convexity of  $f$ :

$$f(y_k) - f^* = f(y_k) - f(x^*) \leq -\langle \nabla f(y_k), x^* - y_k \rangle \leq \max_{x \in \mathcal{X}} \langle \nabla f(y_k), y_k - x \rangle \leq \varepsilon. \quad (15)$$

However, one can only show that the termination criterion through Wolfe gap is satisfied with significantly worse number  $\mathcal{O}(L/\varepsilon)$  of gradient evaluations (see, e.g., [5]).

Fourth, it is necessary to point out the backtracking strategy we implement in the proposed CGS-ls method. During implementation, we compute the estimated  $L_k$  in the following way: we start with  $L_k = L_{k-1}$

and compute  $\gamma_k$ ,  $\Gamma_k$ ,  $z_k$ ,  $x_k$ , and  $y_k$  in (5), (6), (7), (8), and (9). After the computation, we verify whether condition (4) is satisfied. If not, we will multiply  $L_k$  by 2 and backtrack all the values again. Such backtracking procedure stops when the Lipschitz condition (4) is satisfied. It should be noted that we can derive from the convexity of  $f$  and the Lipschitz condition (2) that

$$f(y) \leq f(z) + \langle \nabla f(z), y - z \rangle + \frac{L}{2} \|y - z\|^2, \quad \forall y, z \in \mathcal{X}.$$

Therefore, if  $L_0 \geq L_{min}$ , where  $L_{min}$  is the smallest Lipschitz constant that satisfies the Lipschitz condition (2), then we have  $L_k \equiv L_0 \geq L_{min}$ , and CGS-ls reduces to CGS with Lipschitz constant  $L_0$  and parameter  $\Gamma_k = L_0 \gamma_k^3$ . If  $L_0 < L_{min}$ , it is straightforward to observe that the number of backtracking required throughout the entire iterates of the CGS-ls method is  $\lceil \log(2L_{min}/L_0) \rceil$ . This is because that whenever  $L_k \geq 2L_{min}$ , then the condition (4) is always satisfied. Summarizing the above description of  $L_k$ 's and accounting for both the cases  $L_0 \geq L_{min}$  and  $L_0 < L_{min}$ , we have the following relation:

$$L_0 \leq L_1 \leq \dots \leq L_k \leq \max\{2L_{min}, L_0\}. \quad (16)$$

Finally, in order to compute  $\gamma_k$  when  $k \geq 2$  it suffices to solve the positive root to a cubic polynomial equation  $L_k \gamma_k^3 = \Gamma_{k-1}(1 - \gamma_k)$ . It is easy to verify that

$$\gamma_k = \sqrt[3]{\frac{\Gamma_{k-1}}{2L_k} + \frac{\Gamma_{k-1}}{L_k} \sqrt{\frac{1}{4} + \frac{\Gamma_{k-1}}{27L_k}}} + \sqrt[3]{\frac{\Gamma_{k-1}}{2L_k} - \frac{\Gamma_{k-1}}{L_k} \sqrt{\frac{1}{4} + \frac{\Gamma_{k-1}}{27L_k}}} \in (0, 1), \quad \forall k \geq 2. \quad (17)$$

is the unique positive real root we are looking for through the cubic formula. Here, to prove that  $\gamma_k \in (0, 1)$  for all  $k \geq 2$ , note that when  $\Gamma_{k-1}/L_k \in (0, 1)$ , from the above cubic formula description of  $\gamma_k$  we have  $\gamma_k > 0$ . Also, applying  $\Gamma_{k-1}/L_k \in (0, 1)$  to the relation  $L_k \gamma_k^3 = \Gamma_{k-1}(1 - \gamma_k)$  we have  $1 - \gamma_k > \gamma_k^3 > 0$ . Therefore  $\Gamma_{k-1}/L_k \in (0, 1)$  leads to  $\gamma_k \in (0, 1)$ . Moreover, noting from the description of  $L_k$  in Algorithm 1 we have  $L_k \geq L_{k-1}$ , hence  $\gamma_k \in (0, 1)$  implies that  $\Gamma_k/L_{k+1} \in (0, 1)$ . Therefore, applying induction we can prove that  $\gamma_k \in (0, 1)$  for all  $k \geq 2$ . Also, note that  $\gamma_1 = 1$  in (5). Consequently,  $y_k$  in (9) is the convex combination of  $y_{k-1}$  and  $x_k$ ; noting that  $x_k \in \mathcal{X}$  and that  $y_0 = x_0 \in \mathcal{X}$  we can have that the approximation solution  $y_k \in \mathcal{X}$ .

