A FRAMEWORK FOR APPLYING SUBGRADIENT METHODS TO CONIC OPTIMIZATION PROBLEMS

JAMES RENEGAR

Abstract. A framework is presented whereby a general convex conic optimization problem is transformed into an equivalent convex optimization problem whose only constraints are linear equations and whose objective function is Lipschitz continuous, with Lipschitz constant no greater than 1. Virtually any subgradient method can be applied to solving the equivalent problem. The computational complexity ramifications are explored for a well-known “optimal” subgradient method.

1. Introduction

Given a conic optimization problem for which a strictly feasible point is known, we provide a transformation to an equivalent convex optimization problem which is of the same dimension, has only linear equations as constraints (one more linear equation than the original problem), and has Lipschitz-continuous objective function defined on the whole space, for which the Lipschitz constant is at most 1 for specified norms. Virtually any subgradient method can be applied to solving the equivalent problem, the cost per iteration dominated by computation of a subgradient and its orthogonal projection onto a subspace (the same subspace at every iteration, a situation for which preprocessing is effective).

We develop representative complexity results, relying on a well-known “optimal” subgradient method.

Perhaps most surprising is that the transformation to an equivalent problem is simple and so is the basic theory, and yet the approach has been overlooked until now, a blind spot.

The following section presents the transformation and basic theory. Implications regarding computational complexity are developed in Sections 3 and 4. An illustrative example is presented in Section 5, clarifying the results in earlier sections, and highlighting some key differences with traditional literature on subgradient methods.

Whereas here we focus on applying subgradient methods directly to the equivalent problem, in a companion paper the focus is on (accelerated) gradient methods. There, the conic optimization problem is assumed to have particular structure sufficient for making a tractable “smoothing” to the objective function of the equivalent problem. The general theory presented in the following section provides the foundation for that paper as well as for this one.

Together, the two papers refine and significantly extend our arXiv posting [3].

Date: March 9, 2015.

Special thanks to Yurii Nesterov for encouragement at an early, critical stage of the research. And thanks to Rob Freund, whose correspondence sparked the realization of how to best partition the results into two papers.
2. Basic Theory

The theory given here is elementary and yet surprising, the “blind spot” referred to in the introduction.

Let $\mathcal{E}$ be a finite-dimensional real Euclidean space. Let $K \subset \mathcal{E}$ be a proper, closed, convex cone with non-empty interior.

Fix a vector $e$ in the interior of $K$. We refer to $e$ as the “distinguished direction.” For each $x \in \mathcal{E}$, let

$$\lambda_{\min}(x) := \inf \{ \lambda : x - \lambda e \notin K \},$$

that is, the scalar $\lambda$ for which $x - \lambda e$ lies in the boundary of $K$. (Existence and uniqueness of $\lambda_{\min}(x)$ follows from $e \in \text{int}(K) \neq \mathcal{E}$ and convexity of $K$.)

If, for example, $\mathcal{E} = S^n_+$ (cone of positive definite matrices), $K = S^n_+$ (cone of positive semidefinite matrices), and $e = I$ (the identity), then $\lambda_{\min}(X)$ is the minimum eigenvalue of $X$.

On the other hand, if $K = \mathbb{R}_+^n$ (non-negative orthant) and $e$ is a vector with all positive coordinates, then $\lambda_{\min}(x) = \min_j x_j/e_j$ for $x \in \mathbb{R}_+^n$. Clearly, the value of $\lambda_{\min}(x)$ depends on the distinguished direction $e$ (a fact the reader should keep in mind since the notation does not reflect the dependence).

Obviously, $K = \{ x : \lambda_{\min}(x) \geq 0 \}$ and $\text{int}(K) = \{ x : \lambda_{\min}(x) > 0 \}$. Also,

$$\lambda_{\min}(sx + te) = s \lambda_{\min}(x) + t \quad \text{for all } x \in \mathcal{E} \text{ and scalars } s \geq 0, t. \tag{2.1}$$

Let

$$\bar{B} := \{ v \in \mathcal{E} : e + v, e - v \in K \},$$

a closed, centrally-symmetric, convex set with nonempty interior. Define a seminorm\(^1\) on $\mathcal{E}$ according to

$$\|u\|_\infty := \min \{ t : u = tv \text{ for some } v \in \bar{B} \}. \tag{2.2}$$

Let $\bar{B}_\infty(x, r)$ denote the closed ball centered at $x$ and of radius $r$. Clearly, $\bar{B}_\infty(0, 1) = \bar{B}$, and $\bar{B}_\infty(e, 1)$ is the largest subset of $K$ that has symmetry point $e$, i.e., for each $v$, either both points $e + v$ and $e - v$ are in the set, or neither point is in the set.

It is straightforward to show $\| \|_\infty$ is a norm if and only if $K$ is pointed (i.e., contains no subspace other than $\{0\}$).

**Proposition 2.1.** The function $x \mapsto \lambda_{\min}(x)$ is concave and Lipschitz continuous:

$$|\lambda_{\min}(x) - \lambda_{\min}(y)| \leq \| x - y \|_\infty \quad \text{for all } x, y \in \mathcal{E}. \tag{2.3}$$

**Proof:** Concavity follows easily from the convexity of $K$, so we focus on establishing Lipschitz continuity.

Let $x, y \in \mathcal{E}$. According to (2.1), the difference $\lambda_{\min}(x + te) - \lambda_{\min}(y + te)$ is independent of $t$, and of course so is the quantity $\|(x + te) - (y + te)\|_\infty$. Consequently, in proving the Lipschitz continuity, we may assume $x$ lies in the boundary of $K$, that is, we may assume $\lambda_{\min}(x) = 0$. The goal, then, is to prove

$$|\lambda_{\min}(x + v)| \leq \|v\|_\infty \quad \text{for all } v \in \mathcal{E}. \tag{2.4}$$

\(^1\)Recall that a seminorm $\| \|$ satisfies $\|tv\| = |t| \|v\|$ and $\|u + v\| \leq \|u\| + \|v\|$, but unlike a norm, is allowed to satisfy $\|v\| = 0$ for $v \neq 0$. 

We consider two cases. First assume \( x + v \) does not lie in the interior of \( K \), that is, assume \( \lambda_{\min}(x + v) \leq 0 \). Then, to establish (2.2), it suffices to show 
\[
\lambda_{\min}(x + v) \geq -\|v\|_{\infty},
\]
that is, to show 
\[
x + v + \|v\|_{\infty}e \in K. \tag{2.3}
\]
However,
\[
v + \|v\|_{\infty}e \in \bar{B}_{\infty}(\|v\|_{\infty}e, \|v\|_{\infty}) \subseteq K, \tag{2.4}
\]
the set containment due to \( K \) being a cone and, by construction, \( \bar{B}_{\infty}(e, 1) \subseteq K \).

Since \( x \in K \) (indeed, \( x \) is in the boundary of \( K \)), (2.3) follows.

Now consider the case \( x + v \in K \), i.e., \( \lambda_{\min}(x + v) \geq 0 \). To establish (2.2), it suffices to show 
\[
\lambda_{\min}(x + v) \leq \|v\|_{\infty},
\]
that is, to show 
\[
x + v - \|v\|_{\infty}e \notin \text{int}(K). \tag{2.5}
\]
Assume otherwise, that is, assume 
\[
x = w + \|v\|_{\infty}e - v \quad \text{for some } w \in \text{int}(K).
\]
Since \( \|v\|_{\infty}e - v \in K \) (by the set containment on the right of (2.4)), it then follows that
\( x \in \text{int}(K) \), a contradiction to \( x \) lying in the boundary of \( K \).

