IMPROVED ALGORITHMS FOR CONVEX MINIMIZATION IN RELATIVE SCALE*

PETER RICHTÁRIK†

Abstract. In this paper we propose two modifications to Nesterov's algorithms for minimizing convex functions in relative scale. The first is based on a bisection technique and leads to improved theoretical iteration complexity and the second is a heuristic for avoiding restarting behavior. The fastest of our algorithms produces a solution within relative error $O(1/k)$ of the optimum, with k being the iteration counter.

Key words. convex optimization, relative scale, sublinearity, Nesterov's smoothing technique, Löwner-John ellipsoids

AMS subject classifications. 62K05, 65K05, 68Q25, 90C06, 90C25, 90C47, 90C60

1. Introduction.

The theory of modern convex optimization almost uniformly assumes boundedness of the feasible set. This assumption is usually artificially enforced even for naturally unconstrained problems via the so-called "big M" method. A clear advantage of dealing with bounded sets is the availability of a *scale* in which one can measure the absolute accuracy of a solution. However, there always seems to be the issue of keeping a balance between the size of the artificially imposed bounds (large feasible sets tend to slow algorithms down) and the possibility of exclusion of minimizers from the feasible sets in so doing. Since there is no natural absolute scale for measuring the solutions of an unconstrained problem, it seems to be reasonable to be looking for solutions that are approximately optimal in *relative scale*. Although results of this type are rare in the convex optimization literature, some work has recently been done in this area [14, 15, 17, 18]. This contrasts with the enormous literature on combinatorial optimization where approximation algorithms are studied extensively.

In particular, Nesterov [14] showed that the above obstacles can be overcome for the problem class of minimizing a convex *homogeneous* function over an affine subspace. The essence of his approach involves the computation of an *ellipsoidal rounding* of the subdifferential of the objective function (at the origin) by utilizing the knowledge about the structure of the problem. This family of problems encompasses essentially all unconstrained convex minimization problems via a dimension-lifting procedure. However, certain assumptions about the ellipsoidal rounding effectively limit the class of problems that can be treated.

Contribution. In this paper we improve the algorithms of Nesterov [13,14] for solving unconstrained convex minimization problems within a prescribed error δ in relative scale. We propose two modifications of the original method: the first is based on a bisection technique and leads to improved theoretical iteration complexity. The second is a heuristic for avoiding certain restarting behavior of the method. The fastest of

*Version from December 19, 2008. The results of this paper were obtained in the years 2005 and 2006 while the author was a research assistant at Cornell University, working under the guidance of Mike Todd. They form a part of the PhD thesis [16, Chapter 2] of the author, and were not previously published. This research was partially supported by NSF through grants DMS-0209457 and DMS-0513337 and by ONR through grant N00014-02-0057.

†Center for Operations Research and Econometrics (CORE) and Department of Mathematical Engineering (INMA), Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (peter.richtarik@uclouvain.be).

our algorithms produces a solution within relative error $O(1/k)$ of the optimum, with k being the iteration counter. The bisection idea was independently used by Chudak and Eleutério [4] for obtaining the same theoretical improvement in complexity in the context of several combinatorial problems.

Contents. The paper is organized as follows. In Section 2 we formulate the central sublinear minimization problem and briefly describe a dimension-lifting procedure for converting an unconstrained minimization instance into a linearly constrained sublinear minimization instance. Section 3 is devoted to defining basic notions and deriving key consequences of the necessary pre-processing stage of our algorithms: the computation of a pair of Löwner-John ellipsoids of a certain set. The next two parts are devoted to the description and analysis of algorithms. In Section 4 we describe methods based on a simple subgradient subroutine. We first summarize Nesterov's results and then improve them by incorporating a bisection speedup idea. We also modify the methods, at no or only negligible cost in the theoretical complexity, to allow for the "nonrestarting" behavior. In Section 5 we propose more efficient methods, which are grounded in Nesterov's smoothing technique. These are of an order of magnitue faster than those based on the subgradient routine. Next follows Section 6 in which we briefly summarize the theoretical complexities of all methods. The final section is devoted to describing several special cases and applications.