### 3 Theoretical analysis

In this section we perform the convergence analysis of the proposed CGS-ls algorithm in Algorithm 1. We begin with two technical results that will be used in the analysis.

**Lemma 1** *Suppose that  $\{\lambda_i\}_{i \geq 1}$  and  $\{a_i\}_{i \geq 0}$  are two sequences of nonnegative real numbers, in which the sequence  $\{\lambda_i\}_{i \geq 1}$  is non-decreasing. For any fixed  $k$ , we have*

$$\sum_{i=1}^k \lambda_i (a_{i-1} - a_i) \leq \lambda_k \max_{0 \leq t \leq k} a_t.$$

*Proof.* Since  $\lambda_i \geq \lambda_{i-1} \geq 0$  for all  $i = 2, 3, \dots$ , we have immediately that

$$\sum_{i=1}^k \lambda_i (a_{i-1} - a_i) = \lambda_1 a_0 + \sum_{i=2}^k (\lambda_i - \lambda_{i-1}) a_{i-1} - \lambda_k a_k \leq \lambda_1 \max_{0 \leq t \leq k} a_t + \sum_{i=2}^k (\lambda_i - \lambda_{i-1}) \max_{0 \leq t \leq k} a_t = \lambda_k \max_{0 \leq t \leq k} a_t. \quad \square$$

**Lemma 2** *In Algorithm 1, suppose that  $\gamma_1 = 1$ ,  $\gamma_k \in (0, 1)$ ,  $k = 2, 3, \dots$ , and the value of  $\Gamma_k$  satisfies*

$$\Gamma_k := \begin{cases} 1, & k = 1, \\ \Gamma_{k-1}(1 - \gamma_k), & k \geq 2. \end{cases} \quad (18)$$

*If the sequence  $\{\delta_k\}_{k \geq 1}$  satisfies*

$$\delta_k \leq (1 - \gamma_k)\delta_{k-1} + B_k, \quad k = 1, 2, \dots, \quad (19)$$

*then for any  $k \geq 1$  we have*

$$\delta_k \leq \Gamma_k \sum_{i=1}^k \frac{B_i}{\Gamma_i}.$$

*In particular, the above inequality becomes equality when the relations in (19) are all equality relations.*

*Proof.* The result follows from dividing both sides of (19) by  $\Gamma_k$  and then summing up the resulting inequalities or equalities.  $\square$

A few remarks are in place regarding the above lemma. First, by the descriptions of  $\gamma_k$  and  $\Gamma_k$  in Algorithm 1, the condition (18) is clearly satisfied. Second, applying the above lemma with  $\delta_k \equiv 1$  and  $B_k = \gamma_k$  we have the following equality:

$$\Gamma_k \sum_{i=1}^k \frac{\gamma_i}{\Gamma_i} = 1. \quad (20)$$

Similarly, applying the above lemma to the definition of the lower bound function  $\xi_k(\cdot)$  in (10) (with  $\delta_k = \xi_k(x)$  and  $B_k = \gamma_k(f(z_k) + \langle \nabla f(z_k), x - z_k \rangle)$ ) we also have

$$\xi_k(x) = \Gamma_k \sum_{i=1}^k \frac{\gamma_i}{\Gamma_i} (f(z_i) + \langle \nabla f(z_i), x - z_i \rangle). \quad (21)$$

We are now ready to analyze the convergence of the proposed CGS-ls algorithm. Theorem 1 below describes the main convergence property of Algorithm 1.

**Theorem 1** *Suppose that the parameters in Algorithm 1 satisfy  $\beta_k \geq L_k \gamma_k$  for all  $k$ . Then we have*

$$f(y_k) - \xi_k(x) \leq \frac{\varepsilon}{2} + \Gamma_k \sum_{i=1}^k \frac{\gamma_i \beta_i}{2\Gamma_i} \left( \|x - x_{i-1}\|^2 - \|x - x_i\|^2 \right) + \Gamma_k \sum_{i=1}^k \frac{\gamma_i \eta_i}{\Gamma_i}, \quad \forall x \in \mathcal{X}.$$

