OPERATIONS RESEARCH CENTER

Working Paper

New Analysis and Results for the Conditional Gradient Method

by

Robert M. Freund
Paul Grigas

OR 395-13 **July 2013**

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

New Analysis and Results for the Conditional Gradient Method

Robert M. Freund* Paul Grigas†

July 1, 2013

Abstract

We present new results for the conditional gradient method (also known as the Frank-Wolfe method). We derive computational guarantees for arbitrary step-size sequences, which are then applied to various step-size rules, including simple averaging and constant step-sizes. We also develop step-size rules and computational guarantees that depend naturally on the warm-start quality of the initial (and subsequent) iterates. Our results include computational guarantees for both duality/bound gaps and the so-called Wolfe gaps. Lastly, we present complexity bounds in the presence of approximate computation of gradients and/or linear optimization subproblem solutions.

1 Introduction

The use and analysis of first-order methods in convex optimization has gained a considerable amount of attention in recent years. For many applications – such as LASSO regression, boosting/classification, matrix completion, and other machine learning problems – first-order methods are appealing for a number of reasons. First, these problems are often very high-dimensional and thus, without any special structural knowledge, interior-point methods or other polynomial-time methods are unappealing. Second, in many applications the optimization models are dependent on data that can be noisy or otherwise limited, it is not necessary or even sensible to require a very high-accuracy solutions. Thus the weaker rates of convergence of first-order methods are typically satisfactory for such applications. Finally, first-order methods are appealing in many applications due to the lower computational burden per iteration, and the structural implications thereof. Indeed, most first-order methods require, at each iteration, the computation of an exact, approximate, or stochastic (sub)gradient and the computation of a solution to a particular "simple" subproblem. These computations typically scale well with the dimension of the problem and are often amenable to parallelization, distributed architectures, efficient management of sparse data-structures, and the like.

Our interest herein is the conditional gradient method, which is also referred to as the "Frank-Wolfe method." The original conditional gradient method, developed for quadratic programming

*MIT Sloan School of Management, 77 Massachusetts Avenue, Cambridge, MA 02139 (rfreund@mit.edu). This author's research is supported by AFOSR Grant No. FA9550-11-1-0141 and the MIT-Chile-Pontificia Universidad Católica de Chile Seed Fund.

†MIT Operations Research Center, 77 Massachusetts Avenue, Cambridge, MA 02139 (pgrigas@mit.edu). This author's research has been partially supported through NSF Graduate Research Fellowship No. 1122374 and the MIT-Chile-Pontificia Universidad Católica de Chile Seed Fund.

over a polytope, dates back to Frank and Wolfe [4], and was generalized to the more general smooth convex objective function over a bounded convex feasible region thereafter, see Levitin and Polyak [14], also Polyak [19]. More recently there has been renewed interest in the conditional gradient method due to some of its properties that we will shortly discuss, see for example Clarkson [1], Hazan [9], Jaggi [10], Giesen et al. [6], and most recently Harchaoui et al. [8] and Lan [13]. The conditional gradient method is premised on being able to easily solve (at each iteration) linear optimization problems over the feasible region of interest. This is in contrast to other first-order methods, such as the accelerated methods of Nesterov [16, 17], which are premised on being able to easily solve (at each iteration) certain projection problems defined by a strongly convex prox function. In many applications, solving a linear optimization subproblem is much simpler than solving the relevant projection subproblem. Moreover, in many applications the solutions to the linear optimization subproblem are often highly structured and exhibit particular sparsity and/or low-rank properties. The conditional gradient method solves one subproblem at each iteration and produces a sequence of feasible solutions that are each a convex combination of all previous subproblem solutions, for which one can derive an $O(\frac{1}{k})$ rate of convergence for appropriately chosen step-sizes. Due to the structure of the subproblem solutions and the fact that iterates are convex combinations of subproblem solutions, the feasible solutions returned by the conditional gradient method are also typically very highly-structured. For example, when the feasible region is the unit simplex $\Delta_n := \{\lambda \in \mathbb{R}^n : e^T\lambda = 1, \lambda \geq 0\}$ and the linear optimization oracle always returns an extreme point, then the conditional gradient method has the following sparsity property: the solution that the method produces at iteration k has at most k non-zero entries. (This observation generalizes to the matrix optimization setting: if the feasible region is a ball induced by the nuclear norm, then at iteration k the rank of the matrix produced by the method is at most k.) In many applications, such structural properties are highly desirable, and in such cases the conditional gradient method may be more attractive than the faster accelerated methods, even though the conditional gradient method has a slower rate of convergence.

The first set of contributions in this paper concern computational guarantees for arbitrary step-size sequences. In Section 2, we present a new complexity analysis of the conditional gradient method wherein we derive an exact functional dependence of the complexity bound at iteration k as a function of the step-size sequence $\{\bar{\alpha}_k\}$. We derive bounds on the deviation from the optimal objective function value (and on the duality gap in the presence of minmax structure), and on the so-called Wolfe gaps as first treated by Giesen et al. [6]. In Section 3, we use the technical theorems developed in Section 2 to derive computational guarantees for a variety of simple step-size rules including the well-studied step-size rule $\bar{\alpha}_k := \frac{2}{k+2}$, simple averaging, and constant step-sizes. Our analysis retains the well-known optimal $O(\frac{1}{k})$ rate (optimal for linear optimization oracle-based methods [13]) when the step-size is either given by the $\bar{\alpha}_k := \frac{2}{k+2}$ rule or is determined by a line-search. We also derive an $O\left(\frac{\ln(k)}{k}\right)$ rate for both the case when the step-size is given by simple averaging and in the case when the step-size is simply a suitably chosen constant.

The second set of contributions in this paper concern "warm-start" step-size rules and associated computational guarantees that reflect the the quality of the given initial iterate. The $O(\frac{1}{k})$ computational guarantees associated with the step-size sequence $\bar{\alpha}_k := \frac{2}{k+2}$ are independent of quality of the initial iterate. This is good if objective function value of the initial iterate is very far from the optimal value, as the poor quality of the initial iterate does not affect the computational guarantee. But if the objective function value of the initial iterate is moderately close to

the optimal value, one would want the conditional gradient method, with an appropriate step-size sequence, to have computational guarantees that reflect the closeness to optimality of the initial objective function value. In Section 4, we introduce a modification of the $\bar{\alpha}_k := \frac{2}{k+2}$ step-size rule that incorporates the quality of the initial iterate. Our new step-size rule maintains the $O(\frac{1}{k})$ complexity bound but now the bound is enhanced by the quality of the initial iterate. We also introduce a dynamic version of this warm start step-size rule, which dynamically incorporates all new bound information at each iteration. For the dynamic step-size rule, we also derive a $O(\frac{1}{k})$ complexity bound that depends naturally on all of the bound information obtained throughout the course of the algorithm.

The third set of contributions concern computational guarantees in the presence of approximate computation of gradients and linear optimization subproblem solutions. In Section 5, we first consider a variation of the conditional gradient method where the linear optimization subproblem at iteration k is solved approximately to an absolute accuracy of δ_k. We show that, independent of the choice of step-size sequence $\{\bar{\alpha}_k\}$, the conditional gradient method does not suffer from an accumulation of errors in the presence of approximate subproblem solutions. We extend the "technical" complexity theorems of Section 2, which imply, for instance, that when an optimal step-size such as $\bar{\alpha}_k := \frac{2}{k+2}$ is used and the $\{\delta_k\}$ accuracy sequence is a constant δ, then a solution with accuracy $O(\frac{1}{k} + \delta)$ can be achieved in k iterations. We next examine variations of the conditional gradient method where exact gradient computations are replaced with inexact gradient computations, under two different models of inexact gradient computations. We show that all of the complexity results under the previously examined approximate subproblem solution case (including, for instance, the non-accumulation of errors) directly apply to the case where exact gradient computations are replaced with the δ-oracle approximate gradient model introduced by d'Aspremont [2]. We also examine replacing exact gradient computations with the (δ, L)-oracle model introduced by Devolder et al. [3]. In this case the conditional gradient method suffers from an accumulation of errors under essentially any step-size sequence $\{\bar{\alpha}_k\}$. These results provide some insight into the inherent tradeoffs faced in choosing among several first-order methods.

1.1 Notation

Let E be a finite-dimensional real vector space with dual vector space E^*. For a given $s \in E^*$ and a given $\lambda \in E$, let $s^T\lambda$ denote the evaluation of the linear functional s at λ. For a norm $\|\cdot\|$ on E, let $B(c,r) = \{\lambda \in E : \|\lambda - c\| \leq r\}$. The dual norm $\|\cdot\|_*$ on the space E^* is defined by $\|s\|_* := \max_{\lambda \in B(0,1)} \{s^T\lambda\}$ for a given $s \in E^*$. The notation "$\tilde{v} \leftarrow \arg\max_{v \in S}\{f(v)\}$" denotes assigning $\tilde{v}$ to be any optimal solution of the problem $\max_{v \in S}\{f(v)\}$.

2 The Conditional Gradient Method

We recall the conditional gradient method for convex optimization, see Levitin and Polyak [14] and Polyak [19] (also referred to as the "Frank-Wolfe algorithm" from [4]), stated here for maximization problems:

$$\begin{array}{ll} \max\limits_{\lambda} & h(\lambda) \\ \text{s.t.} & \lambda \in Q \ , \end{array} \tag{1}$$

where $Q \subset E$ is convex and compact, and $h(\cdot) : Q \to \mathbb{R}$ is concave and differentiable on Q. Let h^* denote the optimal objective function value of (1). The basic conditional gradient method is presented in Method 1, where the main computational requirement at each iteration is to solve a linear optimization problem over Q in Step (2.) of the method. The step-size $\bar{\alpha}_k$ in Step (4.) could be chosen by inexact or exact line-search, or by a pre-determined or dynamically determined step-size sequence $\{\bar{\alpha}_k\}$. Also note that the version of the conditional gradient method in Method 1 does not allow a (full) step-size $\bar{\alpha}_k = 1$, the reasons for which will become apparent below.

Method 1 Conditional Gradient Method for maximizing $h(\lambda)$

Initialize at $\lambda_1 \in Q$, (optional) initial upper bound B_0, $k \leftarrow 1$.
At iteration k:
1. Compute $\nabla h(\lambda_k)$.
2. Compute $\tilde{\lambda}_k \leftarrow \arg\max_{\lambda \in Q} \{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\}$.
   $B_k^w \leftarrow h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$.
   $G_k \leftarrow \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$.
3. (Optional: compute other upper bound B_k^o), update best bound $B_k \leftarrow \min\{B_{k-1}, B_k^w, B_k^o\}$.
4. Set $\lambda_{k+1} \leftarrow \lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$, where $\bar{\alpha}_k \in [0, 1)$.

As a consequence of solving the linear optimization problem in Step (2.) of the method, one conveniently obtains the "Wolfe upper bound" on the optimal value h^* of (1):

$$B_k^w := h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) \ , \tag{2}$$

and it follows from the fact that the linearization of $h(\cdot)$ at λ_k dominates $h(\cdot)$ that B_k^w is a valid upper bound on h^*. We are also interested in the "Wolfe gap" G_k at each iteration:

$$G_k := B_k^w - h(\lambda_k) = \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) \ . \tag{3}$$

Note that $G_k \geq h^* - h(\lambda_k) \geq 0$. In certain contexts, G_k is an important quantity by itself, see Khachiyan [12], Giesen et al. [6], as well as [5]. Jaggi [10] first showed that the conditional gradient method generates upper bound guarantees on G_k, see also Harchaoui et al. [8], although Khachiyan implicitly derived such bounds for the conditional gradient method applied to the minimum volume covering ellipsoid problem in [12]. Both B_k^w and G_k are computed directly from the solution of the linear optimization problem in Step (2.) and are recorded therein for convenience.