Let the Euclidean space \( \mathcal{E} \) have inner product written \( u \cdot v \). Let \( \text{Affine} \subseteq \mathcal{E} \) be an affine space, i.e., the translate of a subspace. For fixed \( c \in \mathcal{E} \), consider the conic program
\[
\begin{align*}
\inf & \quad c \cdot x \\
\text{s.t.} & \quad x \in \text{Affine} \\
& \quad x \in K
\end{align*}
\]
\( \text{CP} \)

Let \( \text{val}^* \) denote the optimal value.

Assume \( c \) is not orthogonal to the subspace of which \( \text{Affine} \) is a translate, since otherwise all feasible points are optimal. This assumption implies that all optimal solutions for \( \text{CP} \) lie in the boundary of \( K \).

Fix a strictly feasible point \( e \), i.e., \( e \in \text{Affine} \cap \text{int}(K) \). The feasible point \( e \) serves as the distinguished direction.

For scalars \( \text{val} \in \mathbb{R} \), we introduce the affine space
\[
\text{Affine}_{\text{val}} := \{ x \in \text{Affine} : c \cdot x = \text{val} \}.
\]

Presently we show that for any choice of \( \text{val} \) satisfying \( \text{val} < c \cdot e \), \( \text{CP} \) can be easily transformed into an equivalent optimization problem in which the only constraint is \( x \in \text{Affine}_{\text{val}} \). We need a simple observation.

**Lemma 2.2.** Assume \( \text{CP} \) has bounded optimal value.

If \( x \in \text{Affine} \) satisfies \( c \cdot x < c \cdot e \), then \( \lambda_{\min}(x) < 1 \).

**Proof:** If \( \lambda_{\min}(x) \geq 1 \), then \( e + t(x - e) \) is feasible for all \( t \geq 0 \) (using (2.1)). As the function \( t \mapsto c \cdot (e + t(x - e)) \) is strictly decreasing (because \( c \cdot x < c \cdot e \)), this implies \( \text{CP} \) has unbounded optimal value, contrary to assumption.

For \( x \in \mathcal{E} \) satisfying \( \lambda_{\min}(x) < 1 \), let \( z(x) \) denote the point where the half-line beginning at \( e \) in direction \( x - e \) intersects the boundary of \( K \):
\[
z(x) := e + \frac{1}{1-\lambda_{\min}(x)}(x - e)
\]
(to verify correctness of the expression, observe (2.1) implies \( \lambda_{\min}(z(x)) = 0 \)).
We refer to $z(x)$ as “the projection (from $e$) of $x$ to the boundary (of the cone $K$).”

The centrality of the following result to the development makes the result be a theorem even if the proof is straightforward.

**Theorem 2.3.** Let $\text{val}$ be any value satisfying $\text{val} < c \cdot e$. If $x^*$ solves

$$
\sup_{x} \lambda_{\text{min}}(x)
\text{s.t. } x \in \text{Affine}_{\text{val}} ,
$$

then $z(x^*)$ is optimal for CP. Conversely, if $z^*$ is optimal for CP, then $x^* := e + \frac{c - \text{val}}{c - \text{val}^*}(z^* - e)$ is optimal for (2.5), and $z^* = z(x^*)$.

**Proof:** Fix a value satisfying $\text{val} < c \cdot e$. It is easily proven from the convexity of $K$ that $x \mapsto z(x)$ gives a one-to-one map from $\text{Affine}_{\text{val}}$ onto

$$
\{z \in \text{Affine} \cap \text{bdy}(K) : c \cdot z < c \cdot e\},
$$

where $\text{bdy}(K)$ denotes the boundary of $K$.

For $x \in \text{Affine}_{\text{val}}$, the CP objective value of $z(x)$ is

$$
c \cdot z(x) = c \cdot \left( e + \frac{1}{1 - \lambda_{\text{min}}(x)}(x - e) \right)
= c \cdot e + \frac{1}{1 - \lambda_{\text{min}}(x)}(\text{val} - c \cdot e),
$$

a strictly-decreasing function of $\lambda_{\text{min}}(x)$. Since the map $x \mapsto z(x)$ is a bijection between $\text{Affine}_{\text{val}}$ and the set (2.6), the theorem readily follows. □

CP has been transformed into an equivalent linearly-constrained maximization problem with concave, Lipschitz-continuous objective function. Virtually any subgradient method – rather, supgradient method – can be applied to this problem, the main cost per iteration being in computing a supgradient and projecting it onto the subspace $L$ of which the affine space $\text{Affine}_{\text{val}}$ is a translate.

For illustration, we digress to interpret the implications of the development thus far for the linear program

$$
\min_{x \in \mathbb{R}^n} \begin{bmatrix} c^T x \\ \text{s.t. } Ax = b \\ x \geq 0 \end{bmatrix} \quad \text{LP} \quad \text{(2.8)}
$$

assuming $e = 1$ (the vector of all ones), in which case $\lambda_{\text{min}}(x) = \min_{j} x_j$, and $\|v\|_{\infty}$ is the $\ell_{\infty}$ norm, i.e., $\|v\|_{\infty} = \max_{j} |v_j|$. Let the number of rows of $A$ be $m \geq 1$.

For any scalar $\text{val} < c^T 1$, Theorem 2.3 asserts that LP is equivalent to

$$
\max_{x} \min_{j} x_j
\text{s.t. } Ax = b
\quad c^T x = \text{val},
$$

in that when $x$ is feasible for (2.9), $x$ is optimal if and only if the projection $z(x) = 1 + \frac{1}{1 - \lambda_{\text{min}},x_j}(x - 1)$ is optimal for LP. The setup is shown schematically in the following figure.
Proposition 2.1 asserts that, as is obviously true, \( x \mapsto \min_j x_j \) is \( \ell_\infty \)-Lipschitz continuous with constant 1. Consequently, the function also is \( \| \cdot \|_2 \)-Lipschitz continuous with constant 1, as is relevant if supgradient methods rely on the standard inner product in computing supgradients and their orthogonal projections onto the subspace \( L \) of which Affine\(_{\text{val}} \) is a translate, i.e., \( L = \{ v : Av = 0 \text{ and } c^T v = 0 \} \).

With respect to the standard inner product, the supgradients of \( x \mapsto \min_j x_j \) at \( x \) are the convex combinations of the standard basis vectors \( e(k) \) for which \( x_k = \min_j x_j \). Consequently, the projected supgradients at \( x \) are the convex combinations of the vectors \( \bar{P}_k \) for which \( x_k = \min_j x_j \), where \( \bar{P}_k \) is the \( k \)th column of the matrix projecting \( \mathbb{R}^n \) onto the nullspace of \( \bar{A} = [\bar{A} \ c^T] \), that is \( \bar{P} := I - \bar{A}^T (\bar{A} \bar{A}^T)^{-1} \bar{A} \).