*Proof.* Let us fix any  $x \in \mathcal{X}$ . In order to prove the result, we will estimate a lower bound of  $\xi_k(x)$ . From the description of  $y_k$  in (9) we observe that  $\gamma_k(x_k - z_k) = (y_k - z_k) - (1 - \gamma_k)(y_{k-1} - z_k)$ . Applying such observation to the description of  $\xi_k(x)$  in (21) we have

$$\begin{aligned} \frac{1}{\Gamma_k} \xi_k(x) &= \sum_{i=1}^k \frac{1}{\Gamma_i} [\gamma_i f(z_i) + \gamma_i \langle \nabla f(z_i), x - x_i \rangle + \langle \nabla f(z_i), \gamma_i(x_i - z_i) \rangle] \\ &= \sum_{i=1}^k \frac{1}{\Gamma_i} [f(z_i) + \langle \nabla f(z_i), y_i - z_i \rangle - (1 - \gamma_i)(f(z_i) + \langle \nabla f(z_i), y_{i-1} - z_i \rangle) + \gamma_i \langle \nabla f(z_i), x - x_i \rangle]. \end{aligned}$$

We make three observations in the above equation. First, by (4) we have

$$f(z_i) + \langle \nabla f(z_i), y_i - z_i \rangle \geq f(y_i) - \frac{L_i}{2} \|y_i - z_i\|^2 - \frac{\varepsilon}{2} \gamma_i = f(y_i) - \frac{L_i \gamma_i^2}{2} \|x_i - x_{i-1}\|^2 - \frac{\varepsilon}{2} \gamma_i.$$

Here the last equality is from the descriptions of  $z_k$  and  $y_k$  in (7) and (9) respectively. Second, by the convexity of  $f$  we have

$$-(f(z_i) + \langle \nabla f(z_i), y_{i-1} - z_i \rangle) \geq -f(y_{i-1}).$$

Third, by the stopping criterion of the CndG procedure in (12) and our assumption that  $\beta_k \geq L_k \gamma_k$  for all  $k$ , we have

$$\begin{aligned} \gamma_i \langle \nabla f(z_i), x - x_i \rangle &\geq \gamma_i \beta_i \langle x_i - x_{i-1}, x_i - x \rangle - \gamma_i \eta_i \\ &= -\frac{\gamma_i \beta_i}{2} \left( \|x - x_{i-1}\|^2 - \|x_i - x_{i-1}\|^2 - \|x - x_i\|^2 \right) - \gamma_i \eta_i \\ &\geq -\frac{\gamma_i \beta_i}{2} \left( \|x - x_{i-1}\|^2 - \|x - x_i\|^2 \right) - \gamma_i \eta_i + \frac{L_i \gamma_i^2}{2} \|x_i - x_{i-1}\|^2. \end{aligned}$$

Applying the above three observations and recalling that  $\gamma_1 = 1$  and  $\gamma_k \in (0, 1)$  for all  $k \geq 2$  in (5) and (17) respectively, we obtain that

$$\frac{1}{\Gamma_k} \xi_k(x) \geq \sum_{i=1}^k \frac{1}{\Gamma_i} \left[ f(y_i) - (1 - \gamma_i) f(y_{i-1}) - \frac{\gamma_i \beta_i}{2} \left( \|x - x_{i-1}\|^2 - \|x - x_i\|^2 \right) - \frac{\varepsilon}{2} \gamma_i - \gamma_i \eta_i \right].$$

In the above result, noting from the relation (18) between  $\gamma_k$  and  $\Gamma_k$  and the fact that  $\gamma_1 = 1$ , we have

$$\sum_{i=1}^k \frac{1}{\Gamma_i} f(y_i) - \frac{1 - \gamma_i}{\Gamma_i} f(y_{i-1}) = \frac{f(y_k)}{\Gamma_k}.$$

We conclude the theorem immediately by combining the above two equations and using the relation (20).  $\square$

**Corollary 1** *Suppose that the parameters of Algorithm 1 are set to*

$$\beta_k = L_k \gamma_k \text{ and } \eta_k = \frac{L_k \gamma_k D^2}{k}, \quad (22)$$

where  $D$  is any constant that estimates the diameter  $D_{\mathcal{X}} := \max_{x, y \in \mathcal{X}} \|x - y\|$  of  $\mathcal{X}$ . Then Algorithm 1 terminates with an  $\varepsilon$ -approximate solution  $y_k$  after  $k \geq N_{\text{grad}}$  gradient evaluations, in which

$$N_{\text{grad}} := C \sqrt{\frac{\max\{2L_{\min}, L_0\} D_{\mathcal{X}}^2}{\varepsilon}}, \text{ where } \sqrt{\frac{27}{2} + \frac{27D^2}{D_{\mathcal{X}}^2}} \sqrt{\frac{\max\{2L_{\min}, L_0\}}{L_0}}. \quad (23)$$