In some of our analysis of the conditional gradient method, the computational guarantees will depend on the quality of upper bounds on h^*. In addition to the Wolfe bound B_k^w, Step (3.) allows for an "optional other upper bound B_k^o " that also might be computed at iteration k. Sometimes there is structural knowledge of an upper bound as a consequence of a dual problem associated with (1), as when $h(\cdot)$ is conveyed with minmax structure, namely:

$$h(\lambda) = \min_{x \in P} \ \phi(x, \lambda) \ , \tag{4}$$

where P is a closed convex set and $\phi(\cdot, \cdot) : P \times Q \to \mathbb{R}$ is a continuous function that is convex in the first variable x and concave in the second variable λ. In this case define the convex function $f(\cdot) : P \to \mathbb{R}$ given by $f(x) := \max_{\lambda \in Q} \ \phi(x, \lambda)$ and consider the following duality paired problems:

$$\text{(Primal):}\quad \min_{x\in P} f(x) \qquad \text{and} \qquad \text{(Dual):}\quad \max_{\lambda\in Q} h(\lambda)\ , \tag{5}$$

where here it is the dual problem that corresponds to our problem of interest (1). Weak duality holds, namely $h(\lambda) \leq h^* \leq f(x)$ for all $x \in P, \lambda \in Q$. At any iterate $\lambda_k \in Q$ of the conditional gradient method one can construct a "minmax" upper bound on h^* by considering the variable x in that structure:

$$B_k^m := f(x_k) := \max_{\lambda\in Q}\{\phi(x_k,\lambda)\} \quad \text{where} \quad x_k \in \arg\min_{x\in P}\{\phi(x,\lambda_k)\}\ , \tag{6}$$

and it follows from weak duality that $B_k^o := B_k^m$ is a valid upper bound for all k. Notice that x_k defined above is the "optimal response" to λ_k in a minmax sense and hence is a natural choice of duality-paired variable associated with the variable λ_k. Under certain regularity conditions, for instance when $h(\cdot)$ is globally differentiable on E, one can show that B_k^m is at least as tight a bound as Wolfe's bound, namely $B_k^m \leq B_k^w$ for all k (see Proposition A.1), and therefore the Wolfe gap G_k conveniently bounds this minmax duality gap: $B_k^m - h(\lambda_k) \leq B_k^w - h(\lambda_k) = G_k$.

(Indeed, in the minmax setting notice that the optimal response x_k in (6) is a function of the current iterate λ_k and hence $f(x_k) - h(\lambda_k) = B_k^m - h(\lambda_k)$ is not just *any* duality gap but rather is determined completely by the current iterate λ_k. This special feature of the duality gap $B_k^m - h(\lambda_k)$ is exploited in the application of the conditional gradient method to rounding of polytopes [12], parametric optimization on the spectrahedron [6], and to regularized regression [5] (and perhaps elsewhere as well), where bounds on the Wolfe gap G_k are used to bound $B_k^m - h(\lambda_k)$ directly.)

We also mention that in some applications there might be exact knowledge of the optimal value h^* (as in certain linear regression applications where one knows *a priori* that the optimal value of the residuals is zero), whereby one can set $B_k^o \leftarrow h^*$.

It will also be useful to analyze a version of the conditional gradient method wherein there is a single "pre-start" step. In this case we are given some $\lambda_0 \in Q$ and some upper bound B_{-1} on h^* (one can use $B_{-1} = +\infty$ if no information is available) and we proceed like any other iteration except that in Step (4.) we set $\lambda_1 \leftarrow \tilde{\lambda}_0$, which is equivalent to setting $\bar{\alpha}_0 := 1$. This is shown formally in the Pre-start Procedure 2.

Procedure 2 Pre-start Step of Conditional Gradient Method given $\lambda_0 \in Q$ and (optional) upper bound B_{-1}

1. Compute $\nabla h(\lambda_0)$.
2. Compute $\tilde{\lambda}_0 \leftarrow \arg\max_{\lambda\in Q}\{h(\lambda_0) + \nabla h(\lambda_0)^T(\lambda - \lambda_0)\}$.
   $B_0^w \leftarrow h(\lambda_0) + \nabla h(\lambda_0)^T(\tilde{\lambda}_0 - \lambda_0)$.
   $G_0 \leftarrow \nabla h(\lambda_0)^T(\tilde{\lambda}_0 - \lambda_0)$.
3. (Optional: compute other upper bound B_0^o), update best bound $B_0 \leftarrow \min\{B_{-1}, B_0^w, B_0^o\}$.
4. Set $\lambda_1 \leftarrow \tilde{\lambda}_0$.

Towards stating and proving complexity bounds for the conditional gradient method, resembling Clarkson [1] we consider the following curvature constant $C_{h,Q}$, which is defined to be the minimal value of C satisfying:

$$h(\lambda + \alpha(\bar{\lambda} - \lambda)) \geq h(\lambda) + \nabla h(\lambda)^T(\alpha(\bar{\lambda} - \lambda)) - \frac{1}{2}C\alpha^2 \quad \text{for all } \lambda, \bar{\lambda} \in Q \ \text{ and all } \alpha \in [0,1]\ . \tag{7}$$

For any choice of norm $\|\cdot\|$ on E, let Diam_Q denote the diameter of Q measured with the norm $\|\cdot\|$, namely $\mathrm{Diam}_Q := \max_{\lambda,\bar{\lambda}\in Q}\{\|\lambda - \bar{\lambda}\|\}$ and let $L_{h,Q}$ be the Lipschitz constant for $\nabla h(\cdot)$ on Q, namely $L_{h,Q}$ is the smallest constant L for which it holds that:

$$\|\nabla h(\lambda) - \nabla h(\bar{\lambda})\|_* \leq L\|\lambda - \bar{\lambda}\| \quad \text{for all } \lambda, \bar{\lambda} \in Q \ .$$

It is straightforward to show that $C_{h,Q}$ is bounded above by the more classical metrics Diam_Q and $L_{h,Q}$, namely

$$C_{h,Q} \leq L_{h,Q}(\mathrm{Diam}_Q)^2 \ , \tag{8}$$

see Proposition A.2. In contrast to other (proximal) first-order methods, the conditional gradient method does not depend on a choice of norm. The norm invariant definition of $C_{h,Q}$ and the fact that (8) holds for *any* norm are therefore particularly appealing properties of $C_{h,Q}$ as a behavioral measure for the conditional gradient method.

Towards stating our main technical results, we define the following two auxiliary sequences, where α_k and β_k are functions of the first k step-size sequence values, $\bar{\alpha}_1, \ldots, \bar{\alpha}_k$, from the conditional gradient method:

$$\beta_k = \frac{1}{\prod_{j=1}^{k-1}(1-\bar{\alpha}_j)} \ , \qquad \alpha_k = \frac{\beta_k \bar{\alpha}_k}{1-\bar{\alpha}_k} \ , \qquad k \geq 1 \ . \tag{9}$$

(Here and in what follows we use the conventions: $\prod_{j=1}^{0} \cdot = 1$ and $\sum_{i=1}^{0} \cdot = 0$.)

The following two theorems are our main technical constructs that will be used to develop the results herein. The first theorem concerns optimality gap bounds.

Theorem 2.1. *Consider the iterate sequences of the conditional gradient method (Method 1) $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$ and the sequence of upper bounds $\{B_k\}$ on h^*, using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$ given by (9), and for any $k \geq 0$, the following inequality holds:*

$$B_k - h(\lambda_{k+1}) \leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \ . \tag{10}$$

□

The summation expression in the rightmost term above appears also in the bound given on the dual averaging method of Nesterov [18]. Indeed, this is no coincidence as the sequences $\{\alpha_k\}$ and $\{\beta_k\}$ given by (9) arise precisely by a connection between the conditional gradient method and the dual averaging method, see [7]. For this reason we will henceforth refer to the sequences (9) as the "dual averages" sequences associated with $\{\bar{\alpha}_k\}$. The second theorem concerns the Wolfe gap values G_k from Step (2.) in particular.

Theorem 2.2. *Consider the iterate sequences of the conditional gradient method (Method 1) $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$, the sequence of upper bounds $\{B_k\}$ on h^*, and the sequence of Wolfe gaps $\{G_k\}$ from*

Step (2.), using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$ given by (9), and for any $\ell \geq 0$ and $k \geq \ell + 1$, the following inequality holds:

$$\min_{i \in \{\ell+1,\ldots,k\}} G_i \leq \frac{1}{\sum_{i=\ell+1}^{k} \bar{\alpha}_i} \left[\frac{B_\ell - h(\lambda_1)}{\beta_{\ell+1}} + \frac{\frac{1}{2} C_{h,Q} \sum_{i=1}^{\ell} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{\ell+1}} \right] + \frac{\frac{1}{2} C_{h,Q} \sum_{i=\ell+1}^{k} \bar{\alpha}_i^2}{\sum_{i=\ell+1}^{k} \bar{\alpha}_i} \ . \tag{11}$$

□

Theorems 2.1 and 2.2 can be applied to yield specific complexity results for *any* specific step-size sequence $\{\bar{\alpha}_k\}$ (satisfying the mild assumption that $\bar{\alpha}_k < 1$) through the use of the implied $\{\alpha_k\}$ and $\{\beta_k\}$ dual averages sequences. This is shown for several useful step-size sequences in the next section.

Proof of Theorem 2.1: We will show the slightly more general result for $k \geq 0$:

$$\min\{B, B_k\} - h(\lambda_{k+1}) \leq \frac{B - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2} C_{h,Q} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \qquad \text{for any } B \ , \tag{12}$$

from which (10) follows by substituting $B = B_k$ above.

For $k = 0$ the result follows trivially since $\beta_1 = 1$ and the summation term on the right side of (12) is zero by the conventions for null products and summations stated earlier. For $k \geq 1$, we begin by observing that the following equalities hold for the dual averages sequences (9):

$$\beta_{i+1} - \beta_i = \bar{\alpha}_i \beta_{i+1} = \alpha_i \quad \text{and} \quad \beta_{i+1} \bar{\alpha}_i^2 = \frac{\alpha_i^2}{\beta_{i+1}} \quad \text{for } i \geq 1 \ , \tag{13}$$

and

$$1 + \sum_{i=1}^{k} \alpha_i = \beta_{k+1} \quad \text{for } k \geq 1 \ . \tag{14}$$

We then have for $i \geq 1$:

$$\begin{aligned}
\beta_{i+1} h(\lambda_{i+1}) &\geq \beta_{i+1} \left[h(\lambda_i) + \nabla h(\lambda_i)^T (\tilde{\lambda}_i - \lambda_i) \bar{\alpha}_i - \frac{1}{2} \bar{\alpha}_i^2 C_{h,Q} \right] \\
&= \beta_i h(\lambda_i) + (\beta_{i+1} - \beta_i) h(\lambda_i) + \beta_{i+1} \bar{\alpha}_i \nabla h(\lambda_i)^T (\tilde{\lambda}_i - \lambda_i) - \frac{1}{2} \beta_{i+1} \bar{\alpha}_i^2 C_{h,Q} \\
&= \beta_i h(\lambda_i) + \alpha_i h(\lambda_i) + \alpha_i \nabla h(\lambda_i)^T (\tilde{\lambda}_i - \lambda_i) - \frac{1}{2} \frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \\
&= \beta_i h(\lambda_i) + \alpha_i \left[h(\lambda_i) + \nabla h(\lambda_i)^T (\tilde{\lambda}_i - \lambda_i) \right] - \frac{1}{2} \frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \\
&= \beta_i h(\lambda_i) + \alpha_i B_i^w - \frac{1}{2} \frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \ .
\end{aligned}$$

The inequality in the first line above follows from the definition of $C_{h,Q}$ in (7) and $\lambda_{i+1} - \lambda_i = \bar{\alpha}_i (\tilde{\lambda}_i - \lambda_i)$. The second equality above uses the identities (13), and the fourth equality uses the

definition of the Wolfe upper bound (2). Rearranging and summing the above over i, it follows that for any scalar B:

$$B + \sum_{i=1}^{k} \alpha_i B_i^w \leq B + \beta_{k+1} h(\lambda_{k+1}) - \beta_1 h(\lambda_1) + \frac{1}{2} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \ . \tag{15}$$