Notation. Throughout the paper, $\mathbf{E}$ (possibly with subscripts) is a finite dimensional real vector space and $\mathbf{E}^*$ is its dual, i.e. the space of all linear functionals on $\mathbf{E}$. The action of $g \in \mathbf{E}^*$ on $x \in \mathbf{E}$ is written as $\langle g, x\rangle$. Coordinates of a vector $y \in \mathbf{R}^l$ are denoted by superscripts in brackets; for example, $y = (y^{(1)}, \ldots, y^{(l)})$, whereas subscripts designate vector labels. By $\mathbf{R}^l_+$ we mean the nonnegative orthant of $\mathbf{R}^l$. More notation is introduced at the relevant spot in the text.

2. Sublinear minimization.

The central problem of this paper is

$$(P) \qquad \boxed{\varphi^* \stackrel{\text{def}}{=} \min_{x \in \mathcal{L}} \varphi(x),}$$

where $\mathcal{L}$ is an affine subspace of a finite-dimensional real vector space $\mathbf{E}$ not containing the origin and $\varphi\colon \mathbf{E} \to \mathbf{R}$ is a *sublinear* function: convex and (positively) homogeneous of degree one. The last property means that the function is linear on every ray emanating from the origin: $\varphi(\tau x) = \tau\varphi(x)$ for all $\tau \geq 0$ and $x \in \mathbf{E}$. Note that convexity and homogeneity imply subadditivity. Define $n := \dim \mathbf{E} = \dim \mathbf{E}^*$.

We will further make the assumption that the zero vector lies in the interior of the (convex) subdifferential of φ evaluated at the origin:

$$0 \in \operatorname{int} \partial\varphi(0). \tag{2.1}$$

Given the properties of φ, condition (2.1) essentially amounts to requiring that the origin is the *unique* global minimizer of φ. The above assumptions imply that $\partial\varphi(0)$ is a full-dimensional compact and convex subset of $\mathbf{E}^*$ and that we can write[1]

$$\varphi(x) = \max\{\langle g, x\rangle \ : \ g \in \partial\varphi(0).\} \tag{2.2}$$

[1]There is a one-to-one correspondence between finite sublinear functions and nonempty compact convex sets via the relation $\varphi(x) = \max\{\langle g, x\rangle \mid g \in \mathcal{G}\}$ (this is the *support function* of $\mathcal{G}$). It then follows from the definition of the subdifferential that $\mathcal{G} = \partial\varphi(0)$. We refer the reader Rockafellar [19]. An detailed account of the properties of sublinear functions and subdifferentials of convex functions can be found in Chapters IV and V of Hiriart-Urruty and Lemaréchal [6]. For a more compact and up-to-date treatment see Borwein and Lewis [3, Corollary 4.2.3].

That is, φ is the *support function* of its subdifferential at the origin. For geometric understanding of the situation implied by the assumptions it is helpful to note that the epigraph of φ is a convex cone in $\mathbf{E} \times \mathbf{R}_+$ whose only intersection with $\mathbf{E} \times \{0\}$ is the origin.

2.1. Approximate solutions. Our aim is to find an approximate solution of (P), within relative error δ. The formal definition of the concept follows.