In particular, if for a supgradient method the current iterate is \( x \), then the chosen projected supgradient can simply be any of the columns \( \bar{P}_k \) for which \( x_k = \min_j x_j \). Choosing the projected supgradient in this way gives the supgradient method a combinatorial feel. If, additionally, the supgradient method does exact line searches, then the algorithm possesses distinct combinatorial structure. (In this regard, it is noteworthy that the work required for an exact line search is only \( O(n \log n) \), dominated by the cost of sorting.)

If \( m \ll n \), then \( \bar{P} \) is not computed in its entirety, but instead the matrix \( \bar{M} = (\bar{A} \bar{A}^T)^{-1} \) if formed as a preprocessing step, at cost \( O(m^2n) \). Then, for any iterate \( x \) and an index \( k \) satisfying \( x_k = \min_j x_j \), the projected supgradient \( \bar{P}_k \) is computed according to

\[
 u = \bar{M} \bar{A}_k \quad \rightarrow \quad v = \bar{A}^T u \quad \rightarrow \quad \bar{P}_k = e(k) - v ,
\]

for a cost of \( O(m^2 + \#\text{non_zero_entries_in_A} + n \log n) \) per iteration, where \( O(n \log n) \) is the cost of finding a smallest coordinate of \( x \).

Before returning to the general theory, we note that if the choices are \( \mathcal{E} = \mathbb{S}^n \), \( K = \mathbb{S}^+_n \) and \( e = I \) (and thus \( \lambda_{\text{min}}(X) \) is the minimum eigenvalue of \( X \)), then with respect to the trace inner product, the supgradients at \( X \) for the function \( X \mapsto \lambda_{\text{min}}(X) \) are the convex combinations of the matrices \( vv^T \), where \( X v = \lambda_{\text{min}}(X) v \) and \( \|v\|_2 = 1 \). (Supgradients arising from second-order cone programs, and some other conic programs, are discussed in the companion paper.)

Assume, henceforth, that CP has at least one optimal solution, and that val is a fixed scalar satisfying \( \text{val} < c \cdot e \). Then the equivalent problem (2.5) has at least one optimal solution. Let \( x^*_{\text{val}} \) denote any of the optimal solutions for the equivalent
problem, and recall \( \text{val}^* \) denotes the optimal value of CP. A useful characterization of the optimal value for the equivalent problem is easily provided.

**Lemma 2.4.**

\[
\lambda_{\min}(x_{\text{val}}^*) = \frac{\text{val} - \text{val}^*}{c \cdot e - \text{val}^*}
\]

**Proof:** By Theorem 2.3, \( z(x_{\text{val}}^*) \) is optimal for CP – in particular, \( c \cdot z(x_{\text{val}}^*) = \text{val}^* \). Thus, according to (2.7),

\[
\text{val}^* = c \cdot e + \frac{1}{1 - \lambda_{\min}(x_{\text{val}}^*)} (\text{val} - c \cdot e).
\]

Rearrangement completes the proof. \( \square \)

We focus on the goal of computing a point \( z \) which is feasible for CP and has better objective value than \( e \) in that

\[
\frac{c \cdot z - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon,
\]

where \( 0 < \epsilon < 1 \) is user-chosen. Thus, for the problem of main interest, CP, the focus is on relative improvement in the objective value.

The following proposition provides a useful characterization of the accuracy needed in approximately solving the CP-equivalent problem (2.5) so as to ensure that for the computed point \( x \), the projection \( z = z(x) \) satisfies (2.10).

**Proposition 2.5.** If \( x \in \text{Affine}_{\text{val}} \) and \( 0 < \epsilon < 1 \), then

\[
\frac{c \cdot z(x) - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon
\]

if and only if

\[
\lambda_{\min}(x_{\text{val}}^*) - \lambda_{\min}(x) \leq \frac{\epsilon}{1 - \epsilon} \frac{c \cdot e - \text{val}}{c \cdot e - \text{val}^*}.
\]

**Proof:** Assume \( x \in \text{Affine}_{\text{val}} \). For \( y = x \) and \( y = x_{\text{val}}^* \), we have the equality (2.7), that is,

\[
c \cdot z(y) = c \cdot e + \frac{1}{1 - \lambda_{\min}(y)} (\text{val} - c \cdot e).
\]

Thus,

\[
\frac{c \cdot z(x) - \text{val}^*}{c \cdot e - \text{val}^*} = \frac{c \cdot z(x) - c \cdot z(x_{\text{val}}^*)}{c \cdot e - c \cdot z(x_{\text{val}}^*)} = \frac{1}{1 - \lambda_{\min}(x)} - \frac{1}{1 - \lambda_{\min}(x_{\text{val}}^*)} = \frac{\lambda_{\min}(x_{\text{val}}^*) - \lambda_{\min}(x)}{1 - \lambda_{\min}(x)}.
\]
Hence,
\[
\frac{c \cdot z(x) - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon
\]
\[
\iff
\lambda_{\min}(x_{val}^*) - \lambda_{\min}(x) \leq \epsilon (1 - \lambda_{\min}(x))
\]
\[
(1 - \epsilon)(\lambda_{\min}(x_{val}^*) - \lambda_{\min}(x)) \leq \epsilon(1 - \lambda_{\min}(x_{val}^*))
\]
\[
\iff
\lambda_{\min}(x_{val}^*) - \lambda_{\min}(x) \leq \frac{\epsilon}{1 - \epsilon}(1 - \lambda_{\min}(x_{val}^*)).
\]
Using Lemma 2.4 to substitute for the rightmost occurrence of \(\lambda_{\min}(x_{val}^*)\) completes the proof.

In concluding the section, we remark that the basic theory holds for convex conic optimization problems generally. For example, consider a conic program
\[
\begin{aligned}
\min_{x \in \mathcal{E}} & \quad c \cdot x \\
\text{s.t.} & \quad x \in \text{Affine} \\
& \quad Ax + b \in K'
\end{aligned}
\] (2.11)
Here, \(A\) is a linear operator from \(\mathcal{E}\) to a Euclidean space \(\mathcal{E}'\), \(b \in \mathcal{E}'\) and \(K'\) is a proper, closed, convex cone in \(\mathcal{E}'\) with nonempty interior.

For a problem with multiple conic constraints, simply let \(K'\) be the Cartesian product of the cones.

Obviously, the optimization problem \(\text{CP}\) corresponds to the case that \(A\) is the identity, \(b = 0\) and \(K' = K\). (Thus, on the surface, \(\text{CP}'\) appears to be more general than \(\text{CP}\).)

Fix a feasible point \(e\) for which \(e' := Ae + b \in \text{int}(K')\). Using \(e'\) as the distinguished direction results in a function \(x' \mapsto \lambda'_{\min}(x')\) and a seminorm \(\| \cdot \|_\infty\) on \(\mathcal{E}'\). A seminorm is induced on \(\mathcal{E}\), according to \(v \mapsto \|Av\|_\infty\), for which the closed unit ball centered at \(e\) is the largest subset of \(\{x : Ax + b \in K'\}\) with symmetry point \(e\). The map \(x \mapsto \lambda'_{\min}(Ax + b)\) is Lipschitz continuous:
\[
|\lambda'_{\min}(Ax + b) - \lambda'_{\min}(Ay + b)| \leq \|A(x - y)\|_\infty
\]
for all \(x, y \in \mathcal{E}\).