At termination, the total number of linear objective optimization (the problem in (11)) is bounded by

$$N_{\text{lin}} := \frac{6D_{\mathcal{X}}^2 C^2 \max\{2L_{\min}, L_0\} D_{\mathcal{X}}}{D^2 \varepsilon} + C \sqrt{\frac{\max\{2L_{\min}, L_0\} D_{\mathcal{X}}^2}{\varepsilon}}. \quad (24)$$

Here  $L_{\min}$  is the smallest Lipschitz constant that satisfies the Lipschitz condition (2) of the gradient  $\nabla f$ .



*Proof.* Applying Theorem 1 with the parameters described in (22), and noting from the description of  $\gamma_k$  and  $\Gamma_k$  in Algorithm 1 that  $\Gamma_k = L_k \gamma_k^3 = (1 - \gamma_k) \Gamma_{k-1}$ , we have

$$f(y_k) - \xi_k(x) \leq \frac{\varepsilon}{2} + \Gamma_k \sum_{i=1}^k \frac{1}{2\gamma_i} \left( \|x - x_{i-1}\|^2 - \|x - x_i\|^2 \right) + \Gamma_k \sum_{i=1}^k \frac{D^2}{i\gamma_i}, \quad \forall x \in \mathcal{X}$$

Since  $\gamma_k \in (0, 1)$  for all  $k \geq 2$  (see (17)), we observe that  $L_k \gamma_k^3 = \Gamma_k = (1 - \gamma_k) \Gamma_{k-1} < \Gamma_{k-1} = L_{k-1} \gamma_{k-1}^3$  for all  $k \geq 2$ . Using this observation and noting from (16) that  $L_k \geq L_{k-1}$ , we have  $\gamma_k^3 < \gamma_{k-1}^3$ . Consequently, the sequence  $\{1/\gamma_i\}_{i \geq 1}$  at the right hand side of the above estimate of  $f(y_k) - \xi_k(x)$  is an increasing sequence. Applying Lemma 1 we have

$$f(y_k) - \xi_k(x) \leq \frac{\varepsilon}{2} + \frac{\Gamma_k}{2\gamma_k} \max_{0 \leq i \leq k} \|x - x_i\|^2 + \Gamma_k \sum_{i=1}^k \frac{D^2}{i\gamma_i} \leq \frac{\varepsilon}{2} + \frac{\Gamma_k}{2\gamma_k} D_{\mathcal{X}}^2 + \Gamma_k \sum_{i=1}^k \frac{D^2}{i\gamma_i}. \quad (25)$$

Here in the last inequality we use the definition of diameter  $D_{\mathcal{X}}$ . We will estimate the right most side of the above relation.

Using the relation  $\Gamma_k = L_k \gamma_k^3 = (1 - \gamma_k) \Gamma_{k-1}$  again, we have

$$\sqrt[3]{\frac{1}{\Gamma_k}} - \sqrt[3]{\frac{1}{\Gamma_{k-1}}} = \frac{\frac{1}{\Gamma_k} - \frac{1}{\Gamma_{k-1}}}{\sqrt[3]{\frac{1}{\Gamma_k^2}} + \sqrt[3]{\frac{1}{\Gamma_k \Gamma_{k-1}}} + \sqrt[3]{\frac{1}{\Gamma_{k-1}^2}}} = \frac{\frac{\gamma_k}{\Gamma_k}}{\sqrt[3]{\frac{1}{\Gamma_k^2}} + \sqrt[3]{\frac{1}{\Gamma_k \Gamma_{k-1}}} + \sqrt[3]{\frac{1}{\Gamma_{k-1}^2}}}.$$