Therefore

$$\begin{aligned} \min\{B, B_k\} \beta_{k+1} &= \min\{B, B_k\} \left(1 + \sum_{i=1}^{k} \alpha_i\right) \\ &\leq B + \sum_{i=1}^{k} \alpha_i B_i^w \\ &\leq B + \beta_{k+1} h(\lambda_{k+1}) - h(\lambda_1) + \frac{1}{2} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \ , \end{aligned}$$

where the first equality above uses identity (14), the first inequality uses the fact that $B_k \leq B_i^w$ for $i \leq k$, and the second inequality uses (15) and the fact that $\beta_1 = 1$. The result then follows by dividing by β_{k+1} and rearranging terms. □

Proof of Theorem 2.2: For $i \geq 1$ we have:

$$\begin{aligned} h(\lambda_{i+1}) &\geq h(\lambda_i) + \nabla h(\lambda_i)^T (\tilde{\lambda}_i - \lambda_i) \bar{\alpha}_i - \tfrac{1}{2} \bar{\alpha}_i^2 C_{h,Q} \\ &= h(\lambda_i) + \bar{\alpha}_i G_i - \tfrac{1}{2} \bar{\alpha}_i^2 C_{h,Q} \ , \end{aligned} \tag{16}$$

where the inequality follows from the definition of the curvature constant in (7), and the equality follows from the definition of the Wolfe gap in (3). Summing the above over $i \in \{\ell + 1, \ldots, k\}$ and rearranging yields:

$$\textstyle\sum_{i=\ell+1}^{k} \bar{\alpha}_i G_i \ \leq h(\lambda_{k+1}) - h(\lambda_{\ell+1}) + \sum_{i=\ell+1}^{k} \frac{1}{2} \bar{\alpha}_i^2 C_{h,Q} \ . \tag{17}$$

Combining (17) with Theorem 2.1 we obtain:

$$\sum_{i=\ell+1}^{k} \bar{\alpha}_i G_i \leq h(\lambda_{k+1}) - B_\ell + \frac{B_\ell - h(\lambda_1)}{\beta_{\ell+1}} + \frac{\frac{1}{2} C_{h,Q} \sum_{i=1}^{\ell} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{\ell+1}} + \sum_{i=\ell+1}^{k} \frac{1}{2} \bar{\alpha}_i^2 C_{h,Q} \ ,$$

and since $B_\ell \geq h^* \geq h(\lambda_{k+1})$ we obtain:

$$\left(\min_{i \in \{\ell+1, \ldots, k\}} G_i\right) \left(\sum_{i=\ell+1}^{k} \bar{\alpha}_i\right) \leq \sum_{i=\ell+1}^{k} \bar{\alpha}_i G_i \leq \frac{B_\ell - h(\lambda_1)}{\beta_{\ell+1}} + \frac{\frac{1}{2} C_{h,Q} \sum_{i=1}^{\ell} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{\ell+1}} + \sum_{i=\ell+1}^{k} \frac{1}{2} \bar{\alpha}_i^2 C_{h,Q} \ ,$$

and dividing by $\sum_{i=\ell+1}^{k} \bar{\alpha}_i$ yields the result. □

3 Computational Guarantees for Specific Step-size Sequences

Herein we use Theorems 2.1 and 2.2 to derive computational guarantees for a variety of specific step-size sequences. We first present a property of the pre-start step (Procedure 2) that has implications for such computational guarantees.

Proposition 3.1. *Let λ_1 and B_0 be computed by the pre-start step Procedure 2. Then $B_0 - h(\lambda_1) \leq \frac{1}{2}C_{h,Q}$.*

Proof. We have $\lambda_1 = \tilde{\lambda}_0$ and $B_0 \leq B_0^w$, whereby from the definition of $C_{h,Q}$ using $\alpha = 1$ we have:

$$h(\lambda_1) = h(\tilde{\lambda}_0) \geq h(\lambda_0) + \nabla h(\lambda_0)^T(\tilde{\lambda}_0 - \lambda_0) - \frac{1}{2}C_{h,Q} = B_0^w - \frac{1}{2}C_{h,Q} \geq B_0 - \frac{1}{2}C_{h,Q} \ ,$$

and the result follows by rearranging terms. □

3.1 A Well-studied Step-size Sequence

Suppose we initiate the conditional gradient method with the pre-start step Procedure 2 from a given value $\lambda_0 \in Q$ (which by definition assigns the step-size $\bar{\alpha}_0 = 1$ as discussed earlier), and then use the step-size $\bar{\alpha}_i = 2/(i+2)$ for $i \geq 1$. This can be written equivalently as:

$$\bar{\alpha}_i = \frac{2}{i+2} \qquad \text{for } i \geq 0 \ . \tag{18}$$

Computational guarantees for this sequence appeared in Hazan [9] (with a corrected proof in Giesen et al. [6]). In unpublished correspondence with the second author in 2007, Nemirovski [15] presented a short inductive proof of convergence of the conditional gradient method using this step-size rule.

We use the phrase "bound gap" to generically refer to the difference between an upper bound B on h^* and the value $h(\lambda)$, namely $B - h(\lambda)$. The following result describes guarantees on the bound gap $B_k - h(\lambda_{k+1})$ and the Wolfe gap G_k using the step-size sequence (18), that are applications of Theorems 2.1 and 2.2, and that are very minor improvements of existing results as discussed below.

Bound 3.1. *Under the step-size sequence (18), the following inequalities hold for all $k \geq 1$:*

$$B_k - h(\lambda_{k+1}) \leq \frac{2C_{h,Q}}{k+4} \tag{19}$$

and

$$\min_{i \in \{1,\ldots,k\}} G_i \leq \frac{4.5C_{h,Q}}{k} \ . \tag{20}$$

The bound (19) is a very minor improvement over that in Hazan [9] and Giesen et al. [6], see also Harchaoui et al. [8], as the denominator is additively larger by 1 (after accounting for the pre-start step and the different indexing conventions). The bound (20) is a modification of the original bound in Jaggi [10] (which required changing to a constant step-size for iterations $k+1, \ldots, 2k$), and is also a slight improvement of the bound in Harchaoui et al. [8] inasmuch as the denominator is additively larger by 1 and the bound is valid for all $k \geq 1$.

Proof of Bound 3.1: Using (18) it is easy to show that the dual averages sequences (9) satisfy $\beta_k = \frac{k(k+1)}{2}$ and $\alpha_k = k+1$ for $k \geq 1$. Utilizing Theorem 2.1, we have for $k \geq 1$:

$$\begin{aligned}
B_k - h(\lambda_{k+1}) &\leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \\
&\leq \frac{B_0 - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \\
&\leq \frac{\frac{1}{2}C_{h,Q}}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \\
&= \frac{C_{h,Q}}{(k+1)(k+2)}\left[1 + \sum_{i=1}^{k}\frac{2(i+1)^2}{(i+1)(i+2)}\right] \\
&= \frac{C_{h,Q}}{(k+1)(k+2)}\left[\sum_{i=0}^{k}\frac{2(i+1)}{(i+2)}\right] \\
&\leq \frac{2C_{h,Q}}{k+4} \ ,
\end{aligned}$$

where the second inequality uses $B_k \leq B_0$, the third inequality uses Proposition 3.1, the first equality substitutes the dual averages sequence values, and the final inequality follows from Proposition A.3. This proves (19).

To prove (20) we proceed as follows. First apply Theorem 2.2 with $\ell = 0$ and $k = 1$ to obtain:

$$G_1 \leq \frac{1}{\bar{\alpha}_1}\left[B_0 - h(\lambda_1)\right] + \frac{1}{2}C_{h,Q}\bar{\alpha}_1 \leq \frac{1}{2}C_{h,Q}\left[\frac{1}{\bar{\alpha}_1} + \bar{\alpha}_1\right] = \frac{1}{2}C_{h,Q}\left[\frac{3}{2} + \frac{2}{3}\right] = \frac{13}{12}C_{h,Q} \ ,$$

where the second inequality uses Proposition 3.1. Since $\frac{13}{12} \leq 4.5$ and $\frac{13}{12} \leq \frac{4.5}{2}$, this proves (20) for $k = 1, 2$. Assume now that $k \geq 3$. Let $\ell = \lceil \frac{k}{2} \rceil - 2$ so that $\ell \geq 0$. We have:

$$\sum_{i=\ell+1}^{k}\bar{\alpha}_i = 2\sum_{i=\ell+1}^{k}\frac{1}{i+2} = 2\sum_{i=\ell+3}^{k+2}\frac{1}{i} \geq 2\ln\left(\frac{k+3}{\ell+3}\right) \geq 2\ln\left(\frac{k+3}{\frac{k}{2}+1.5}\right) = 2\ln(2) \ , \tag{21}$$

where the first inequality uses Proposition A.5 and the second inequality uses $\lceil \frac{k}{2} \rceil \leq \frac{k}{2} + \frac{1}{2}$. We also have:

$$\sum_{i=\ell+1}^{k}\bar{\alpha}_i^2 = 4\sum_{i=\ell+1}^{k}\frac{1}{(i+2)^2} = 4\sum_{i=\ell+3}^{k+2}\frac{1}{i^2} \leq \frac{4(k-\ell)}{(\ell+2)(k+2)} \leq \frac{4\left(\frac{k}{2}+2\right)}{\frac{k}{2}(k+2)} = \frac{4(k+4)}{k(k+2)} \ , \tag{22}$$

where the first inequality uses Proposition A.5 and the second inequality uses $\lceil \frac{k}{2} \rceil \geq \frac{k}{2}$. Applying

Theorem 2.2 and using (21) and (22) yields:

$$\begin{aligned}\min_{i\in\{1,\ldots,k\}} G_i &\leq \frac{1}{2\ln(2)}\left[\frac{B_\ell - h(\lambda_1)}{\beta_{\ell+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{\ell}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{\ell+1}} + \frac{2C_{h,Q}(k+4)}{k(k+2)}\right] \\ &\leq \frac{1}{2\ln(2)}\left[\frac{2C_{h,Q}}{\ell+4} + \frac{2C_{h,Q}(k+4)}{k(k+2)}\right] \\ &\leq \frac{2C_{h,Q}}{2\ln(2)}\left[\frac{2}{k+4} + \frac{k+4}{k(k+2)}\right] = \frac{2C_{h,Q}}{2\ln(2)}\left[\frac{3k^2+12k+16}{(k+4)(k+2)k}\right] \leq \frac{2C_{h,Q}}{2\ln(2)}\left(\frac{3}{k}\right) \leq \frac{4.5C_{h,Q}}{k}\ ,\end{aligned}$$

where the second inequality uses the chain of inequalities used to prove (19), the third inequality uses $\ell + 4 \geq \frac{k}{2} + 2$, and the fourth inequality uses $k^2 + 4k + \frac{16}{3} \leq k^2 + 6k + 8 = (k+4)(k+2)$. □

3.2 Simple Averaging

Consider the following step-size sequence:

$$\bar{\alpha}_i = \frac{1}{i+1} \qquad \text{for } i \geq 0\ , \tag{23}$$

where, as with the step-size sequence (18), we write $\bar{\alpha}_0 = 1$ to indicate the use of the pre-start step Procedure 2. It follows from a simple inductive argument that, under the step-size sequence (23), λ_{k+1} is the simple average of $\tilde{\lambda}_0, \tilde{\lambda}_1, \ldots, \tilde{\lambda}_k$, i.e., we have

$$\lambda_{k+1} = \frac{1}{k+1}\sum_{i=0}^{k}\tilde{\lambda}_i \qquad \text{for all } k \geq 0\ .$$

Bound 3.2. *Under the step-size sequence* (23)*, the following inequality holds for all* $k \geq 0$*:*

$$B_k - h(\lambda_{k+1}) \leq \frac{\frac{1}{2}C_{h,Q}(1+\ln(k+1))}{k+1}\ , \tag{24}$$

and the following inequality holds for all $k \geq 2$*:*

$$\min_{i\in\{1,\ldots,k\}} G_i \leq \frac{\frac{3}{4}C_{h,Q}\,(2.3 + 2\ln(k))}{k-1}\ . \tag{25}$$