If \(x \in \text{Affine}\) satisfies \(c \cdot x < c \cdot e\), then \(\lambda'_{\min}(Ax + b) < 1\), and the projection of \(x\) (from \(e\)) to the boundary of the feasible region is given by
\[
z(x) = e + \frac{1}{1 - \lambda_{\min}(Ax + b)} (x - e).
\]
Assuming \(\text{val}\) is a scalar satisfying \(\text{val} < c \cdot e\), the problem
\[
\begin{aligned}
\max & \quad \lambda'_{\min}(Ax + b) \\
\text{s.t.} & \quad x \in \text{Affine}_{\text{val}} \quad (:= \{x \in \text{Affine} : c \cdot x = \text{val}\})
\end{aligned}
\] (2.12)
is equivalent to \(\text{CP}'\) in that when \(x \in \text{Affine}_{\text{val}}\), \(x\) is optimal for (2.12) if and only if \(z(x)\) is optimal for \(\text{CP}'\). Moreover, letting \(x_{val}^*\) denote any optimal solution of (2.12), there holds the relation for all \(x \in \text{Affine}_{\text{val}}\) and \(0 < \epsilon < 1\),
\[
\frac{c \cdot x - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon \iff \lambda'_{\min}(Ax_{val}^* + b) - \lambda'_{\min}(Ax + b) \leq \frac{\epsilon}{1 - \epsilon} \frac{c \cdot e - \text{val}^*}{c \cdot e - \text{val}^*},
\]
where \(\text{val}^*\) is the optimal value of \(\text{CP}'\).
These claims regarding $CP'$ are justified with proofs that are essentially identical to the proofs for $CP$. Alternatively, they can be deduced from the results for $CP$ by introducing a new variable $t$ into $CP'$ and an equation $t = 1$, then replacing $K'$ by $K := \{(x, t) : Ax + tb \in K'\}$, thereby recasting $CP'$ to be of the same form as $CP$. (Only on the surface does $CP'$ appear to be more general than $CP$.)

We focus on $CP$ because notationally its form is least cumbersome. For every result derived in the following sections, an essentially identical result holds for any conic program, even identical in the specific constants.

3. Regarding Supgradient Methods

In this and the next section, we show how the basic theory from Section 2 leads to complexity results regarding the solution of the conic program

$$\min \begin{cases} c \cdot x \\ s.t. \quad x \in \text{Affine} \\ x \in K \end{cases}$$

$CP$

Continue to assume $CP$ has an optimal solution and let $e$ denote a strictly feasible point, the distinguished direction.

Given $\epsilon > 0$ and a value satisfying $\text{val} < c \cdot e$, the approach is to apply supgradient methods to approximately solve

$$\max \lambda_{\min}(x) \quad \text{s.t. Affine}_{\text{val}} ,$$

where by “approximately solve” we mean that $x \in \text{Affine}_{\text{val}}$ is computed for which

$$\lambda_{\min}(x_{\text{val}}^*) - \lambda_{\min}(x) \leq \epsilon'$$

with $\epsilon'$ satisfying

$$\epsilon' \leq \frac{\epsilon}{1 - \epsilon} \frac{c \cdot e - \text{val}}{c \cdot e - \text{val}^*}.$$  

Indeed, according to Proposition 2.5, the projection $z = z(x)$ will then satisfy

$$\frac{c \cdot z - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon .$$  

Let $L$ be the subspace of which the affine space $\text{Affine}_{\text{val}}$ is a translate. When supgradient methods are applied to solving (3.1), $L$ is the subspace onto which supgradients are orthogonally projected.

Supgradients and their orthogonal projections depend on the inner product. We allow the “computational inner product” to differ from the one relied upon in expressing $CP$, the inner product written $\langle u, v \rangle$. We denote the computational inner product by $\langle \cdot , \cdot \rangle$, and let $\| \cdot \|$ be its norm.

---

2 Of course orthogonality in a Euclidean space $E$ depends on the chosen inner product ($\langle u, v \rangle = 0$), and hence so do orthogonal projections. Recall that supgradients also depend on the chosen inner product, in that for a concave function $f : E \to \mathbb{R}$, the supgradients of $f$ at $x$ are the vectors $\nabla f(x) \in E$ satisfying $f(x) + \langle \nabla f(x), v \rangle \geq f(x + v)$ for all $v \in E$. 
It is an instructive exercise to show that the supdifferential (set of supgradients) at $x$ for the function $x \mapsto \lambda_{\text{min}}(x)$ is

$$\{ g : \langle g, e \rangle = 1 \text{ and } \langle g, y - (x - \lambda_{\text{min}}(x) e) \rangle \geq 0 \text{ for all } y \in K \},$$

that is, the supdifferential consists of vectors $g$ such that $\langle g, e \rangle = 1$ and $-g$ is in the normal cone to $K$ at $x - \lambda_{\text{min}}(x) e$. (To begin, note it may be assumed that $\lambda_{\text{min}}(x) = 0$, due to (2.1).)

Supgradient methods rely on an upper bound for the Lipschitz constant with respect to $\|\| \text{ rather than with respect to the seminorm } \|\|_{\infty}$ appearing in Proposition 2.1. However, so long as the computational inner product satisfies $\| v \| \geq \| v \|_{\infty}$ for all $v \in \mathcal{E}$, the Lipschitz constant with respect to $\|\|$ will be at most 1, as follows trivially from Proposition 2.1.

In fact, all that is needed by supgradient methods in solving (3.1) is an upper bound on the Lipschitz constant for the restriction of the function $x \mapsto \lambda_{\text{min}}(x)$ to $\text{Affine}_{\text{val}}$. To have the relevant Lipschitz constant bounded above by 1, it thus suffices for the computational inner product to satisfy

$$\| v \| \geq \| v \|_{\infty} \text{ for all } v \in L.$$  \hfill (3.3)

An equivalent condition, albeit more useful for modeling, is

$$B(e, 1) \cap \text{Affine}_{c \cdot e} \subseteq K,$$  \hfill (3.4)

where $B(x, r)$ denotes the $\|\|$-ball centered at $x$ and of radius $r$. The equivalence of (3.3) and (3.4) is due simply to $B_{\infty}(e, 1)$ being the largest set contained in $K$ for which $e$ is a symmetry point.

The analogous condition in the context of $\text{CP}'$ – the conic program (2.11) – is for the computational inner product to satisfy

$$B(e, 1) \cap \text{Affine}_{c \cdot e} \subseteq \{ x : Ax + b \in K' \}.$$  \hfill (3.5)

Below, we recall a complexity result for a particular supgradient method, interpreted for when the method is applied to solving the $\text{CP}$-equivalent problem (3.1). From the complexity result is deduced a bound on the number of iterations sufficient to obtain $x$ whose projection $z = z(x)$ satisfies (3.2). We observe, however, that in a certain respect, the iteration bound is disappointing. In the next section, the framework is embellished by applying the supgradient method not to (3.1) for only one value $\text{val}$, but for a small and carefully chosen sequence of values, $\text{val} = \text{val}_\ell$. The embellishment results in a computational scheme which possesses the desired improvement.

The particular supgradient method is chosen precisely because its theory is widely known from Nesterov’s book [2]. The results deduced below are meant only to be representative, in that similar results can also be established for other “optimal” supgradient methods when applied in the framework. (Almost surely for practice, however, the best supgradient methods to use in the framework are not among the so-called “optimal” methods.)