Noting that  $\gamma_k \in (0, 1)$  (see (17)) and recalling that  $\Gamma_k = (1 - \gamma_k) \Gamma_{k-1}$  we have  $\Gamma_k \leq \Gamma_{k-1}$ . Therefore, we have

$$\sqrt[3]{\frac{1}{\Gamma_k^2}} \leq \sqrt[3]{\frac{1}{\Gamma_k^2}} + \sqrt[3]{\frac{1}{\Gamma_k \Gamma_{k-1}}} + \sqrt[3]{\frac{1}{\Gamma_{k-1}^2}} \leq 3 \sqrt[3]{\frac{1}{\Gamma_k^2}}$$

Recalling that  $\Gamma_k = L_k \gamma_k^3$ , the above two relations imply that

$$\frac{1}{\sqrt[3]{L_k}} \geq \sqrt[3]{\frac{1}{\Gamma_k}} - \sqrt[3]{\frac{1}{\Gamma_{k-1}}} \geq \frac{1}{3 \sqrt[3]{L_k}}.$$

Here, recalling the relations of  $L_k$ 's in (16), we have

$$\frac{1}{\sqrt[3]{L_0}} \geq \sqrt[3]{\frac{1}{\Gamma_k}} - \sqrt[3]{\frac{1}{\Gamma_{k-1}}} \geq \frac{1}{3 \sqrt[3]{\max\{2L_{min}, L_0\}}}.$$

Summing the above relation from 1 to  $k$  we obtain that

$$\frac{k-1}{\sqrt[3]{L_0}} \geq \frac{1}{\sqrt[3]{\Gamma_k}} - \frac{1}{\sqrt[3]{\Gamma_1}} \geq \frac{k-1}{3 \sqrt[3]{\max\{2L_{min}, L_0\}}}.$$

Recalling from (16) that  $L_0 \leq L_1 \leq \max\{2L_{min}, L_0\}$  and noting from (5) and (6) that  $\Gamma_1 = L_1$ , the above becomes

$$\frac{k}{\sqrt[3]{L_0}} \geq \frac{1}{\sqrt[3]{\Gamma_k}} \geq \frac{k-1}{3 \sqrt[3]{\max\{2L_{min}, L_0\}}} + \frac{1}{\sqrt[3]{\max\{2L_{min}, L_0\}}} > \frac{k}{3 \sqrt[3]{\max\{2L_{min}, L_0\}}},$$

i.e.,

$$\frac{L_0}{k^3} \leq \Gamma_k \leq \frac{27 \max\{2L_{min}, L_0\}}{k^3}.$$

Using the first inequality above and recalling the relations  $\Gamma_k = L_k \gamma_k^3$  and  $L_0 \leq L_k$  we have

$$\gamma_k \geq \sqrt[3]{\frac{L_0}{L_k k^3}} \geq \frac{1}{k} \sqrt[3]{\frac{L_0}{\max\{2L_{min}, L_0\}}}.$$

Applying the above two results to (25), we conclude that

$$\begin{aligned} & f(y_k) - \xi_k(x) \\ & \leq \frac{\varepsilon}{2} + \frac{27 \max\{2L_{min}, L_0\} D_{\mathcal{X}}^2}{2k^2} \cdot \sqrt[3]{\frac{\max\{2L_{min}, L_0\}}{L_0}} + \frac{27 \max\{2L_{min}, L_0\} D^2}{k^3} \sum_{i=1}^k \sqrt[3]{\frac{\max\{2L_{min}, L_0\}}{L_0}} \\ & = \frac{\varepsilon}{2} + \frac{\max\{2L_{min}, L_0\} D_{\mathcal{X}}^2}{k^2} \left[ \frac{27}{2} + \frac{27 D^2}{D_{\mathcal{X}}^2} \right] \sqrt[3]{\frac{\max\{2L_{min}, L_0\}}{L_0}}, \quad \forall x \in \mathcal{X}. \end{aligned}$$

Noting the above result and (14), we conclude that the proposed CGS-ls algorithm will terminate with an  $\varepsilon$ -approximate solution after  $k \geq N_{grad}$  iterations, where  $N_{grad}$  is defined in (23). Also, by Theorem 2.2(c) in [9] and our parameter setting (22), the total number of linear objective optimization that is performed in the  $k$ -th call to the CndG procedure is bounded by

$$T_k := \left\lceil \frac{6\beta_k D_{\mathcal{X}}^2}{\eta_k} \right\rceil = \left\lceil \frac{6D_{\mathcal{X}}^2}{D^2} k \right\rceil$$