DEFINITION 1. *Point $x \in \mathcal{L}$ is a* δ-approximate solution *to (P) if*

$$\varphi(x) \leq (1+\delta)\varphi^*.$$

In proving theorems we will often use the following equivalent characterization:

$$\varphi(x) - \varphi^* \leq \frac{\delta}{1+\delta}\varphi(x).$$

2.2. Treating unconstrained convex minimization. The general unconstrained convex minimization problem can be reformulated as a constrained sublinear problem. Let us briefly describe the construction. If $\phi\colon \mathbf{E} \to \mathbf{R}$ is a convex function, its *perspective* is the function $\varphi\colon \mathbf{E} \times \mathbf{R}_{++} \to \mathbf{R}$ defined by

$$\varphi(x) \stackrel{\text{def}}{=} \varphi(y,\tau) = \tau\phi\left(\frac{y}{\tau}\right).$$

It is clearly linear on every feasible ray leaving from the origin. In fact, it can be shown that φ is convex on its domain (see, eg. Hiriart-Urruty and Lemaréchal [6, Proposition 2.2.1]). It is not in general possible to extend φ onto the entire space $\mathbf{E}\times\mathbf{R}$ if we want to preserve both convexity and finiteness. However, there are at least some important classes of functions for which this can be done. Consider the following example.

Example 1. Define

$$\phi(y) = \max\{|\langle a_i, y\rangle + b^{(i)}| \ : \ i = 1,2,\ldots,m\},$$

where $y \in \mathbf{E}$, $a_1,\ldots,a_m \in \mathbf{E}^*$ and $b \in \mathbf{R}^m$. If we let $x = (y,\tau)$ and $a_i' = (a_i, b^{(i)})$ for $i = 1,2,\ldots,m$ then for $\tau > 0$ we obtain

$$\begin{aligned}\varphi(x) = \varphi(y,\tau) = \tau\phi\left(\frac{y}{\tau}\right) &= \tau \max_{1\leq i\leq m} |\langle a_i, y/\tau\rangle + b^{(i)}| \\ &= \max_{1\leq i\leq m} |\langle a_i, y\rangle + b^{(i)}\tau| \\ &= \max_{1\leq i\leq m} |\langle a_i', x\rangle|,\end{aligned}$$

where the last equality defines a new inner product on $\mathbf{E}\times\mathbf{R}$. Clearly, φ can be extended to a sublinear function defined on the entire space. Assumption (2.1) will be satisfied if $0 \in \operatorname{int}\partial\varphi(0) = \operatorname{conv}\{\pm a_i' \ : \ i = 1,2,\ldots,m\}$.

3. Ellipsoidal rounding and key inequalities. As a pre-processing phase, Nesterov [14] first finds a positive definite operator $G\colon \mathbf{E} \to \mathbf{E}^*$ giving rise to a pair of central ellipsoids in $\mathbf{E}^*$, one being contained in $\partial\varphi(0)$ and the other containing it. This can be done, for example, using Khachiyan's algorithm [8], or the recent method of Ahipaşaoğlu, Sun and Todd [1]. This way we find radii $0 < \gamma_0 \leq \gamma_1$ such that

$$\mathcal{B}(G,\gamma_0) \subseteq \partial\varphi(0) \subseteq \mathcal{B}(G,\gamma_1), \tag{3.1}$$

where

$$\mathcal{B}(G,\gamma) \stackrel{\text{def}}{=} \{g \in \mathbf{E}^* \ : \ \sqrt{\langle g, G^{-1}g\rangle} \leq \gamma\}$$

defines an ellipsoid in $\mathbf{E}^*$ of radius γ.

The theoretical guarantees of the algorithms of this paper depend on the quantity $\alpha := \gamma_0/\gamma_1$, which characterizes the quality of the ellipsoidal rounding (3.1). It is clearly always the case that $0 < \alpha \leq 1$, with bigger α corresponding to a tighter rounding and, as will be shown, faster algorithms. The following result, a celebrated theorem of John [7], gives lower bounds on the quality of rounding admitted by full-dimensional convex sets.

PROPOSITION 2 (John [7]). *Any convex body $Q \subset \mathbf{E}^*$ admits a rounding by concentric ellipsoids with $\frac{1}{\alpha} \leq \dim \mathbf{E}^*$. If Q is centrally symmetric, then there exists a rounding with $\frac{1}{\alpha} \leq \sqrt{\dim \mathbf{E}^*}$.*

To see that the above result gives tight bounds, consider the following simple example.