For specifying a supgradient method and stating a bound on its complexity, we follow [2, Chapter 2.3.2]:
Supgradient Method (for the CP-equivalent problem)

(0) Inputs:
- Number of iterations: $N$
- Initial iterate: $x_0 \in \text{Affine} \text{satisfying } c \cdot x_0 < c \cdot e$ .
  Let $\text{val} := c \cdot x_0$. (Hence, $x_0 \in \text{Affine}_{\text{val}}$)
- Distance upper bound: $R$, a value for which there exists optimal $x^*_{\text{val}}$ satisfying $\|x_0 - x^*_{\text{val}}\| \leq R$ .
- Initial “best” iterate: $x = x_0$
- Initial counter value: $k = -1$

(1) Update counter: $k + 1 \to k$
(2) Compute a supgradient $\nabla \lambda_{\text{min}}(x_k)$ and orthogonally project it onto the subspace $L$. Denoting the projection by $g_k$, compute
$$x_{k+1} := x_k + \frac{R}{\sqrt{N} \|g_k\|} g_k .$$
(3) If $\lambda_{\text{min}}(x_{k+1}) > \lambda_{\text{min}}(x)$, then make the replacement $x_{k+1} \to x$ .
(4) Check for termination: If $k = N$, then output $x$ and terminate. Else, go to Step 1.

Proposition 3.1. If the computational inner product $(, )$ satisfies condition (3.4), then the output $x$ from Supgradient Method satisfies
$$\lambda_{\text{min}}(x^*_{\text{val}}) - \lambda_{\text{min}}(x) \leq \frac{R}{\sqrt{N}} ,$$
where $\text{val} := c \cdot x_0$ and $x_0$ is the input.

Proof: We know condition (3.4) implies that with respect to $\| \|$, the value 1 serves as an upper bound on the Lipschitz constant. Consequently, the proposition is a simple corollary to [2, Theorem 3.2.2], by choosing the parameter values there to be $h_k = \frac{R}{\sqrt{N}}$ for $k = 0, \ldots, N$. \hfill \square

Later, we require the input $x_0$ for Supgradient Method to be feasible for CP, mainly so that the input $R$ can be chosen as a value with clear relevance to CP, a value we now describe.

The level sets for CP are the sets
$$\text{Level}_{\text{val}} = \text{Affine}_{\text{val}} \cap K ,$$
the largest feasible sets for CP on which the objective function is constant. If $\text{val} < c \cdot e$, then $\text{Level}_{\text{val}} = \emptyset$ .

If some level set is unbounded, then either CP has unbounded optimal value or can be made to have unbounded value with an arbitrarily small perturbation of $c$. Thus, in developing numerical optimization methods, it is natural to focus on the case that level sets for CP are bounded.

For scalars $\text{val} \in \mathbb{R}$, let
$$\text{diam}_{\text{val}} := \max\{\|x - y\| : x, y \in \text{Level}_{\text{val}}\} ,$$
the diameter of $\text{Level}_{\text{val}}$. (The norm in the definition is the norm for the computational inner product.) In the next result, we assume an upper bound on $\text{diam}_{\text{val}}$ is known for a fixed scalar $\text{val} < c \cdot e$. We use $D_{\text{val}}$ to denote the known upper bound.
Similarly, in Section 4 we assume known an upper bound

\[ D \geq \max\{\text{diam}_{\text{val}} : \text{val} \leq c \cdot u\}, \]

where \( u \) is input to our main computational scheme (\( u \) is required to be feasible for CP and satisfy \( c \cdot u < c \cdot e \)). The value \( D \) also is input to the scheme. The output is \( z \) which is feasible for CP and guaranteed to satisfy (3.2), with \( \epsilon \) being user-specified. The iteration bound proven for the scheme increases with \( D \). The computational inner product affects the iteration bound only in that the inner product determines the values \( \text{diam}_{\text{val}}, \) and thus limits how small \( D \) can be chosen.

Although the assumption of knowing an upper bound on diameters is strong, it is consistent with assumptions found throughout the literature on first-order methods when the goal is to compute a feasible point whose objective value is guaranteed to be within \( \epsilon \) of optimal, with \( \epsilon \) being user-specified.

Moreover, even though the assumption of knowing an upper bound on the diameters is strong, still there are many interesting situations in which the assumption is valid, particularly when a problem is specifically modeled in such a way as to make the diameter of level sets be of reasonable magnitude. (If nothing else, one can always introduce a bounding constraint so as to enforce an upper bound on diameters.\(^3\))

Worthy of mention is that for a problem modeled as a second-order cone program with \( m \) second-order constraints and with a known point \( e \) on or near the central path (the path ubiquitous in interior-point method theory), the inner product arising from the Hessian at \( e \) for the logarithmic barrier function satisfies both the required condition (3.4) and \( \text{diam}_{\text{val}} \leq 4m \) for all \( \text{val} \leq c \cdot e \), regardless of the number of variables. Thus, second-order cone programming with few constraints and many variables fits especially nicely into the framework. (Second-order cone programming is highlighted in the companion paper.)

We mention, additionally, that the general example presented in Section 5 provides, among other things, further perspective on the assumption of knowing an upper bound on diameters.

In the following corollary regarding Supgradient Method, the choice of input \( N \) depends on \( \text{val}^* \), the optimal value for CP. Naturally the reader will infer that in addition to knowing an upper bound on the diameter of level sets, our main algorithmic scheme will require knowing a lower bound on \( \text{val}^* \), but this is not the case for the scheme. The corollary serves to motivate the next step in specifying the scheme.

**Corollary 3.2.** Assume the computational inner product satisfies (3.4). Assume \( x_0 \) is feasible for CP and satisfies \( c \cdot x_0 < c \cdot e \). Let \( \text{val} := c \cdot x_0 \) and assume \( D_{\text{val}} \) is a scalar satisfying \( D_{\text{val}} \geq \text{diam}_{\text{val}} \). Let \( 0 < \epsilon < 1 \).

---

\(^3\)For example, in the context of CP’ (i.e., the conic program (2.11)), choose \( r \geq 1 \) and replace \( Ax + b \in K' \) by \( (Ax + b, (x - e, r)) \in K' \times \{(y, t) : \|y\| \leq t\} \), where \( \| \| \) is the norm for the computational inner product chosen for CP’. A point \( x \) is feasible for the modified problem if and only if \( x \) is feasible for CP’ and satisfies \( \|x - e\| \leq r \). Since \( r \geq 1 \), the same inner product can be used as the computational inner product for the modified problem (i.e., the condition (3.5) remains satisfied). Trivially, moreover, \( \text{diam}_{\text{val}} \leq 2r \) for every scalar \( \text{val} \).
If $x_0$ and $R = D_{\text{val}}$ are inputs to Supgradient Method, along with an integer $N$ satisfying

$$N \geq \left( \frac{D_{\text{val}}}{\epsilon} \frac{c \cdot e - \text{val}^*}{c \cdot e - \text{val}} \right)^2,$$

then for the output $x$, the projection $z = z(x)$ satisfies

$$\frac{c \cdot z - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon.$$

**Proof:** Since $x_0$ is feasible for CP, so is $x^\text{val}$ (because $0 \leq \lambda_{\min}(x_0) \leq \lambda_{\min}(x^\text{val})$). Thus, $\|x_0 - x^\text{val}\| \leq \text{diam}_{\text{val}}$, making $R = D_{\text{val}}$ a valid input to Supgradient Method.