Therefore, at termination the total number of linear objective optimization that is performed by Algorithm 1 is bounded by

$$N_{lin} := \sum_{i=1}^{N_{grad}} T_i \leq \sum_{i=1}^{N_{grad}} T_i \left( \frac{6D_{\mathcal{X}}^2}{D^2} i + 1 \right) = \frac{6D_{\mathcal{X}}^2}{D^2} N_{grad}^2 + N_{grad}.$$

Substituting the value of  $N_{grad}$  in (23) we obtain (24).  $\square$

A few remarks are in place for the above corollary. First, from (23) and (24) we conclude that the proposed CGS-ls method has the same theoretical convergence property as that of the CGS method in [9]. Specifically, to compute an  $\varepsilon$ -approximate solution, the CGS-ls method reaches the theoretical performance limit by requiring at most  $\mathcal{O}(\sqrt{L/\varepsilon})$  gradient evaluations and  $\mathcal{O}(L/\varepsilon)$  linear objective optimizations, where  $L := \max\{2L_{min}, L_0\}$  is a Lipschitz constant that satisfies the Lipschitz condition (2). Second, in our parameter setting (22) we need to choose an estimate  $D$  for the exact diameter  $D_{\mathcal{X}}$ . However, as long as  $D$  is relatively close to  $D_{\mathcal{X}}$  (e.g., smaller or larger than  $D_{\mathcal{X}}$  but within an order of  $\mathcal{O}(1/\sqrt{\varepsilon})$ ), our convergence properties will not be affected. Finally, when the initial guess of Lipschitz constant  $L_0$  is larger than  $L_{min}$ , no backtracking linesearch will be performed, and the proposed CGS-ls method becomes a version of the CGS method in [9] that is equipped with a proper termination criterion. However, unlike the CGS method, when the initial guess  $L_0$  is significantly smaller than  $L_{min}$ , the convergence property of the CGS-ls method will not be affected significantly. As an example, if we choose  $L_0 = 0.001L_{min}$ , then in the convergence result of the

above corollary we have  $\sqrt[6]{\max\{2L_{min}, L_0\}/L_0} = \sqrt[6]{2000} \approx 3.5$ , namely, the number of gradient evaluations will be enlarged by a constant of approximately 3.5. In terms of theoretical convergence, such enlargement will not change the order of the gradient evaluations in terms of its dependence on  $1/\varepsilon$ . However, in terms of numerical implementation, we have the flexibility of choosing much smaller choice of  $L_0$ . Moreover, if the smaller choice of  $L_0$  is satisfied along the iterates of the CGS-ls method, then we can expect that the practical performance of CGS-ls is much faster than algorithms that use a conservative choice of global Lipschitz constant estimate.

#### 4 Numerical results

In this section we present the results from our numerical experiments. We will compare the performance of the proposed CGS-ls method with that of the conditional gradient (CG) method (described in (3) with weights  $\lambda_i = 2i/(k(k+1))$ ) and the CGS method in [9] (parameters follow Corollary 2.3 in the paper). We consider two quadratic optimization problems with different subsets; the first is over the standard spectrahedron and the second is over the convex hull of all Hamiltonian cycles. All numerical experiments are performed on a compute with Intel Core i5 2.7 GHz CPU.

In the first numerical experiment, we consider the optimization problem over the standard spectrahedron:

$$\min_{X \in \text{Spe}_n} f(X) := \frac{1}{2} \|\mathcal{A}X - B\|_F^2 \quad \text{where} \quad \text{Spe}_n := \{X \in \mathbb{R}^{n \times n} : \text{Tr}(X) = 1, X \succeq 0\}.$$

Here  $\mathcal{A} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^m$  is a linear operator,  $\|\cdot\|_F$  is the Frobenius norm, and the feasible set  $\text{Spe}_n$  is a standard spectrahedron. Note that the linear objective optimization over the spectrahedron can be solved by computing a maximum eigenvalue problem (see, e.g., [4]). For each random instance in this numerical experiment, we generate  $\mathcal{A}$  first by equivalently generating a  $m \times n^2$  matrix with 20%, 60% or 80% nonzero entries that follow i.i.d. standard normal distribution. We then generate  $B = \mathcal{A}U\Sigma U^\top$  where  $U$  is a random orthogonal matrix and  $\Sigma$  is a diagonal matrix with uniformly random entries between 0 and 1 (normalized afterwards so that they sum to 1). Therefore, the optimal value of all generated instances are 0.