Proof of Bound 3.2: Using (23) it is easy to show that the dual averages sequences (9) are given by $\beta_k = k$ and $\alpha_k = 1$ for $k \geq 1$. Utilizing Theorem 2.1 and Proposition 3.1, we have for $k \geq 1$:

$$\begin{aligned}B_k - h(\lambda_{k+1}) &\leq \frac{\frac{1}{2}C_{h,Q}}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \\ &= \frac{\frac{1}{2}C_{h,Q}}{k+1}\left[1 + \sum_{i=1}^{k}\frac{1}{i+1}\right] \\ &\leq \frac{\frac{1}{2}C_{h,Q}}{k+1}\left[1+\ln(k+1)\right]\ ,\end{aligned}$$

where the first equality substitutes the dual averages sequence values and the second inequality uses Proposition A.5. This proves (24). To prove (25), we proceed as follows. Let $\ell = \lfloor \frac{k}{2} \rfloor - 1$, whereby $\ell \geq 0$ since $k \geq 2$. We have:

$$\sum_{i=\ell+1}^{k} \bar{\alpha}_i = \sum_{i=\ell+1}^{k} \frac{1}{i+1} = \sum_{i=\ell+2}^{k+1} \frac{1}{i} \geq \ln\left(\frac{k+2}{\ell+2}\right) \geq \ln\left(\frac{k+2}{\frac{k}{2}+1}\right) = \ln(2) \ , \tag{26}$$

where the first inequality uses Proposition A.5 and the second inequality uses $\ell \leq \frac{k}{2} - 1$. We also have:

$$\sum_{i=\ell+1}^{k} \bar{\alpha}_i^2 = \sum_{i=\ell+1}^{k} \frac{1}{(i+1)^2} = \sum_{i=\ell+2}^{k+1} \frac{1}{i^2} \leq \frac{k-\ell}{(\ell+1)(k+1)} \leq \frac{\frac{k}{2}+1.5}{(\frac{k}{2}-\frac{1}{2})(k+1)} = \frac{k+3}{(k-1)(k+1)} \ , \tag{27}$$

where the first inequality uses Proposition A.5 and the second inequality uses $\ell \geq \frac{k}{2} - 1.5$. Applying Theorem 2.2 and using (26) and (27) yields:

$$\begin{aligned}
\min_{i \in \{1,\ldots,k\}} G_i &\leq \frac{1}{\ln(2)} \left[\frac{B_\ell - h(\lambda_1)}{\beta_{\ell+1}} + \frac{\frac{1}{2}C_{h,Q} \sum_{i=1}^{\ell} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{\ell+1}} + \frac{\frac{1}{2}C_{h,Q}(k+3)}{(k-1)(k+1)} \right] \\
&\leq \frac{1}{\ln(2)} \left[\frac{\frac{1}{2}C_{h,Q}(1+\ln(\ell+1))}{\ell+1} + \frac{\frac{1}{2}C_{h,Q}(k+3)}{(k-1)(k+1)} \right] \\
&\leq \frac{\frac{1}{2}C_{h,Q}}{\ln(2)} \left[\frac{1+\ln(\frac{k}{2})}{\frac{k}{2}-\frac{1}{2}} + \frac{k+3}{(k-1)(k+1)} \right] \\
&\leq \frac{\frac{1}{2}C_{h,Q}}{\ln(2)} \left[\frac{2+2\ln(k)-2\ln(2)}{k-1} + \frac{\frac{5}{3}}{k-1} \right] \\
&\leq \frac{\frac{3}{4}C_{h,Q}\,(2.3+2\ln(k))}{k-1} \ ,
\end{aligned}$$

where the second inequality uses the bound that proves (24), the third inequality uses $\frac{k}{2} - 1.5 \leq \ell \leq \frac{k}{2} - 1$ and the fourth inequality uses $\frac{k+3}{k+1} \leq \frac{5}{3}$ for $k \geq 2$. □

3.3 Constant Step-size

Given $\bar{\alpha} \in (0, 1)$, consider using the following constant step-size rule:

$$\bar{\alpha}_i = \bar{\alpha} \qquad \text{for } i \geq 1 \ . \tag{28}$$

This step-size rule arises in the analysis of the Incremental Forward Stagewise Regression algorithm (FS_ε), see [5], and perhaps elsewhere as well.

Bound 3.3. *Under the step-size sequence (28), the following inequality holds for all $k \geq 1$:*

$$B_k - h(\lambda_{k+1}) \leq (B_k - h(\lambda_1))\,(1-\bar{\alpha})^k + \tfrac{1}{2}C_{h,Q}\left[\bar{\alpha} - \bar{\alpha}(1-\bar{\alpha})^k\right] \ . \tag{29}$$

If the pre-start step Procedure 2 is used, then:

$$B_k - h(\lambda_{k+1}) \leq \tfrac{1}{2}C_{h,Q}\left[(1-\bar{\alpha})^{k+1} + \bar{\alpha}\right] \ . \tag{30}$$

If we decide *a priori* to run the conditional gradient method for k iterations after the pre-start step Procedure 2, then we can optimize the bound (30) with respect to $\bar{\alpha}$. The optimized value of $\bar{\alpha}$ in the bound (30) is easily derived to be:

$$\bar{\alpha}^* = 1 - \frac{1}{\sqrt[k]{k+1}} \ . \tag{31}$$

With $\bar{\alpha}$ determined by (31), we obtain a simplified bound from (30) and also a guarantee for the Wolfe Gap sequence $\{G_k\}$ if the method is continued with the same constant step-size (31) for an additional $k+1$ iterations.

Bound 3.4. *If we use the pre-start step Procedure 2 and the constant step-size sequence* (31) *for all iterations, then after k iterations the following inequality holds:*

$$B_k - h(\lambda_{k+1}) \leq \frac{\frac{1}{2}C_{h,Q}\left(1+\ln(k+1)\right)}{k} \ . \tag{32}$$

Furthermore, after $2k+1$ iterations the following inequality holds:

$$\min_{i\in\{1,\ldots,2k+1\}} G_i \leq \frac{\frac{1}{2}C_{h,Q}\left(1+2\ln(k+1)\right)}{k} \tag{33}$$

It is curious to note that the bounds (24) and (32) are almost identical, although (32) requires fixing *a priori* the number of iterations k.

Proof of Bound 3.3: Under the step-size rule (28) it is straightforward to show that the dual averages sequences (9) are for $i \geq 1$:

$$\beta_i = (1-\bar{\alpha})^{-k+1} \quad \text{and} \quad \alpha_i = \bar{\alpha}(1-\bar{\alpha})^{-k} \ ,$$

whereby

$$\sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}} = \sum_{i=1}^{k} \bar{\alpha}^2(1-\bar{\alpha})^{-i} = \bar{\alpha}^2 \left(\frac{\frac{1}{(1-\bar{\alpha})^k} - 1}{\bar{\alpha}} \right) = \bar{\alpha}\left[(1-\bar{\alpha})^{-k} - 1\right] \ .$$

It therefore follows from Theorem 2.1 that:

$$\begin{aligned} B_k - h(\lambda_{k+1}) &\leq \tfrac{B_k - h(\lambda_1)}{\beta_{k+1}} + \tfrac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \\ &= \left(B_k - h(\lambda_1)\right)(1-\bar{\alpha})^k + \left(\tfrac{C_{h,Q}}{2}\right)\bar{\alpha}\left[(1-\bar{\alpha})^{-k} - 1\right](1-\bar{\alpha})^k \\ &= \left(B_k - h(\lambda_1)\right)(1-\bar{\alpha})^k + \left(\tfrac{C_{h,Q}}{2}\right)\left[\bar{\alpha} - \bar{\alpha}(1-\bar{\alpha})^k\right] \ , \end{aligned} \tag{34}$$

which proves (29). If the pre-start step Procedure 2 is used, then using Proposition 3.1 it follows that $B_k - h(\lambda_1) \leq B_0 - h(\lambda_1) \leq \frac{1}{2}C_{h,Q}$, whereby from (29) we obtain:

$$B_k - h(\lambda_{k+1}) \leq \frac{1}{2}C_{h,Q}(1-\bar{\alpha})^k + \left(\frac{C_{h,Q}}{2}\right)\left[\bar{\alpha} - \bar{\alpha}(1-\bar{\alpha})^k\right]$$

$$= \frac{1}{2}C_{h,Q}\left[(1-\bar{\alpha})^{k+1} + \bar{\alpha}\right] \ ,$$

completing the proof. □

Proof of Bound 3.4: Substituting the step-size (31) into (30) we obtain:

$$B_k - h(\lambda_{k+1}) \leq \frac{1}{2}C_{h,Q}\left[\left(\frac{1}{\sqrt[k]{k+1}}\right)^{k+1} + 1 - \frac{1}{\sqrt[k]{k+1}}\right]$$

$$\leq \frac{1}{2}C_{h,Q}\left[\left(\frac{1}{\sqrt[k]{k+1}}\right)^{k+1} + \frac{\ln(k+1)}{k}\right]$$

$$\leq \frac{1}{2}C_{h,Q}\left[\frac{1}{k+1} + \frac{\ln(k+1)}{k}\right]$$

$$\leq \frac{1}{2}C_{h,Q}\left[\frac{1}{k} + \frac{\ln(k+1)}{k}\right] ,$$

where the second inequality follows from *(i)* of Proposition A.4. This proves (32). To prove (33), notice that inequality (34) together with the subsequent chain of inequalities in the proofs of (29), (30), and (32) show that:

$$\left[\frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}}\right] \leq \frac{1}{2}C_{h,Q}\left(\frac{1+\ln(k+1)}{k}\right) . \tag{35}$$

Using (35) and the substitution $\sum_{i=k+1}^{2k+1}\bar{\alpha}_i = (k+1)\bar{\alpha}$ and $\sum_{i=k+1}^{2k+1}\bar{\alpha}_i^2 = (k+1)\bar{\alpha}^2$ in Theorem 2.2 yields:

$$\min_{i\in\{1,\ldots,2k+1\}} G_i \leq \frac{1}{(k+1)\bar{\alpha}}\left(\frac{\frac{1}{2}C_{h,Q}(1+\ln(k+1))}{k}\right) + \frac{\frac{1}{2}C_{h,Q}(k+1)\bar{\alpha}^2}{(k+1)\bar{\alpha}}$$

$$\leq \tfrac{1}{2}C_{h,Q}\left(\frac{1+\ln(k+1)}{k}\right) + \tfrac{1}{2}C_{h,Q}\cdot\bar{\alpha}$$

$$\leq \frac{\frac{1}{2}C_{h,Q}\,(1+2\ln(k+1))}{k} ,$$

where the second inequality uses *(ii)* of Proposition A.4 and the third inequality uses *(i)* of Proposition A.4. □

3.4 Extensions using Line-Searches

The original method of Frank and Wolfe [4] used a line-search to determine the next iterate λ_{k+1} by assigning $\hat{\alpha}_k \leftarrow \arg\max_{\alpha\in[0,1]}\{h(\lambda_k + \alpha(\tilde{\lambda}_k - \lambda_k))\}$ and $\lambda_{k+1} \leftarrow \lambda_k + \hat{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$. When $h(\cdot)$ is a quadratic and the dimension of the space E of variables λ is not huge, an exact line-search is

easy to compute analytically, otherwise an inexact line-search can be used. It is a straightforward extension of Theorem 2.1 to show that if an exact line-search is utilized at every iteration, then the bound (10) holds for *any* choice of step-size sequence $\{\bar{\alpha}_k\}$, and not just the sequence $\{\hat{\alpha}_k\}$ of line-search step-sizes. In particular, the $O(\frac{1}{k})$ computational guarantee (19) holds, as does (24) and (29), as well as the bound (38) to be developed in Section 4.