For inputs as specified, Proposition 3.1 immediately implies the output $x$ for Supgradient Method satisfies

$$\lambda_{\min}(x^\text{val}) - \lambda_{\min}(x) \leq \epsilon \cdot \frac{c \cdot e - \text{val}}{c \cdot e - \text{val}^*}.$$

Invoking Proposition 2.5 completes the proof. □

It has long been known – Nemirovski and Yudin [1] – that for general, unconstrained, Lipschitz-continuous convex functions $f$, to compute $x$ for which $f(x)$ is within $\epsilon$ of optimal requires $\Omega(1/\epsilon^2)$ iterations in the worst case, assuming the algorithm has access to the function only by making a call to an oracle at each iteration, asking for a subgradient at the current iterate. Examples for establishing the lower bound are simple, such as the example used in proving [2, Thm 3.2.1], an example that can easily be recast as a second-order cone program, a special case of CP. Thus, almost surely, the occurrence of $1/\epsilon^2$ in the bound (3.6) is unavoidable, at least in the “black box” setting, where the oracle is free to provide the least useful of all the supgradients for the current iterate.\(^4\)

In a similar vein, occurrence of $D_{\text{val}}^2$ (rather, $\text{diam}_{\text{val}}^2$) almost surely is unavoidable (again see [2, Thm 3.2.1]).

On the other hand, the manner in which the bound (3.6) depends on $\frac{c \cdot e - \text{val}^*}{c \cdot e - \text{val}}$ is disconcerting.

To understand why the dependence is disconcerting, consider that the most natural choice for the input is $x_0 = e - \frac{1}{\lambda_{\max}(\pi(c^\text{val}))} \pi(c^\text{val})$, where $c^\text{val}$ is the vector that replaces $c$ if the objective of CP is written in terms of the computational inner product (i.e., $c^\text{val} \cdot x = c \cdot x$ for all $x \in \mathcal{E}$), and where $\pi(c^\text{val})$ is the orthogonal projection of $c^\text{val}$ onto the subspace of which the affine space Affine is a translate. This is the choice for $x_0$ obtained by moving from $e$ in direction $-\pi(c^\text{val})$ until the boundary of the feasible region is reached.

\(^4\)We include the phrase “almost surely” because otherwise we should present a proof, as in our setting, the oracle provides supgradients for a function $x \mapsto \lambda_{\min}(x)$ rather than subgradients for the function used in establishing [2, Thm 3.2.1]. In other words, it does not immediately follow from [2, Thm 3.2.1] that because the example used in the proof can easily be recast as an equivalent second-order cone program, the occurrence of $1/\epsilon^2$ in (3.6) is unavoidable. Instead, one proceeds by applying the reasoning used in the proof directly to the equivalent problem with objective function $x \mapsto \lambda_{\min}(x)$. Including such a proof would be too much of a digression. Thus we content ourselves with the phrase “almost surely.”
However, for the linear program (2.8), even if \( 1 \) is on the central path and \( \langle \cdot, \cdot \rangle \) is chosen to be the standard inner product (in which case \( c^\circ = c \), and \( -\pi(c) \) is tangent to the central path), it can happen that the value \( c \cdot e - \text{val}^* \) is of magnitude \( \sqrt{n} \) (for \( \text{val} = c \cdot x_0 \) where \( x_0 = 1 - \frac{1}{\lambda_{\max}(\pi(x))} \pi(c) \)), while at the same time, \( \text{diam}_{\text{val}} \leq 4 \) for all scalars \( \text{val} \leq c \cdot 1 \) (in particular, independent of \( n \)). Thus, even for linear programs modeled carefully so that \( 1 \) is on the central path and the upper bound \( D_{\text{val}} \) can be chosen small, the iteration bound (3.6), for fixed \( \epsilon \), can grow like \( n \).

This is disconcerting.

Moreover, we want an algorithm for which \( \text{val}^* \) does not explicitly figure into choosing the inputs. We already assume an upper bound is known for diameters of particular level sets. We want to avoid also assuming a lower bound on \( \text{val}^* \) is known.

4. A Supgradient Scheme

The concerns expressed above raise a question:

Is it possible to efficiently move from an initial CP feasible point \( u \) satisfying \( c \cdot u < c \cdot e \), to a feasible point \( y \) for which, say, \( \frac{c \cdot e - \text{val}^*}{c \cdot e - c \cdot y} \leq 3 \)? (As then, choosing \( x_0 = y \) for input to Supgradient Method, the ratio \( \frac{c \cdot e - \text{val}^*}{c \cdot e - \text{val}} \) is harmless.)

We begin the section by providing an affirmative answer, but first let us again display the pertinent optimization problem:

\[
\max \quad \lambda_{\min}(x) \\
\text{s.t.} \quad x \in \text{Affine}_{\text{val}}.
\]  

(4.1)

Recall that \( x_{\text{val}}^* \) denotes any optimal solution of (4.1), an optimization problem which is equivalent to CP (assuming \( \text{val} < c \cdot e \)).

Recall

\[
\text{diam}_{\text{val}} := \max \{ \| x - y \| : x, y \in \text{Level}_{\text{val}} \}.
\]

Consider the following computational procedure:

- **Supgradient SubScheme**
  
  (0) Initiation:
  - Inputs:
    - A point \( u \) that is feasible for CP and satisfies \( c \cdot u < c \cdot e \).
    - An upper bound \( D \geq \max \{ \text{diam}_{\text{val}} : \text{val} \leq c \cdot u \} \).
  - Let \( u_0 = u \) and \( \text{val}_0 = c \cdot u \).
  - Let \( \ell = -1 \).
  
  (1) Outer Iteration Counter Step: \( \ell + 1 \to \ell \)
  
  (2) Inner Iterations:
    - Apply Supgradient Method to solving (4.1) with \( \text{val} = \text{val}_\ell \), using inputs \( x_0 = u_\ell \), \( R = D \) and \( N = \lceil 9D^2 \rceil \).
    - Rename the output \( x \) as \( v_\ell \).
  
  (3) Check for Termination:
    - If \( \lambda_{\min}(v_\ell) \leq 1/3 \), then output \( y = u_\ell \) and terminate.
    - Else, compute the projection
      \[
      u_{\ell+1} := z(v_\ell) \quad \text{let} \quad \text{val}_{\ell+1} := c \cdot u_{\ell+1},
      \]
      and go to Step 1.
Proposition 4.1. Assume the computational inner product satisfies (3.4).

Supgradient SubScheme outputs $y$ that is feasible for CP and satisfies
\[
\frac{c \cdot e - \text{val}^*}{c \cdot e - c \cdot y} \leq 3. \tag{4.2}
\]
The total number of outer iterations does not exceed
\[
\log_{3/2} \left( \frac{c \cdot e - \text{val}^*}{c \cdot e - c \cdot u} \right),
\]
where $u$ is the input to Supgradient SubScheme.

Proof: It is easily verified that all of the points $u_\ell$ and $v_\ell$ computed by Supgradient SubScheme are contained in the affine space Affine and have objective values less than or equal to $c \cdot u$. Moreover, $u_\ell$ is clearly feasible for CP, lying in the boundary of the feasible region.