In the second numerical experiment, we consider the optimization problem over the convex hull of Hamiltonian cycles in [8]:

$$\min_{x \in \mathcal{H}} f(x) := \frac{1}{2} \|Ax - b\|_2^2 \quad \text{where} \quad \mathcal{H} = \text{conv}\{x \in \mathbb{R}^{n(n-1)/2} : x \text{ is a Hamiltonian cycle}\}.$$

Here  $A : \mathbb{R}^{m \times [n(n-1)/2]}$ ,  $b \in \mathbb{R}^m$ , and the feasible set  $\mathcal{H}$  is the convex hull of all Hamiltonian cycles in a complete graph with  $n$  nodes. We describe any Hamiltonian cycle through a vector of dimension  $n(n-1)/2$  (the lower triangular part of the adjacency matrix). Note that the linear objective optimization over  $\mathcal{H}$  can be solved by computing the solution to a traveling salesman problem (solved through Gurobi [3]). For each random instance,  $A$  is randomly generated with 60% nonzero entries that follow i.i.d. standard uniform distribution. We then generate  $b = A(0.8v_1 + 0.2v_2)$  where  $v_1$  and  $v_2$  are two Hamiltonian cycles that are generated from random permutations of all nodes. The optimal value of all generated instances are 0.

We report the performance of CG, CGS, and CGS-ls in the above two numerical experiments in Table 1 and Table 2 respectively. For CG and CGS, we terminate when the Wolfe gap described in (15) is smaller than 0.01; for CGS-ls, we terminate when the gap describe in (10) is smaller than 0.01. Consequently, all algorithms terminate either when an approximate solution is certified with  $\varepsilon = 0.01$  (or when the algorithm runs over 30 minutes). Note that CGS requires the Lipschitz constant of the objective function; we compute

**Table 1** Comparison of CG, CGS and CGS-ls on the first numerical experiment (minimization over standard spectrahedron).

Instance info.			CG			CGS				CGS-ls			
m	n	density	iter.	time	obj.	outer	inner	time	obj.	outer	inner	time	obj.
1000	100	.2	21908	543	3e-6	264	528	10	3e-6	148	919	13	6e-8
2000	100	.2	38861	1427	3e-6	842	1871	38	4e-6	232	1961	26	8e-8
3000	100	.2	36402	1800	5e-6	900	2368	53	8e-6	219	2175	26	2e-7
1000	100	.6	45170	1800	1e-6	379	758	18	1e-6	307	1540	24	2e-8
2000	100	.6	26855	1800	1e-5	1309	2964	91	2e-6	343	2578	35	2e-8
3000	100	.6	18830	1800	5e-5	1410	3811	138	3e-6	291	2797	38	1e-7
1000	100	.8	39448	1800	1e-6	393	786	21	1e-6	328	1625	27	1e-8
2000	100	.8	22728	1800	1e-5	1475	3361	117	1e-6	446	2704	42	3e-8
3000	100	.8	16055	1800	8e-5	1551	4223	170	3e-6	320	3360	45	5e-8

**Table 2** Comparison of CG, CGS and CGS-ls on the second numerical experiment (minimization over the convex hull of Hamiltonian cycles).

Instance info.			CG			CGS				CGS-ls			
m	n	density	iter.	time	obj.	outer	inner	time	obj.	outer	inner	time	obj.
1e3	20	.6	37259	596	3e-6	6246	12492	197	4e-3	1108	2493	39	4e-4
1e3	25	.6	48415	968	1e-5	9236	18472	502	4e-3	1526	3730	91	5e-4
1e3	30	.6	56254	1800	1e-5	12542	25084	735	4e-3	1883	4734	142	4e-4
1e3	35	.6	45355	1800	2e-5	15390	30780	1323	5e-3	2210	5636	233	4e-4
1e3	40	.6	27638	1800	1e-4	14445	28890	1800	9e-3	2511	6335	374	4e-4
1e4	15	.6	92429	1800	4e-6	11927	23854	473	5e-3	2589	6711	117	7e-4
1e4	20	.6	67756	1800	1e-5	19508	39016	841	5e-3	3535	8702	180	3e-4
1e4	25	.6	52912	1800	8e-5	23044	46088	1800	7e-3	4529	11964	364	3e-4
1e4	30	.6	31736	1800	2e-4	19272	38544	1800	2e-2	5564	15378	623	2e-4
1e4	35	.6	24348	1800	6e-4	14771	29542	1800	5e-2	6480	17848	894	2e-4
1e5	15	.6	18433	1800	1e-3	14924	29848	1800	3e-2	8036	15825	774	1e-3
1e5	17	.6	14281	1800	5e-3	12791	25582	1800	7e-2	8659	18406	943	5e-4
1e5	20	.6	9715	1800	3e-3	9019	18038	1800	2e-1	9277	18801	1427	3e-4