This observation generalizes as follows. At iteration k of the conditional gradient method, let $A_k \subseteq [0,1)$ be a closed set of potential step-sizes and suppose we select the next iterate λ_{k+1} by assigning $\hat{\alpha}_k \leftarrow \arg\max_{\alpha \in A_k}\{h(\lambda_k + \alpha(\tilde{\lambda}_k - \lambda_k))\}$ and $\lambda_{k+1} \leftarrow \lambda_k + \hat{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$. Then after k iterations of the conditional gradient method, we can apply the bound (10) for any choice of step-size sequence $\{\bar{\alpha}_i\}_{i=1}^k$ in the cross-product $A_1 \times \cdots \times A_k$.

4 Computational Guarantees for a Warm Start

In the framework of this study, the well-studied step-size sequence (18) and associated computational guarantees (Bound 3.1) corresponds to running the conditional gradient method initiated with the pre-start step from the initial point λ_0. One feature of the main computational guarantees as presented in the bounds (19) and (20) is their insensitivity to the quality of the initial point λ_0. This is good if $h(\lambda_0)$ is very far from the optimal value h^*, as the poor quality of the initial point does not affect the computational guarantee. But if $h(\lambda_0)$ is moderately close to the optimal value, one would want the conditional gradient method, with an appropriate step-size sequence, to have computational guarantees that reflect the closeness to optimality of the initial objective function value $h(\lambda_0)$. Let us see how this can be done.

We will consider starting the conditional gradient method *without* the pre-start step, started at an initial point λ_1, and let C_1 be a given *estimate* of the curvature constant $C_{h,Q}$. Consider the following step-size sequence:

$$\bar{\alpha}_i = \frac{2}{\frac{2C_1}{B_1 - h(\lambda_1)} + i + 1} \qquad \text{for } i \geq 1 \ . \tag{36}$$

Comparing (36) to the well-studied step-size rule (18), one can think of the above step-size rule as acting "as if" the conditional gradient method had run for $\frac{2C_1}{B_1 - h(\lambda_1)}$ iterations before arriving at λ_1. The next result presents a computational guarantee associated with this step-size rule.

Bound 4.1. *Under the step-size sequence (36), the following inequality holds for all $k \geq 1$:*

$$B_k - h(\lambda_{k+1}) \leq \frac{2\max\{C_1, C_{h,Q}\}}{\frac{2C_1}{B_1 - h(\lambda_1)} \ + \ k} \ . \tag{37}$$

Notice that in the case when $C_1 = C_{h,Q}$, the bound in (37) simplifies conveniently to:

$$B_k - h(\lambda_{k+1}) \leq \frac{2C_{h,Q}}{\frac{2C_{h,Q}}{B_1 - h(\lambda_1)} \ + \ k} \ . \tag{38}$$

Also, as a function of the estimate C_1 of the curvature constant, it is easily verified that the bound in (37) is optimized at $C_1 = C_{h,Q}$.

We remark that the bound (37) (or (38)) is small to the extent that the initial bound gap $B_1 - h(\lambda_1)$ is small, as one would want. However, to the extent that $B_1 - h(\lambda_1)$ is small, the incremental decrease in the bound due to an additional iteration is less. In other words, while the bound (37) is nicely sensitive to the initial bound gap, there is no longer rapid decrease in the bound in the early iterations. It is as if the algorithm had already run for $\left(\frac{2C_1}{B_1 - h(\lambda_1)}\right)$ iterations to arrive at the initial iterate λ_1, with a corresponding dampening in the marginal value of each iteration after then. This is a structural feature of the conditional gradient method that is different from first-order methods that use prox functions and/or projections.

Proof of Bound 4.1: Define $s = \frac{2C_1}{B_1 - h(\lambda_1)}$, whereby $\bar{\alpha}_i = \frac{2}{s+1+i}$ for $i \geq 1$. It then is straightforward to show that the dual averages sequences (9) are for $i \geq 1$:

$$\beta_i = \prod_{j=1}^{i-1} (1 - \bar{\alpha}_j)^{-1} = \prod_{j=1}^{i-1} \frac{s+j+1}{s+j-1} = \frac{(s+i-1)(s+i)}{s(s+1)} \ ,$$

and

$$\alpha_i = \frac{\beta_i \bar{\alpha}_i}{1 - \bar{\alpha}_i} = \frac{2(s+i)(s+i-1)(s+i+1)}{s(s+1)(s+i+1)(s+i-1)} = \frac{2(s+i)}{s(s+1)} \ .$$

Furthermore, we have:

$$\sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}} = \sum_{i=1}^{k} \frac{4(s+i)^2(s)(s+1)}{s^2(s+1)^2(s+i)(s+i+1)} = \sum_{i=1}^{k} \frac{4(s+i)}{s(s+1)(s+i+1)} \leq \frac{4k}{s(s+1)} \ . \tag{39}$$

Utilizing Theorem 2.1 and (39), we have for $k \geq 1$:

$$\begin{aligned}
B_k - h(\lambda_{k+1}) &\leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2} C_{h,Q} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \\
&\leq \frac{s(s+1)}{(s+k)(s+k+1)} \left(B_1 - h(\lambda_1) + \frac{C_{h,Q}}{2} \cdot \frac{4k}{s(s+1)} \right) \\
&= \frac{s(s+1)}{(s+k)(s+k+1)} \left(\frac{2C_1}{s} + \frac{2kC_{h,Q}}{s(s+1)} \right) \\
&\leq \frac{2 \max\{C_1, C_{h,Q}\}}{(s+k)(s+k+1)} (s+1+k) \\
&= \frac{2 \max\{C_1, C_{h,Q}\}}{s+k} \ ,
\end{aligned}$$

which completes the proof. □

4.1 A Dynamic Version of the Warm-Start Step-size Strategy

The step-size sequence (36) determines all step-sizes for the conditional gradient method based on two pieces of information at the initial point λ_1: *(i)* the initial bound gap $B_1 - h(\lambda_1)$, and *(ii)* the given estimate C_1 of the curvature constant. The step-size sequence (36) is a static warm-start strategy in that all step-sizes are determined by information that is available or computed at the first iterate. Let us see how we can improve the computational guarantee by treating every iterate as if it were the initial iterate, and hence dynamically determine the steps-size sequence as a function of accumulated information about the bound gap and the curvature constant.

At the start of a given iteration k of the conditional gradient method, we have the iterate value $\lambda_k \in Q$ and an upper bound B_{k-1} on h^* from the previous iteration. We also will now assume that we have an estimate C_{k-1} of the curvature constant from the previous iteration as well. Steps (2.) and (3.) of the conditional gradient method then perform the computation of $\tilde{\lambda}_k$, B_k and G_k. Instead of using a pre-set formula for the step-size $\bar{\alpha}_k$, we will determine the value of $\bar{\alpha}_k$ based on the current bound gap $B_k - h(\lambda_k)$ as well as on a new estimate C_k of the curvature constant. (We will shortly discuss how C_k is computed.) Assuming C_k has been computed, and mimicking the structure of the static warm-start step-size rule (36), we compute the current step-size as follows:

$$\bar{\alpha}_k := \frac{2}{\frac{2C_k}{B_k - h(\lambda_k)} + 2} \ , \tag{40}$$

where we note that $\bar{\alpha}_k$ depends explicitly on the value of C_k. Comparing $\bar{\alpha}_k$ in (40) with (18), we interpret $\frac{2C_k}{B_k - h(\lambda_k)}$ to be "as if" the current iteration k was preceded by $\frac{2C_k}{B_k - h(\lambda_k)}$ iterations of the conditional gradient method using the standard step-size (18). This interpretation is also in concert with that of the static warm-start step-size rule (36).

We now discuss how we propose to compute the new estimate C_k of the curvature constant $C_{h,Q}$ at iteration k. Because C_k will be only an estimate of $C_{h,Q}$, we will need to require that C_k (and the step-size $\bar{\alpha}_k$ (40) that depends explicitly on C_k) satisfy:

$$h(\lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)) \geq h(\lambda_k) + \bar{\alpha}_k(B_k - h(\lambda_k)) - \frac{1}{2}C_k\bar{\alpha}_k^2 \ . \tag{41}$$

In order to find a value $C_k \geq C_{k-1}$ for which (41) is satisfied, we first test if $C_k := C_{k-1}$ satisfies (41), and if so we set $C_k \leftarrow C_{k-1}$. If not, one can perform a standard doubling strategy, testing values $C_k \leftarrow 2C_{k-1}, 4C_{k-1}, 8C_{k-1}, \ldots$, until (41) is satisfied. Since (41) will be satisfied whenever $C_k \geq C_{h,Q}$ from the definition of $C_{h,Q}$ in (7) and the inequality $B_k - h(\lambda_k) \leq B_k^w - h(\lambda_k) = \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$, it follows that the doubling strategy will guarantee $C_k \leq \max\{C_0, 2C_{h,Q}\}$. Of course, if an upper bound $\bar{C} \geq C_{h,Q}$ is known, then $C_k \leftarrow \bar{C}$ is a valid assignment for all $k \geq 1$. Moreover, the structure of $h(\cdot)$ may be sufficiently simple so that a value of $C_k \geq C_{k-1}$ satisfying (41) can be determined analytically via closed-form calculation, as is the case if $h(\cdot)$ is a quadratic function for example. The formal description of the conditional gradient method with dynamic step-size strategy is presented in Method 3.

We have the following computational guarantees for the conditional gradient method with dynamic step-sizes (Method 3):

Method 3 Conditional Gradient Method with Dynamic Step-sizes for maximizing $h(\lambda)$

Initialize at $\lambda_1 \in Q$, initial estimate C_0 of $C_{h,Q}$, (optional) initial upper bound B_0, $k \leftarrow 1$.
At iteration k:

1. Compute $\nabla h(\lambda_k)$.
2. Compute $\tilde{\lambda}_k \leftarrow \arg\max_{\lambda \in Q} \{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\}$.
   $B_k^w \leftarrow h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$.
   $G_k \leftarrow \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$.
3. (Optional: compute other upper bound B_k^o), update best bound $B_k \leftarrow \min\{B_{k-1}, B_k^w, B_k^o\}$.
4. Compute C_k for which the following conditions hold:
   (i) $C_k \geq C_{k-1}$, and
   (ii) $h(\lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)) \geq h(\lambda_k) + \bar{\alpha}_k(B_k - h(\lambda_k)) - \frac{1}{2}C_k\bar{\alpha}_k^2$, where $\bar{\alpha}_k := \frac{2}{\frac{2C_k}{B_k - h(\lambda_k)} + 2}$.
5. Set $\lambda_{k+1} \leftarrow \lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$.

Bound 4.2. *The iterates of the conditional gradient method with dynamic step-sizes (Method 3) satisfy the following for any $k \geq 1$:*

$$B_k - h(\lambda_k) \leq \min_{\ell \in \{1,\ldots,k\}} \left\{ \frac{2C_k}{\frac{2C_k}{B_\ell - h(\lambda_\ell)} \; + \; k - \ell} \right\} . \tag{42}$$

Furthermore, if the doubling strategy is used to update the estimates $\{C_k\}$ of $C_{h,Q}$, it holds that $C_k \leq \max\{C_0, 2C_{h,Q}\}$.

Notice that (42) naturally generalizes the static warm-start bound (37) (or (38)) to this more general dynamic case. Consider, for simplicity, the case where $C_k = C_{h,Q}$ is the known curvature constant. In this case, (42) says that we may apply the bound (38) with any $\ell \in \{1, \ldots, k\}$ as the starting iteration. That is, the computational guarantee for the dynamic case is at least as good as the computational guarantee for the static warm-start step-size (36) initialized at *any* iteration $\ell \in \{1, \ldots, k\}$.