Fix $\ell$ to be any value attained by the counter. We now examine the effects of Steps 2 and 3.

Corollary 3.2 with $N = \lceil 9D^2 \rceil$ shows that in Step 2, the output $x$ from Supgradient Method satisfies
\[
\lambda_{\min}(x^*_\text{val}_\ell) - \lambda_{\min}(x) \leq 1/3,
\]
that is,
\[
\lambda_{\min}(x^*_\text{val}_\ell) - \lambda_{\min}(v_\ell) \leq 1/3. \tag{4.3}
\]

Observe
\[
\frac{c \cdot e - \text{val}_\ell}{c \cdot e - \text{val}^*} = 1 - \lambda_{\min}(x^*_\text{val}_\ell) \quad \text{(by Lemma 2.4)}
\]
\[
\geq \frac{2}{3} - \lambda_{\min}(v_\ell) \quad \text{(by (4.3))}
\]
Hence, if the method terminates in Step 3 – that is, if $\lambda_{\min}(v_\ell) \leq 1/3$ – then the output $y = u_\ell$ satisfies
\[
\frac{c \cdot e - c \cdot y}{c \cdot e - \text{val}^*} \geq \frac{1}{3}.
\]

We have now verified that if Supgradient SubScheme terminates, then the output $y$ is indeed feasible for CP and satisfies the desired inequality (4.2).

On the other hand, if the method does not terminate in Step 3, it computes the point $u_{\ell+1}$ and its objective value, $\text{val}_{\ell+1}$. Here, observe
\[
\text{val}_{\ell+1} = c \cdot e + \frac{1}{1 - \lambda_{\min}(v_\ell)} (\text{val}_\ell - c \cdot e)
\leq c \cdot e - \frac{2}{3} (c \cdot e - \text{val}_\ell),
\]
because $\text{val}_\ell < c \cdot e$ and $\lambda_{\min}(v_\ell) \geq 1/3$ (due to no termination). Hence,
\[
\frac{c \cdot e - \text{val}_{\ell+1}}{c \cdot e - \text{val}^*} \geq \frac{3}{2} \frac{c \cdot e - \text{val}_\ell}{c \cdot e - \text{val}^*}.
\]

Since all values $\text{val}_\ell$ computed by the algorithm satisfy $\text{val}_\ell \geq \text{val}^*$ (as $u_\ell$ is feasible for CP), it immediately follows that
\[
\log_{3/2} \left( \frac{c \cdot e - \text{val}^*}{c \cdot e - \text{val}_0} \right)
\]
is an upper bound on the number of outer iterations. □
Specifying our overall computational scheme relying on the supgradient method, and analyzing the scheme’s complexity, both are now easily accomplished:

- **Supgradient Scheme**

  (0) Inputs:
  
  - A value $0 < \epsilon < 1$.
  - A point $u$ which both is feasible for CP and satisfies $c \cdot u < c \cdot e$.
    
    * For example, the point $u = e - \frac{1}{\max\{\pi(c^*)\}} \pi(c^*)$.
  - An upper bound $D \geq \max\{\text{diam}_{\text{val}} : \text{val} \leq c \cdot u\}$.

  (1) Apply Supgradient SubScheme with inputs $u$ and $D$.
  
  Let $y$ denote the output.

  (2) Apply Supgradient Method with inputs $x_0 = y$, $R = D$ and
  
  $$N = \left\lceil \frac{3D}{\epsilon^2} \right\rceil .$$

  Let $x$ denote the output.

  (3) Compute and output the projection $z = z(x)$, then terminate.

Following is our main theorem for Supgradient Scheme applied to solving CP.

**Theorem 4.2.** Assume the computational inner product satisfies (3.4). Supgradient Scheme outputs $z$ which is feasible for CP and satisfies

$$\frac{c \cdot z - \text{val}^*}{c \cdot e - \text{val}^*} \leq \epsilon. \tag{4.4}$$

The total number of iterations of Supgradient Method is bounded above by

$$(9D^2 + 1) \cdot \left( \frac{1}{\epsilon^2} + \log_{3/2} \left( \frac{c \cdot e - \text{val}^*}{c \cdot e - c \cdot u} \right) \right),$$

where $\epsilon$, $u$ and $D$ are the inputs to Supgradient Scheme.

**Proof:** Proposition 4.1 shows the output $y$ from Step 1 is feasible for CP and satisfies

$$\frac{c \cdot e - \text{val}^*}{c \cdot e - c \cdot u} \leq 3.$$

Thus, by Corollary 3.2, when $x_0 = y$ is input into Supgradient Method, along with $R = D$ and $N = \lceil (3D/\epsilon)^2 \rceil$, the projection $z = z(x)$ of the output $x$ satisfies (4.4), establishing correctness of Supgradient Scheme.

The bound for the total number of iterations of Supgradient Method is immediate from the outer iteration bound of Proposition 4.1, and the choices for the number of iterations in Step 2 of Supgradient SubScheme and in Step 2 of Supgradient Scheme.

$\square$

Theorem 4.2 accomplishes our goals of establishing an iteration bound in which the disconcerting factor $\frac{c \cdot e - \text{val}^*}{c \cdot e - c \cdot u}$ appearing in Corollary 3.2 has been made harmless, and doing so with an algorithmic strategy that does not require input dependent on knowing a lower bound for $\text{val}^*$. Moreover, we also have accomplished the primary objective of devising a strategy which can be readily adapted to accommodate any of a variety of supgradient methods.

We do not mean to suggest that Supgradient Scheme should serve as a prototypical strategy for applying supgradient methods to CP-equivalent problems (4.1) so as to solve CP. The primary purpose of Supgradient Scheme is theoretical, in displaying the existence of an “efficient” strategy (more precisely, in displaying...
the existence of a strategy that in the worst case is as efficient as can be hoped, assuming supgradients are provided by a malicious oracle).

Still, we are inclined to believe that the outline of Supgradient SubScheme, in particular, is important to keep in mind when devising an overall scheme meant for practice. Indeed, the “if and only if” of Proposition 2.5 indicates that for efficiency in practice, it is desirable to start by quickly computing a point with the properties of $y$, the output of Supgradient SubScheme. (The outer-iteration bound provided by Proposition 4.1 serves as an appropriate notion of “quickly,” for practice as well as for theory.)

Stepping back, we feel comfortable in asserting that given the straightforwardness of the underlying theory, the framework seems potentially relevant to the practical solution of commonplace conic optimization problems, such as linear programs, second-order cone programs and semidefinite programs, for each of which the cost of computing supgradients of $x \mapsto \lambda_{\min}(x)$ is modest. Still, whether the potentiality can be translated into actuality is very much open. Success likely hinges critically on developing supgradient methods dovetailed to particular conic settings (unlike our deployment of Supgradient Method in all conic settings).

5. An Illustrative Example

We close with a general example meant to illustrate the flexibility of the preceding development, an example which also serves to highlight a few key differences with much of the literature on subgradient methods.