them through the maximum eigenvalue of the Hessian of the objective function (the time for computing maximum eigenvalue is not counted towards CGS' computation time). For CGS-ls, we set  $L_0 = 10$  for all instances, and  $D = 0.005\sqrt{2}$  and  $D = 0.05\sqrt{n(n-1)/2}$  for the first and second numerical experiments, respectively. We report the running time (in seconds) and the objective value of the approximate solution at termination for all algorithms. For CG, we report its total number of iterations, which is the same as the number of gradient evaluations and linear optimization subproblems. For CGS and CGS-ls, we report the total number of gradient evaluations (denoted "outer") and linear optimization subproblems (denoted as "inner").

We make a few remarks from Tables 1 and 2. First, among the three algorithms, CG is the simplest to implement but has the worst performance; in most of the instances it could not obtain an approximate solution with Wolfe gap smaller than 0.01 within the required 30 minute computation time limit. Such behavior is consistent with its theoretical complexity  $\mathcal{O}(L/\varepsilon)$ , which is the worst among the three algorithms. Second, CGS-ls has better practical performance than CGS-ls in most instances, although both algorithms have the same theoretical convergence properties. The better practical performance is most likely due to the adaptive estimate of Lipschitz constant, which avoids the potentially conservative Lipschitz constant that CGS may suffer throughout the computation. Finally, it is interesting to observe that most objective

values of the approximate solutions computed by all algorithm at termination are much better than the accuracy setup  $\varepsilon = 0.01$ . To the best of our knowledge, it is still unclear in the literature whether there exists termination criterion other than the ones we use in this paper ((10) and (15)) that could guarantee that the approximate solution at termination is an  $\varepsilon$ -solution while achieving good practical performance. We leave the study of better termination criterion as a future work.

**Acknowledgements** Yuyuan Ouyang is partially supported by Office of Naval Research award N00014-20-1-2089.

## References

1. Bubeck, S., et al.: Convex optimization: Algorithms and complexity. Foundations and Trends® in Machine Learning **8**(3-4), 231–357 (2015)
2. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval research logistics quarterly **3**(1-2), 95–110 (1956)
3. Gurobi Optimization, L.: Gurobi optimizer reference manual (2020). URL <http://www.gurobi.com>
4. Harchaoui, Z., Juditsky, A., Nemirovski, A.: Conditional gradient algorithms for norm-regularized smooth convex optimization. Mathematical Programming **152**(1-2), 75–112 (2015)
5. Jaggi, M.: Revisiting frank-wolfe: Projection-free sparse convex optimization. In: ICML (1), pp. 427–435 (2013)
6. Lan, G.: The complexity of large-scale convex programming under a linear optimization oracle. arXiv preprint arXiv:1309.5550 (2013)
7. Lan, G.: First-order and Stochastic Optimization Methods for Machine Learning. Springer (2020)
8. Lan, G., Pokutta, S., Zhou, Y., Zink, D.: Conditional accelerated lazy stochastic gradient descent. In: International Conference on Machine Learning, pp. 1965–1974 (2017)
9. Lan, G., Zhou, Y.: Conditional gradient sliding for convex optimization. SIAM Journal on Optimization **26**(2), 1379–1409 (2016)
10. Nemirovski, A.: Efficient methods for convex optimization (lecture notes) (1994)
11. Nemirovski, A., Onn, S., Rothblum, U.G.: Accuracy certificates for computational problems with convex structure. Mathematics of Operations Research **35**(1), 52–78 (2010)
12. Nemirovski, A.S.: Information-based complexity of linear operator equations. Journal of Complexity **8**(2), 153–175 (1992)
13. Nesterov, Y.E.: A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ . Doklady AN SSSR **269**, 543–547 (1983). Translated as Soviet Math. Docl.
14. Nesterov, Y.E.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, Massachusetts (2004)
15. Nesterov, Y.E.: Universal gradient methods for convex optimization problems. Mathematical Programming **152**(1-2), 381–404 (2015)