Proof of Bound 4.2: Let $i \geq 1$. For convenience define $A_i = \frac{2C_i}{B_i - h(\lambda_i)}$, and in this notation (40)

is $\bar{\alpha}_i = \frac{2}{A_i+2}$. Applying *(ii)* in Step (4.) of Method 3 we have:

$$\begin{aligned}
B_{i+1} - h(\lambda_{i+1}) &\leq B_{i+1} - h(\lambda_i) - \bar{\alpha}_i(B_i - h(\lambda_i)) + \tfrac{1}{2}\bar{\alpha}_i^2 C_i \\
&\leq B_i - h(\lambda_i) - \bar{\alpha}_i(B_i - h(\lambda_i)) + \tfrac{1}{2}\bar{\alpha}_i^2 C_i \\
&= (B_i - h(\lambda_i))(1 - \bar{\alpha}_i) + \tfrac{1}{2}\bar{\alpha}_i^2 C_i \\
&= \frac{2C_i}{A_i}\left(\frac{A_i}{A_i+2}\right) + \frac{2C_i}{(A_i+2)^2} \\
&= 2C_i\left(\frac{A_i+3}{(A_i+2)^2}\right) \\
&< \frac{2C_i}{A_i+1} \ ,
\end{aligned}$$

where the last inequality follows from the fact that $(a+2)^2 > a^2+4a+3 = (a+1)(a+3)$ for $a \geq 0$. Therefore

$$A_{i+1} = \frac{2C_{i+1}}{B_{i+1} - h(\lambda_{i+1})} = \frac{C_{i+1}}{C_i}\left(\frac{2C_i}{B_{i+1} - h(\lambda_{i+1})}\right) > \frac{C_{i+1}}{C_i}\left(A_i + 1\right) \ . \tag{43}$$

We now show by reverse induction that for any $\ell \in \{1, \ldots, k\}$ the following inequality is true:

$$A_k \geq \frac{C_k}{C_\ell} A_\ell + k - \ell \ . \tag{44}$$

Clearly (44) holds for $\ell = k$, so let us suppose (44) holds for some $\ell + 1 \in \{2, \ldots, k\}$. Then

$$\begin{aligned}
A_k &\geq \frac{C_k}{C_{\ell+1}} A_{\ell+1} + k - \ell - 1 \\
&> \frac{C_k}{C_{\ell+1}}\left(\frac{C_{\ell+1}}{C_\ell}\left(A_\ell + 1\right)\right) + k - \ell - 1 \\
&\geq \frac{C_k}{C_\ell} A_\ell + k - \ell \ ,
\end{aligned}$$

where the first inequality is the induction hypothesis, the second inequality uses (43), and the third inequality uses the monotonicity of the $\{C_k\}$ sequence. This proves (44). Now for any $\ell \in \{1, \ldots, k\}$ we have from (44) that:

$$B_k - h(\lambda_k) = \frac{2C_k}{A_k} \leq \frac{2C_k}{\frac{C_k}{C_\ell} A_\ell + k - \ell} = \frac{2C_k}{\frac{2C_k}{B_\ell - h(\lambda_\ell)} + k - \ell} \ ,$$

proving the result. □

5 Analysis of the Conditional Gradient Method with Inexact Gradient Computations and/or Subproblem Solutions

In this section we present and analyze extensions of the conditional gradient method in the presence of inexact computation of gradients and/or subproblem solutions. We first consider the case when the linear optimization subproblem is solved approximately.

5.1 Conditional Gradient Method with Inexact Linear Optimization Subproblem Solutions

Here we consider the case when the linear optimization subproblem is solved approximately, which arises especially in optimization over matrix variables. For example, consider instances of (1) where Q is the spectrahedron of symmetric matrices, namely $Q = \{\Lambda \in \mathbb{S}^{n\times n} : \Lambda \succeq 0,\ I \bullet \Lambda = 1\}$, where $\mathbb{S}^{n\times n}$ is the space of symmetric matrices of order n, "$\succeq$" is the Löwner ordering thereon, and "$\cdot \bullet \cdot$" denotes the trace inner product. For these instances solving the linear optimization subproblem corresponds to computing the leading eigenvector of a symmetric matrix, whose solution when $n \gg 0$ is typically computed inexactly using iterative methods. For $\delta \geq 0$ an (absolute) δ-approximate solution to the linear optimization subproblem $\max_{\lambda\in Q} \left\{c^T\lambda\right\}$ is a vector $\tilde{\lambda} \in Q$ satisfying:

$$c^T\tilde{\lambda} \geq \max_{\lambda\in Q} \left\{c^T\lambda\right\} - \delta \ , \tag{45}$$

and we use the notation $\tilde{\lambda} \leftarrow \text{approx}(\delta)_{\lambda\in Q} \left\{c^T\lambda\right\}$ to denote assigning to $\tilde{\lambda}$ any such δ-approximate solution. In Method 4 we present a version of the conditional gradient algorithm that uses approximate linear optimization subproblem solutions. Note that Method 4 allows for the approximation quality $\delta = \delta_k$ to be a function of the iteration index k. Note also that the definition of the Wolfe upper bound B_k^w and the Wolfe gap G_k in Step (2.) are amended from the original conditional gradient algorithm (Method 1) by an additional term δ_k. It follows from (45) that:

$$B_k^w = h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) + \delta_k \geq \max_{\lambda\in Q} \left\{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\right\} \geq h^* \ ,$$

which shows that B_k^w is a valid upper bound on h^*, with similar properties for G_k. The following two theorems extend Theorem 2.1 and Theorem 2.2 to the case of approximate subproblem solutions. Analogous to the the case of exact subproblem solutions, these two theorems can easily be used to derive suitable bounds for specific step-sizes rules such as those in Sections 3 and 4.

Method 4 Conditional Gradient Method with Approximate Subproblem Solutions

Initialize at $\lambda_1 \in Q$, (optional) initial upper bound B_0, $k \leftarrow 1$.
At iteration k:
1. Compute $\nabla h(\lambda_k)$.
2. Compute $\tilde{\lambda}_k \leftarrow \text{approx}(\delta_k)_{\lambda\in Q}\{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\}$.
   $B_k^w \leftarrow h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) + \delta_k$.
   $G_k \leftarrow \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) + \delta_k$.
3. (Optional: compute other upper bound B_k^o), update best bound $B_k \leftarrow \min\{B_{k-1}, B_k^w, B_k^o\}$.
4. Set $\lambda_{k+1} \leftarrow \lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$, where $\bar{\alpha}_k \in [0, 1)$.

Theorem 5.1. *Consider the iterate sequences of the conditional gradient method with approximate subproblem solutions (Method 4) $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$ and the sequence of upper bounds $\{B_k\}$ on h^*, using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$ given by (9), and for any $k \geq 0$, the following inequality holds:*

$$B_k - h(\lambda_{k+1}) \leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} + \frac{\sum_{i=1}^{k}\alpha_i\delta_i}{\beta_{k+1}} \ . \tag{46}$$

□

Theorem 5.2. *Consider the iterate sequences of the conditional gradient method with approximate subproblem solutions (Method 4) $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$, the sequence of upper bounds $\{B_k\}$ on h^*, and the sequence of Wolfe gaps $\{G_k\}$ from Step (2.), using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$ given by (9), and for any $\ell \geq 0$ and $k \geq \ell + 1$, the following inequality holds:*

$$\begin{aligned}\min_{i\in\{\ell+1,\ldots,k\}} G_i \leq \ & \frac{1}{\sum_{i=\ell+1}^{k}\bar{\alpha}_i}\left[\frac{B_\ell - h(\lambda_1)}{\beta_{\ell+1}} + \frac{\frac{1}{2}C_{h,Q}\sum_{i=1}^{\ell}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{\ell+1}} + \frac{\sum_{i=1}^{\ell}\alpha_i\delta_i}{\beta_{\ell+1}}\right] \\ & + \frac{\frac{1}{2}C_{h,Q}\sum_{i=\ell+1}^{k}\bar{\alpha}_i^2}{\sum_{i=\ell+1}^{k}\bar{\alpha}_i} + \frac{\sum_{i=\ell+1}^{k}\bar{\alpha}_i\delta_i}{\sum_{i=\ell+1}^{k}\bar{\alpha}_i} \ .\end{aligned} \tag{47}$$

□

Remark 5.1. *The pre-start step (Procedure 2 can also be generalized to the case of approximate solution of the linear optimization subproblem. Let λ_1 and B_0 be computed by the pre-start step with a $\delta = \delta_0$-approximate subproblem solution. Then Proposition 3.1 generalizes to:*

$$B_0 - h(\lambda_1) \leq \frac{1}{2}C_{h,Q} + \delta_0 \ ,$$

and hence if the pre-start step is used (46) *implies that:*

$$B_k - h(\lambda_{k+1}) \leq \frac{\frac{1}{2}C_{h,Q}\sum_{i=0}^{k}\frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} + \frac{\sum_{i=0}^{k}\alpha_i\delta_i}{\beta_{k+1}} \ , \tag{48}$$

where $\alpha_0 := 1$.

Let us now discuss implications of Theorems 5.1 and 5.2, and Remark 5.1. Observe that the bounds on the right-hand sides of (46) and (47) are composed of the exact terms which appear on the right-hand sides of (10) and (11), plus additional terms involving the solution accuracy sequence $\delta_1, \ldots, \delta_k$. It follows from (14) that these latter terms are particular convex combinations of the δ_i values and zero, and in (48) the last term is a convex combination of the δ_i values, whereby they are trivially bounded above by $\max\{\delta_1, \ldots, \delta_k\}$. When $\delta_i := \delta$ is a constant, then this bound is simply δ, and we see that the errors due to the approximate computation of linear optimization subproblem solutions do not accumulate, independent of the choice of step-size sequence $\{\bar{\alpha}_k\}$. In other words, Theorem 5.1 implies that if we are able to solve the linear optimization subproblems to an accuracy of δ, then the conditional gradient method can solve (1) to an accuracy of δ plus

a function of the step-size sequence $\{\bar{\alpha}_k\}$, the latter of which can be made to go to zero at an appropriate rate depending on the choice of step-sizes. Similar observations hold for the terms depending on $\delta_1, \ldots, \delta_k$ that appear on the right-hand side of (47).

Note that Jaggi [11] considers the case where $\delta_i := \frac{1}{2}\bar{\alpha}_i C_{h,Q}$ and $\bar{\alpha}_i := \frac{2}{i+2}$ for $i \geq 0$ (or $\bar{\alpha}_i$ is determined by a line-search), and shows that in this case Method 4 achieves $O\left(\frac{1}{k}\right)$ convergence in terms of both the optimality gap and the Wolfe gaps. These results can be recovered as a particular instantiation of Theorems 5.1 and 5.2 using similar logic as in the proof of Bound 3.1.