Let $f : \mathcal{E} \to (-\infty, \infty]$ be an extended-valued and lower-semicontinuous convex function. Consider an optimization problem

$$\min_{x \in \text{Feas}} f(x) \quad \text{s.t.} \quad x \in \text{Feas},$$

where Feas = $\{x \in S : Ax = b\}$ and $S$ is a closed, convex set with nonempty interior. Assume $\bar{x}$ is a point known to lie in the interiors of both $S$ and eff$_{\text{dom}}(f)$, the effective domain of $f$ (i.e., where $f$ is finite). For convenience of exposition, assume $\bar{x}$ is not optimal for (5.1).

Assume $\langle \cdot , \cdot \rangle$ is an inner product on $\mathcal{E}$ whose norm satisfies

$$\{x : ||x - \bar{x}|| \leq 1 \text{ and } Ax = b\} \subseteq \text{Feas} \cap \text{eff}_{\text{dom}}(f).$$

(5.2)

Let $\hat{f}$ be a scalar satisfying $\hat{f} \geq f(x)$ for all $x$ in the set on the left of (5.2).

Assume $D$ is an upper bound on the diameter of the sublevel set $\{x \in \text{Feas} : f(x) \leq f(\bar{x})\}$. Boundedness of this set and lower semicontinuity of $f$ together imply (5.1) has an optimal solution. Let $f^*$ denote the optimal value.

For later reference, observe that (5.2) and the definition of $D$, together with the convexity of $f$, imply

$$f(\bar{x}) \leq \frac{1}{D+1} f^* + \frac{D}{D+1} \hat{f},$$

that is,

$$\frac{\hat{f} - f(\bar{x})}{\hat{f} - f^*} \leq D + 1.$$  

(5.3)

As $S$ is closed and convex, there exists a closed, convex cone $K_1 \subseteq \mathcal{E} \times \mathbb{R}$ for which $S = \{x : (x,1) \in K_1\}$. 

For an extended-valued function to be lower semicontinuous is equivalent to its epigraph being closed. Since the epigraph for $f$ is convex, there thus exists a closed, convex cone $K_2 \subseteq \mathcal{E} \times \mathbb{R} \times \mathbb{R}$ for which

$$\text{epi}(f) := \{(x, t) : f(x) \leq t\} = \{(x, 1, t) : (x, 1, t) \in K_2\}.$$  

Note

$$t > f(\bar{x}) \Rightarrow (\bar{x}, 1, t) \in \text{int}(K_2), \quad (5.4)$$

a consequence of $\bar{x} \in \text{int}(\text{eff}_\text{dom}(f))$.

Let

$$K := \{(x, s, t) : (x, s) \in K_1 \text{ and } (x, s, t) \in K_2\}.$$  

Clearly, the optimization problem (5.1) is equivalent to

$$\min_{x, s, t} t \quad \text{s.t.} \quad Ax = b \quad s = 1 \quad (x, s, t) \in K, \quad (5.5)$$

and has the same optimal value, $f^*$. The conic program (5.5) is of the same form as CP, the focus of preceding sections.

Observe for all scalars $\text{val}$,

$$\text{Level}_\text{val} = \{(x, 1, \text{val}) : x \in \text{Feas and } f(x) \leq \text{val}\}. \quad (5.6)$$

To apply the development of preceding sections, a distinguished direction $e$ is needed, and a computational inner product should be specified. Additionally, an input $u$ is required for Supgradient Scheme. (The other input, $D$, has already been chosen.)

Let $e = (\bar{x}, 1, \hat{f})$, which lies in the interior of $K$, due to (5.4) and $\bar{x} \in \text{int}(S)$.

Choose the computational inner product on $\mathcal{E} \times \mathbb{R} \times \mathbb{R}$ to be any inner product that assigns to pairs $(x_1, 0, 0), (x_2, 0, 0)$ the value $\langle x_1, x_2 \rangle$. By (5.2) and (5.6), the extended inner product then satisfies the required condition (3.4) (since in the present case, $c \cdot e = \hat{f}$).

Choose the input $u$ to Supgradient Scheme as $u = (\bar{x}, 1, f(\bar{x}))$, which clearly is feasible for (5.5). Note that (5.6) then implies $D \geq \text{diam}_{\text{val}}$ for all $\text{val} \leq f(\bar{x})$, and hence $D$ is indeed a valid choice for input to Supgradient Scheme (as in the present setting, $c \cdot u = f(\bar{x})$).

Theorem 4.2 shows that upon applying Supgradient Scheme, the output $(x, 1, t)$ satisfies $x \in \text{Feas}$ and

$$\frac{f(x) - f^*}{f - f^*} \leq \epsilon \quad (5.7)$$

(using that the output $(x, 1, t)$ is feasible for (5.5), and hence $f(x) \leq t$). Moreover, the number of Supgradient Method iterations is bounded above by

$$(9D^2 + 1) \cdot \left(\frac{1}{\epsilon^2} + \log_{3/2}(D + 1)\right) \quad (5.8)$$

(where in the logarithm we rely on (5.3)).

Deserving of emphasis is that the only projections made are onto the subspace $\{(x, s, t) : Ax = 0 \text{ and } s = 0\}$ (the same subspace at every iteration, a situation for which preprocessing is effective). This is quite unlike customary subgradient
approaches where, for example, if during an iteration, computation results in a point lying outside the feasible region, the point is replaced by its projection onto the region. Of course such projections are readily computable only if the feasible region is of an especially nice form, such as a unit cube, a unit simplex or an affine space.

Another difference deserving of emphasis is that the iteration bound (5.8) is independent of a Lipschitz constant for \( f \). In fact, no Lipschitz constant is implied by the assumptions, as is seen by considering the univariate cases in which \( \text{Feas} = \mathbb{R}, \ A = b = 0, \) and \( f \) is allowed to be any lower-semicontinuous convex function with

\[
[0, 2] \subseteq \text{eff,dom}(f) \subseteq [0, \infty),
\]

and which is strictly increasing at 0. Letting \( \bar{x} = 1 \), the assumptions are fulfilled by choosing the computational inner product to be \( \langle u, v \rangle = uv \), and by choosing \( D = 1 \) and \( \bar{f} = f(2) \). The iteration bound (5.8) is then \( 10(\frac{1}{\epsilon^2} + \log_{3/2}(2)) \), whereas the error bound (5.7) is \( \frac{f(x) - f(0)}{f(2) - f(0)} \leq \epsilon \). Clearly, even for the restriction of \( f \) to \([0, 2]\), nothing is implied about the Lipschitz constant other than trivial bounds such as \( L \geq \frac{1}{2}(f(2) - f(0)) \).

For the approach presented herein, Lipschitz continuity matters only with regards to the function \((x, s, t) \mapsto \lambda_{\text{min}}(x, s, t)\) restricted to \( \text{Affine}_{\text{val}} \), which is guaranteed to have Lipschitz constant 1 (as the (extended) computational inner product has been chosen expressly to satisfy condition (3.4)). On the other hand, the error (5.7) is measured relatively, whereas in traditional subgradient-method literature relying on a Lipschitz constant for the objective function \( f \), error is specified absolutely.

References

[1] A.S. Nemirovski and D.B. Yudin. Problem Complexity and Method Efficiency in Optimization. John Wiley & Sons, 1983.

[2] Yu. Nesterov. Introductory Lectures on Convex Optimization: A Basic Course. Springer, 2004.

[3] J. Renegar. Efficient first-order methods for linear programming and semidefinite programming. arXiv preprint arXiv:1409.5882, 2014.

School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, U.S.