Example 2. The rounding obtained by the inscribed and circumscribed balls of

1. a regular n-simplex has quality $\frac{1}{\alpha} = n$,
2. the n-cube has quality $\frac{1}{\alpha} = \sqrt{n}$.

For recent work related to ellipsoidal rounding see Belloni and Freund [2] and the references therein.

3.1. Geometry induced by rounding. The rounding operator G defines an inner product on $\mathbf{E}$ via $\langle x, y\rangle_G := \langle Gx, y\rangle$, which in turn induces the norm $\|x\|_G := \sqrt{\langle x, x\rangle_G}$. The dual space $\mathbf{E}^*$ can be equipped with the dual norm $\|g\|_G^* := \sqrt{\langle g, G^{-1}g\rangle}$. Notice that these norms are themselves sublinear functions and as such admit a representation similar to (2.2):

$$\|x\|_G = \max\{\langle g, x\rangle \ : \ \|g\|_G^* \leq 1\}, \tag{3.2}$$

with $\partial\|\cdot\|_G(0) = \{g \in \mathbf{E}^* \ : \ \|g\|_G^* \leq 1\}$, and

$$\|g\|_G^* = \max\{\langle g, x\rangle \ : \ \|x\|_G \leq 1\}, \tag{3.3}$$

with $\partial\|\cdot\|_G^*(0) = \{x \in \mathbf{E} \ : \ \|x\|_G \leq 1\}$. Also observe that the first and last sets in (3.1) are balls in $\mathbf{E}^*$, with respect to the dual norm induced by G, of radii γ_0 and γ_1, respectively.

3.2. Subgradients in the primal space. By defining

$$\partial_G\varphi(x) \stackrel{\text{def}}{=} \{h \in \mathbf{E} \ : \ \varphi(y) \geq \varphi(x) + \langle h, x\rangle_G, \quad \forall y \in \mathbf{E}\},$$

the subgradients of φ can be thought of as being elements of $\mathbf{E}$ as opposed to elements of $\mathbf{E}^*$. This will enable us to talk about taking steps in $\mathbf{E}$ in the "direction" of a negative subgradient. Note that there is a one-to-one correspondence linking the two concepts:

$$\partial_G\varphi(x) = G^{-1}[\partial\varphi(x)]. \tag{3.4}$$

3.3. Inequalities. In view of (2.2) and (3.2), taking the maximum of the linear functional $\langle\cdot, x\rangle$ over the sets in (3.1) gives

$$\gamma_0\|x\|_G \leq \varphi(x) \leq \gamma_1\|x\|_G \quad \text{for all} \quad x \in \mathbf{E}, \tag{3.5}$$

which together with subadditivity of φ implies that φ is γ_1-Lipschitz:

$$\varphi(x+h) \leq \varphi(x) + \varphi(h) \leq \varphi(x) + \gamma_1 \|h\|_G.$$

From now on let us adopt the following notation. By x^* we denote an arbitrary optimal solution of (P) and by x_0 we denote the minimum norm element of the feasible region—the projection of the origin onto $\mathcal{L}$. From (3.5) we then obtain

$$\alpha\varphi(x_0) \leq \gamma_0 \|x_0\|_G \leq \gamma_0 \|x^*\|_G \leq \varphi^* \leq \varphi(x_0) \leq \gamma_1 \|x_0\|_G. \tag{3.6}$$

Dividing by γ_0 we get

$$\frac{\varphi(x_0)}{\gamma_1} \leq \|x_0\|_G \leq \|x^*\|_G \leq \frac{\varphi^*}{\gamma_0} \leq \frac{\varphi(x_0)}{\gamma_0}. \tag{3.7}$$