Proof of Theorem 5.1: First recall the identities (13) and (14) for the dual averages sequences (9). Following the proof of Theorem 2.1, we then have for $i \geq 1$:

$$\begin{aligned}
\beta_{i+1} h(\lambda_{i+1}) &\geq \beta_{i+1}\left[h(\lambda_i) + \nabla h(\lambda_i)^T(\tilde{\lambda}_i - \lambda_i)\bar{\alpha}_i - \frac{1}{2}\bar{\alpha}_i^2 C_{h,Q}\right] \\
&= \beta_i h(\lambda_i) + (\beta_{i+1} - \beta_i)h(\lambda_i) + \beta_{i+1}\bar{\alpha}_i \nabla h(\lambda_i)^T(\tilde{\lambda}_i - \lambda_i) - \frac{1}{2}\beta_{i+1}\bar{\alpha}_i^2 C_{h,Q} \\
&= \beta_i h(\lambda_i) + \alpha_i\left[h(\lambda_i) + \nabla h(\lambda_i)^T(\tilde{\lambda}_i - \lambda_i)\right] - \frac{1}{2}\frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \\
&= \beta_i h(\lambda_i) + \alpha_i B_i^w - \alpha_i \delta_i - \frac{1}{2}\frac{\alpha_i^2}{\beta_{i+1}} C_{h,Q} \ ,
\end{aligned}$$

where the third equality above uses the definition of the Wolfe upper bound (2) in Method 4. The rest of the proof follows exactly as in the proof of Theorem 2.1. □

Proof of Theorem 5.2: For $i \geq 1$ we have:

$$\begin{aligned}
h(\lambda_{i+1}) &\geq h(\lambda_i) + \nabla h(\lambda_i)^T(\tilde{\lambda}_i - \lambda_i)\bar{\alpha}_i - \tfrac{1}{2}\bar{\alpha}_i^2 C_{h,Q} \\
&= h(\lambda_i) + \bar{\alpha}_i G_i - \bar{\alpha}_i \delta_i - \tfrac{1}{2}\bar{\alpha}_i^2 C_{h,Q} \ ,
\end{aligned}$$

where the equality above follows from the definition of the Wolfe gap in Method 4. Summing the above over $i \in \{\ell+1, \ldots, k\}$ and rearranging yields:

$$\sum_{i=\ell+1}^k \bar{\alpha}_i G_i \leq h(\lambda_{k+1}) - h(\lambda_{\ell+1}) + \sum_{i=\ell+1}^k \tfrac{1}{2}\bar{\alpha}_i^2 C_{h,Q} + \sum_{i=\ell+1}^k \bar{\alpha}_i \delta_i \ . \tag{49}$$

The rest of the proof follows by combining (49) with Theorem 5.1 and proceeding as in the proof of Theorem 2.2. □

5.2 Conditional Gradient Method with Inexact Gradient Computations

We now consider a version of the conditional gradient method where the exact gradient computation is replaced with the computation of an approximate gradient. We analyze two different models of approximate gradients and derive computational guarantees for each model. We first consider the δ-oracle model of d'Aspremont [2], which was developed in the context of accelerated first-order methods. For $\delta \geq 0$, a δ-oracle is a (possibly non-unique) mapping $g_\delta(\cdot) : Q \to E^*$ that satisfies:

$$\left|(\nabla h(\bar{\lambda}) - g_\delta(\bar{\lambda}))^T(\lambda - \bar{\lambda})\right| \leq \delta \quad \text{for all } \lambda, \bar{\lambda} \in Q \ . \tag{50}$$

Note that the definition of the δ-oracle does not consider inexact computation of function values. Depending on the choice of step-size sequence $\{\bar{\alpha}_k\}$, this assumption is acceptable as the conditional

gradient method may or may not need to compute function values. (The warm-start step-size rule (40) requires computing function values, as does the computation of the Wolfe bounds $\{B_k^w\}$, in which case a definition analogous to (50) for function values can be utilized.)

The next proposition states the following: suppose one solves for the exact solution of the linear optimization subproblem using the δ-oracle instead of the exact gradient. Then the absolute suboptimality of the computed solution in terms of the exact gradient is at most 2δ.

Proposition 5.1. *For any $\bar{\lambda} \in Q$ and any $\delta \geq 0$, if $\tilde{\lambda} \in \arg\max_{\lambda \in Q} \{g_\delta(\bar{\lambda})^T \lambda\}$, then $\tilde{\lambda}$ is a 2δ-approximate solution to the linear optimization subproblem $\max_{\lambda \in Q} \{\nabla h(\bar{\lambda})^T \lambda\}$.*

Proof. Let $\hat{\lambda} \in \arg\max_{\lambda \in Q} \{\nabla h(\bar{\lambda})^T \lambda\}$. Then, we have:

$$\begin{aligned}
\nabla h(\bar{\lambda})^T(\tilde{\lambda} - \bar{\lambda}) &\geq g_\delta(\bar{\lambda})^T(\tilde{\lambda} - \bar{\lambda}) - \delta \\
&\geq g_\delta(\bar{\lambda})^T(\hat{\lambda} - \bar{\lambda}) - \delta \\
&\geq \nabla h(\bar{\lambda})^T(\hat{\lambda} - \bar{\lambda}) - 2\delta \\
&= \max_{\lambda \in Q} \{\nabla h(\bar{\lambda})^T \lambda\} - \nabla h(\bar{\lambda})^T \bar{\lambda}) - 2\delta \ ,
\end{aligned}$$

where the first and third inequalities use (50), the second inequality follows since $\tilde{\lambda} \in \arg\max_{\lambda \in Q} \{g_\delta(\bar{\lambda})^T \lambda\}$, and the fourth inequality follows since $\hat{\lambda} \in \arg\max_{\lambda \in Q} \{\nabla h(\bar{\lambda})^T \lambda\}$. Rearranging terms then yields the result. □

Now consider a version of the conditional gradient method where the computation of $\nabla h(\lambda_k)$ at Step (1.) is replaced with the computation of $g_{\delta_k}(\lambda_k)$. Then Proposition 5.1 implies that such a version can be viewed simply as a special case of the version of the conditional gradient method with approximate subproblem solutions (Method 4) of Section 5.1 with δ_k replaced by $2\delta_k$. Thus, we may readily apply Theorems 5.1 and 5.2 and Proposition 5.1 to this case. In particular, similar to the results in [2] regarding error non-accumulation for an accelerated first-order method, the results herein imply that there is no accumulation of errors for a version of the conditional gradient method that computes approximate gradients with a δ-oracle at each iteration. Furthermore, it is a simple extension to consider a version of the conditional gradient method that computes both (i) approximate gradients with a δ-oracle, and (ii) approximate linear optimization subproblem solutions.

5.2.1 Inexact Gradient Computation Model via the (δ, L)-oracle

The premise (50) underlying the δ-oracle is quite strong and can be restrictive in many cases. For this reason among others, Devolder et al. [3] introduce the less restrictive (δ, L)-oracle model. For scalars $\delta, L \geq 0$, the (δ, L)-oracle is defined as a (possibly non-unique) mapping $Q \to \mathbb{R} \times E^*$ that maps $\bar{\lambda} \to (h_{(\delta,L)}(\bar{\lambda}), g_{(\delta,L)}(\bar{\lambda}))$ which satisfy:

$$h(\lambda) \leq h_{(\delta,L)}(\bar{\lambda}) + g_{(\delta,L)}(\bar{\lambda})^T(\lambda - \bar{\lambda}) \ , \quad \text{and} \tag{51}$$

$$h(\lambda) \geq h_{(\delta,L)}(\bar{\lambda}) + g_{(\delta,L)}(\bar{\lambda})^T(\lambda - \bar{\lambda}) - \frac{L}{2}\|\lambda - \bar{\lambda}\|^2 - \delta \quad \text{for all } \lambda, \bar{\lambda} \in Q \ , \tag{52}$$

where $\|\cdot\|$ is a choice of norm on E. Note that in contrast to the δ-oracle model, the (δ, L)-oracle model does assume that the function $h(\cdot)$ is smooth or even concave – it simply assumes that there is an oracle returning the pair $(h_{(\delta,L)}(\bar{\lambda}), g_{(\delta,L)}(\bar{\lambda}))$ satisfying (51) and (52).

In Method 5 we present a version of the conditional gradient method that utilizes the (δ, L)-oracle. Note that we allow the parameters δ and L of the (δ, L)-oracle to be a function of the iteration index k. Inequality (51) in the definition of the (δ, L)-oracle immediately implies that $B_k^w \geq h^*$. We now state the main technical complexity bound for Method 5, in terms of the sequence of bound gaps $\{B_k - h(\lambda_{k+1})\}$. Recall from Section 2 the definition $\text{Diam}_Q := \max_{\lambda,\bar{\lambda}\in Q}\{\|\lambda - \bar{\lambda}\|\}$, where the norm $\|\cdot\|$ is the norm used in the definition of the (δ, L)-oracle (52).

Method 5 Conditional Gradient Method With (δ, L)-Oracle

Initialize at $\lambda_1 \in Q$, (optional) initial upper bound B_0, $k \leftarrow 1$.

At iteration k:

1. Compute $h_k \leftarrow h_{(\delta_k,L_k)}(\lambda_k),\ g_k \leftarrow g_{(\delta_k,L_k)}(\lambda_k)$.
2. Compute $\tilde{\lambda}_k \leftarrow \arg\max_{\lambda\in Q}\{h_k + g_k^T(\lambda - \lambda_k)\}$.

   $B_k^w \leftarrow h_k + g_k^T(\tilde{\lambda}_k - \lambda_k)$.
3. (Optional: compute other upper bound B_k^o), update best bound $B_k \leftarrow \min\{B_{k-1}, B_k^w, B_k^o\}$.
4. Set $\lambda_{k+1} \leftarrow \lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$, where $\bar{\alpha}_k \in [0, 1)$.

Theorem 5.3. *Consider the iterate sequences of the conditional gradient method with the (δ, L)-oracle (Method 5) $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$ and the sequence of upper bounds $\{B_k\}$ on h^*, using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$ given by (9), and for any $k \geq 0$, the following inequality holds:*

$$B_k - h(\lambda_{k+1}) \leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{\frac{1}{2}\text{Diam}_Q^2 \sum_{i=1}^k L_i \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} + \frac{\sum_{i=1}^k \beta_{i+1}\delta_i}{\beta_{k+1}} \ . \tag{53}$$

□

As with Theorem 5.1, observe that the terms on the right-hand side of (53) are composed of the exact terms which appear on the right-hand side of (10), plus an additional term that is a function of $\delta_1, \ldots, \delta_k$. Unfortunately, Theorem 5.3 implies an accumulation of errors for Method 5 under essentially any choice of step-size sequence $\{\bar{\alpha}_k\}$. Indeed, suppose that $\beta_i = O(i^\gamma)$ for some $\gamma \geq 0$, then $\sum_{i=1}^k \beta_{i+1} = O(k^{\gamma+1})$, and in the constant case where $\delta_i := \delta$, we have $\frac{\sum_{i=1}^k \beta_{i+1}\delta_i}{\beta_{k+1}} = O(k\delta)$. Therefore in order to achieve an $O\left(\frac{1}{k}\right)$ rate of convergence (for example with the step-size sequence (18)) we need $\delta = O\left(\frac{1}{k^2}\right)$. This negative result nevertheless contributes to the understanding of the merits and demerits of different first-order methods as follows. Note that in [3] it is shown that the "classical" gradient methods (both primal and dual), which require solving a proximal projection problem at each iteration, achieve an $O\left(\frac{1}{k} + \delta\right)$ accuracy under the (δ, L)-oracle model for constant (δ, L). On the other hand, it is also shown in [3] that all accelerated first-order methods (which also solve proximal projection problems at each iteration) generically achieve an $O\left(\frac{1}{k^2} + k\delta\right)$ accuracy and thus suffer from an accumulation of errors under the (δ, L)-oracle model. As discussed in

Method/ Class	Type of Subproblem	Accuracy with Exact Gradients	Accuracy with (δ, L)-oracle	Special Structure of Iterates
Conditional Gradient	Linear Optimization	$O(1/k)$	$O(1/k) + O(\delta k)$	Yes
Classical Gradient	Prox Projection	$O(1/k)$	$O(1/k) + O(\delta)$	No
Accelerated Gradient	Prox Projection	$O\left(1/k^2\right)$	$O\left(1/k^2\right) + O(\delta k)$	No

Figure 1: Properties of three (classes of) first-order methods after k iterations.

the Introduction herein, the conditional gradient method offers two possible advantages over these proximal methods: (i) the possibility that solving the linear optimization subproblem is easier than the projection-type problem in an iteration of a proximal method, (ii) the possibility of greater structure (sparsity, low rank) of the iterates. In Figure 1 we summarize the cogent properties of these three methods (or classes of methods) under exact gradient computation as well as with the (δ, L)-oracle model. As can be seen from the table in Figure 1, no single method dominates in the three categories of properties shown in the table; thus there are inherent tradeoffs among these methods/classes.