Since $\|x^* - x_0\|_G = \sqrt{\|x^*\|_G^2 - \|x_0\|_G^2}$ and $x_0 \neq 0$ due to the assumption that $\mathcal{L}$ does not pass through the origin, we also obtain

$$\|x^* - x_0\|_G < \|x^*\|_G \leq \frac{\varphi^*}{\gamma_0} \leq \frac{\varphi(x_0)}{\gamma_0}. \tag{3.8}$$

4. Algorithms based on a subgradient subroutine. Subgradient algorithms were studied intensively in the sixties and seventies of the twentieth century by a number of researchers, among them Y.M. Ermoliev, B.T. Polyak and N.Z. Shor. For comprehensive texts, see, e.g. Shor [20] and Goffin [5]. For our purposes we will manage with a result about the performance of a standard constant step-length subgradient algorithm applied to a convex Lipschitz function. This algorithm, together with a simple proof, can be found, for example, in Nesterov [11, Section 3.2.3].

In the first subsection we start by briefly discussing the constant step-length subgradient method and its performance guarantee.

4.1. A constant step-length subgradient algorithm. The subgradient algorithm we are going to describe works in a more general setting than that of problem (P). For the sake of this subsection only, consider the problem of minimizing a convex Lipschitz continuous function $\varphi\colon \mathbf{E} \to \mathbf{R}$ with Lipschitz constant γ over a *simple* closed convex set Q_1:

$$\boxed{\varphi^* \stackrel{\text{def}}{=} \min\{\varphi(x) \ : \ x \in Q_1\}.} \tag{P_{sg}}$$

By simple set we mean one allowing for easy computation of projections onto it (symbol proj will denote the projection operator). In this setting $\mathbf{E}$ is assumed to be equipped with an inner product. Problem (P) is a special case of (P_{sg}) with

- φ having additional properties,
- $\gamma = \gamma_1$ and $Q_1 = \mathcal{L}$, and
- $\mathbf{E}$ made Euclidean by the introduction of the inner product induced by G.

The following is a standard result (see, e.g. Nesterov [11, Theorem 3.2.2]).

PROPOSITION 3. *If $\|x^* - x_0\| \leq R$ for some $x_0 \in \mathbf{E}$, minimizer x^* of (P_{sg}) and $R > 0$, then the output*

$$x = \mathsf{Subgrad}(\varphi, Q_1, x_0, R, N)$$

of Algorithm 1 run on an instance of problem P_{sg} satisfies:

$$\varphi(x) - \varphi^* \leq \frac{\gamma R}{\sqrt{N+1}}. \tag{4.1}$$

For Proposition 3 to hold it suffices to require that φ be Lipschitz on the ball around x^* with radius R.

Algorithm 1 (Subgrad) Constant step-length subgradient scheme

Input: φ, Q_1, x_0, R, N;
$\kappa = R/\sqrt{N+1}$;
for $k = 0$ **to** $N-1$
  pick $g \in \partial\varphi(x_k)$; **if** $g = 0$ **then** x_k is optimal and **exit**;
  $x_{k+1} = \mathrm{proj}_{Q_1}\left(x_k - \kappa \frac{g}{\|g\|}\right)$;
end for
Output: x_k with best objective value

4.2. Basic algorithmic ideas. As the previous subsection indicates, the basic idea for solving (P) will be that of using the subgradient method (Algorithm 1). The main issue with this algorithm, apart from the fact that it is slow (it requires $O(1/\epsilon^2)$ iterations to output an ϵ-optimal solution in the additive sense), is the need to supply an initial point x_0 and bound R satisfying $\|x^* - x_0\| \leq R$.

The particular choice of x_0 as the projection of the origin onto the feasible set of (P) makes sense from at least two reasons. First, notice that if the ellipsoidal rounding of $\partial\varphi(0)$ is perfectly tight ($\alpha = 1$), then by (3.5) we have $\varphi(x) \equiv \|x\|_G$ and therefore x_0 *is* the optimal solution of (P). In fact, notice that (3.7) implies

$$\varphi(x_0) \leq \frac{\varphi^*}{\alpha}, \tag{4.2}$$

and hence x_0 is a $(\frac{1}{\alpha} - 1)$-approximate solution of (P). The better the rounding, the better the guarantee. Second, (3.8) offers the readily available upper bound $R = \varphi(x_0)/\gamma_0$. Of course, φ^*/γ_0 would be better, but we do not know it.