Proof of Theorem 5.3: Note that (51) and (52) with $\bar{\lambda} = \lambda$ imply that:

$$h(\lambda) \leq h_{(\delta,L)}(\lambda) \leq h(\lambda) + \delta \quad \text{for all } \lambda \in Q \ . \tag{54}$$

Recall properties (13) and (14) of the dual averages sequences (9). Following the proof of Theorem 2.1, we then have for $i \geq 1$:

$$\begin{aligned}
\beta_{i+1} h(\lambda_{i+1}) &\geq \beta_{i+1} \left[h_i + g_i^T(\tilde{\lambda}_i - \lambda_i)\bar{\alpha}_i - \frac{1}{2}\bar{\alpha}_i^2 L_i \mathrm{Diam}_Q^2 - \delta_i \right] \\
&= \beta_i h_i + (\beta_{i+1} - \beta_i) h_i + \beta_{i+1}\bar{\alpha}_i g_i^T(\tilde{\lambda}_i - \lambda_i) - \frac{1}{2}\beta_{i+1}\bar{\alpha}_i^2 L_i \mathrm{Diam}_Q^2 - \beta_{i+1}\delta_i \\
&= \beta_i h_i + \alpha_i \left[h_i + g_i^T(\tilde{\lambda}_i - \lambda_i) \right] - \frac{1}{2}\frac{\alpha_i^2}{\beta_{i+1}} L_i \mathrm{Diam}_Q^2 - \beta_{i+1}\delta_i \\
&\geq \beta_i h(\lambda_i) + \alpha_i B_i^w - \frac{1}{2}\frac{\alpha_i^2}{\beta_{i+1}} L_i \mathrm{Diam}_Q^2 - \beta_{i+1}\delta_i \ ,
\end{aligned}$$

where the first inequality uses (52), and the second inequality uses (54) and the definition of the Wolfe upper bound in Method 5. The rest of the proof follows as in the proof of Theorem 2.1. □

References

[1] K.L. Clarkson, *Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm*, 19th ACM-SIAM Symposium on Discrete Algorithms (2008), 922–931.

[2] A. d'Aspremont, *Smooth optimization with approximate gradient*, SIAM Journal on Optimization **19** (2008), no. 3, 1171–1183.

[3] O. Devolder, F. Glineur, and Y.E. Nesterov, *First-order methods of smooth convex optimization with inexact oracle*, Tech. report, CORE, Louvain-la-Neuve, Belgium, 2013.

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

[5] R.M. Freund, P. Grigas, and R. Mazumder, *A first-order view of boosting methods, with implications for regularization and computational guarantees for loss minimization algorithms*, Tech. report, MIT Operations Research Center, in preparation, 2013.

[6] J. Giesen, M. Jaggi, and S. Laue, *Optimizing over the growing spectrahedron*, ESA 2012: 20th Annual European Symposium on Algorithms (2012).

[7] P. Grigas, *Dual averaging as a unifying framework in first-order methods*, Tech. report, MIT Operations Research Center, in preparation, 2013.

[8] Z. Harchaoui, A. Juditsky, and A. Nemirovski, *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Technical Report, 2013.

[9] E. Hazan, *Sparse approximate solutions to semidefinite programs*, Proceedings of Theoretical Informatics, 8th Latin American Symposium (LATIN) (2008), 306–316.

[10] M. Jaggi, *Convex optimization without projection steps*, Technical Report arXiv:1108.1170v6, 2011.

[11] M. Jaggi, *Sparse convex optimization methods for machine learning*, Ph.D. thesis, ETH Zurich, October 2011.

[12] L. Khachiyan, *Rounding of polytopes in the real number model of computation*, Mathematics of Operations Research **21** (1996), no. 2, 307–320.

[13] G. Lan, *The complexity of large-scale convex programming under a linear optimization oracle*, Tech. report, Department of Industrial and Systems Engineering, University of Florida, Gainesville, Florida, 2013.

[14] E. Levitin and B. Polyak, *Constrained minimization methods*, USSR Computational Mathematics and Mathematical Physics **6** (1966), 1.

[15] A. Nemirovski, *private communication*, (2007).

[16] Y.E. Nesterov, *Introductory lectures on convex optimization: a basic course*, Applied Optimization, vol. 87, Kluwer Academic Publishers, Boston, 2003.

[17] Y.E. Nesterov, *Smooth minimization of non-smooth functions*, Mathematical Programming **103** (2005), no. 1, 127–152.

[18] Y.E. Nesterov, *Primal-dual subgradient methods for convex problems*, Mathematical Programming **120** (2009), 221–259.

[19] B. Polyak, *Introduction to optimization*, Optimization Software, Inc., New York, 1987.

A Appendix

Proposition A.1. *Let B_k^w and B_k^m be as defined in Section 2. Suppose that there exists an open set $\hat{Q} \subseteq E$ containing Q such that $\phi(x,\cdot)$ is differentiable on $\hat{Q}$ for each fixed $x \in P$, and that $h(\cdot)$ has the minmax structure* (4) *on $\hat{Q}$ and is differentiable on $\hat{Q}$. Then it holds that:*

$$B_k^w \geq B_k^m \geq h^* \ .$$

Furthermore, it holds that $B_k^w = B_k^m$ in the case when $\phi(x,\cdot)$ is linear in the variable λ.

Proof. It is simple to show that $B_k^m \geq h^*$. At the current iterate $\lambda_k \in Q$, define $x_k \in \arg\min_{x \in P} \phi(x, \lambda_k)$. Then from the definition of $h(\lambda)$ and the concavity of $\phi(x_k,\cdot)$ we have:

$$h(\lambda) \leq \phi(x_k,\lambda) \leq \phi(x_k,\lambda_k) + \nabla_\lambda \phi(x_k,\lambda_k)^T(\lambda - \lambda_k) = h(\lambda_k) + \nabla_\lambda \phi(x_k,\lambda_k)^T(\lambda - \lambda_k) \ , \tag{55}$$

whereby $\nabla_\lambda \phi(x_k,\lambda_k)$ is a subgradient of $h(\cdot)$ at λ_k. It then follows from the differentiability of $h(\cdot)$ that $\nabla h(\lambda_k) = \nabla_\lambda \phi(x_k,\lambda_k)$, and this implies from (55) that:

$$\phi(x_k,\lambda) \leq h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k) \ . \tag{56}$$

Therefore we have:

$$B_k^m = f(x_k) = \max_{\lambda \in Q}\{\phi(x_k,\lambda)\} \leq \max_{\lambda \in Q}\{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\} = B_k^w \ .$$

If $\phi(x,\lambda)$ is linear in λ, then the second inequality in (55) is an equality, as is (56). □

Proposition A.2. *Let $C_{h,Q}$, Diam_Q, and $L_{h,Q}$ be as defined in Section 2. Then it holds that $C_{h,Q} \leq L_{h,Q}(\mathrm{Diam}_Q)^2$.*

Proof. Since Q is convex, we have $\lambda + \alpha(\tilde{\lambda} - \lambda) \in Q$ for all $\lambda, \tilde{\lambda} \in Q$ and for all $\alpha \in [0,1]$. Since the gradient of $h(\cdot)$ is Lipschitz, from the fundamental theorem of calculus we have:

$$\begin{aligned}
h(\lambda + \alpha(\tilde{\lambda} - \lambda)) &= h(\lambda) + \nabla h(\lambda)^T(\alpha(\tilde{\lambda} - \lambda)) + \int_0^1 [\nabla h(\lambda + t\alpha(\tilde{\lambda} - \lambda)) - \nabla h(\lambda)]^T(\alpha(\tilde{\lambda} - \lambda))dt \\
&\geq h(\lambda) + \nabla h(\lambda)^T(\alpha(\tilde{\lambda} - \lambda)) - \int_0^1 \|\nabla h(\lambda + t\alpha(\tilde{\lambda} - \lambda)) - \nabla h(\lambda)\|_*(\alpha)\|\tilde{\lambda} - \lambda\|dt \\
&\geq h(\lambda) + \nabla h(\lambda)^T(\alpha(\tilde{\lambda} - \lambda)) - \int_0^1 L_{h,Q}\|(t\alpha)(\tilde{\lambda} - \lambda)\|(\alpha)\|\tilde{\lambda} - \lambda\|dt \\
&= h(\lambda) + \nabla h(\lambda)^T(\alpha(\tilde{\lambda} - \lambda)) - \frac{\alpha^2}{2} L_{h,Q}\|(\tilde{\lambda} - \lambda)\|^2 \\
&\geq h(\lambda) + \nabla h(\lambda)^T(\alpha(\tilde{\lambda} - \lambda)) - \frac{\alpha^2}{2} L_{h,Q}(\mathrm{Diam}_Q)^2 \ ,
\end{aligned}$$

whereby it follows that $C_{h,Q} \leq L_{h,Q}(\mathrm{Diam}_Q)^2$. □

Proposition A.3. *For $k \geq 0$ the following inequality holds:*

$$\sum_{i=0}^{k} \frac{i+1}{i+2} \leq \frac{(k+1)(k+2)}{k+4} \ .$$

Proof. The inequality above holds at equality for $k = 0$. By induction, suppose the inequality is true for some given $k \geq 0$, then

$$\begin{aligned} \textstyle\sum_{i=0}^{k+1} \frac{i+1}{i+2} &= \textstyle\sum_{i=0}^{k} \frac{i+1}{i+2} + \frac{k+2}{k+3} \\ &\leq \tfrac{(k+1)(k+2)}{k+4} + \tfrac{k+2}{k+3} \\ &= (k+2)\left[\tfrac{k^2+5k+7}{k^2+7k+12}\right] \ . \end{aligned} \tag{57}$$

Now notice that

$$(k^2+5k+7)(k+5) = k^3+10k^2+32k+35 < k^3+10k^2+33k+36 = (k^2+7k+12)(k+3) \ ,$$

which combined with (57) completes the induction. □

Proposition A.4. *For $k \geq 1$ let $\bar{\alpha} := 1 - \frac{1}{\sqrt[k]{k+1}}$. Then the following inequalities holds:*

(i) $\dfrac{\ln(k+1)}{k} \geq \bar{\alpha}$ *, and*

(ii) $(k+1)\bar{\alpha} \geq 1$ *.*

Proof. To prove *(i)*, define $f(t) := 1 - e^{-t}$, and noting that $f(\cdot)$ is a concave function, the gradient inequality for $f(\cdot)$ at $t = 0$ is

$$t \geq 1 - e^{-t} \ .$$

Substituting $t = \frac{\ln(k+1)}{k}$ yields

$$\frac{\ln(k+1)}{k} = t \geq 1 - e^{-t} = 1 - e^{-\frac{\ln(k+1)}{k}} = 1 - \frac{1}{\sqrt[k]{k+1}} = \bar{\alpha} \ .$$

Note that *(ii)* holds for $k = 1$, so assume now that $k \geq 2$. To prove *(ii)* for $k \geq 2$, substitute $t = -\frac{\ln(k+1)}{k}$ into the gradient inequality above to obtain $-\frac{\ln(k+1)}{k} \geq 1 - (k+1)^{\frac{1}{k}}$ which can be rearranged to:

$$(k+1)^{\frac{1}{k}} \geq 1 + \frac{\ln(k+1)}{k} \geq 1 + \frac{\ln(e)}{k} = 1 + \frac{1}{k} = \frac{k+1}{k} \ . \tag{58}$$

Inverting (58) yields:

$$(k+1)^{-\frac{1}{k}} \leq \frac{k}{k+1} = 1 - \frac{1}{k+1} \ . \tag{59}$$

Finally, rearranging (59) and multiplying by $k+1$ yields *(ii)*. □

Proposition A.5. *For any integers ℓ, k with $2 \leq \ell \leq k$, the following inequalities hold:*

$$\ln\left(\frac{k+1}{\ell}\right) \leq \sum_{i=\ell}^{k} \frac{1}{i} \leq \ln\left(\frac{k}{\ell-1}\right) \ , \tag{60}$$

and

$$\frac{k-\ell+1}{(k+1)\ell} \leq \sum_{i=\ell}^{k} \frac{1}{i^2} \leq \frac{k-\ell+1}{k(\ell-1)} \ , \tag{61}$$

Proof. (60) and (61) are specific instances of the following more general fact: if $f(\cdot) : [1, \infty) \to \mathbb{R}$ is a monotonically decreasing continuous function, then

$$\int_{\ell}^{k+1} f(t)dt \leq \sum_{i=\ell}^{k} f(i) \leq \int_{\ell-1}^{k} f(t)dt \ . \tag{62}$$

It is easy to verify that the integral expressions in (62) match the bounds in (60) and (61) for the specific choices of $f(t) = \frac{1}{t}$ and $f(t) = \frac{1}{t^2}$, respectively. □