Good but unavailable upper bound. Let us formally apply Algorithm 1 to (P) with $R = \varphi^*/\gamma_0$. To achieve the required relative accuracy, it then suffices to run it for $N = \lfloor \alpha^{-2}\delta^{-2} \rfloor$ iterations because by Proposition 3

$$\varphi(x) - \varphi^* \leq \frac{\gamma_1 R}{\sqrt{N+1}} \leq \frac{\varphi^*}{\alpha\sqrt{\frac{1}{\alpha^2\delta^2}}} = \delta\varphi^*.$$

Available but bad upper bound. Since the previous upper bound is unknown, it seems reasonable to instead use the worse (but available) bound $R = \varphi(x_0)/\gamma_0$. If we wish to guarantee a solution within relative error δ, we need to take $N = \lfloor \alpha^{-4}\delta^{-2} \rfloor$ iterations. The argument is exactly the same as in the case above except we start by replacing $\varphi(x_0)$ with φ^*/α in view of (4.2).

Iteratively updated upper bound. To move towards the better of the two extremes, Nesterov [14] proposed a scheme (Algorithm 2) which uses the subgradient method as a subroutine and which iteratively decreases the known upper bound. His algorithm starts by running the subgradient method for $O(\alpha^{-2}\delta^{-2})$ iterations with the available upper bound $\varphi(x_0)/\gamma_0$. In the case when the subgradient subroutine is doing well and manages to decrease the objective value by a constant fraction, the previously available upper bound also decreases by the same fraction. This improved bound is then used to run the next subgradient subroutine, again starting from x_0.

The performance of Algorithm 2 is substantially better than the naive one-time application of the subgradient method with the bad but available upper bound. Of course, it underperforms the one-time application of the subgradient method with

Algorithm 2 (SubSearch) Subgradient search scheme.

1: **Input:** $\varphi, \mathcal{L}, x_0, \gamma_0, \gamma_1, \beta > 0, \delta$;
2: $\hat{x}_0 = x_0,\ \alpha = \gamma_0/\gamma_1,\ c = e^\beta,\ k = 1$;
3: $N = \left\lfloor \frac{c^2}{\alpha^2}\left(1+\frac{1}{\delta}\right)^2 \right\rfloor$;
4: $\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, \varphi(\hat{x}_{k-1})/\gamma_0, N)$;
5: **while** $\varphi(\hat{x}_k) < \frac{1}{c}\varphi(\hat{x}_{k-1})$ **do**
6: $\quad k = k + 1$;
7: $\quad \hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, \varphi(\hat{x}_{k-1})/\gamma_0, N)$;
8: **end while**
9: **Output:** $\hat{x}_k$

the good but unknown upper bound, by a factor of $O(\ln \frac{1}{\alpha})$. The performance of the method, as analyzed by Nesterov [14], is summarized in Proposition 4. We include the proof because it is short and offers insight into the subsequent improvements we propose in the following subsections. We will also refer to parts of it later.

PROPOSITION 4 (Nesterov [14, Theorem 3]). *Algorithm 2 returns a δ-approximate solution of (P) and takes at most*

$$\frac{e^{2\beta}}{\alpha^2}\left(1+\frac{1}{\delta}\right)^2\left(1+\frac{1}{\beta}\ln\frac{1}{\alpha}\right)$$

steps of the subgradient method. If β is chosen to be a constant, then the number of steps is

$$O\left(\frac{1}{\alpha^2\delta^2}\ln\frac{1}{\alpha}\right). \tag{4.3}$$

Proof. Assume that the algorithm stops at iteration k, failing to satisfy the while clause at Step 5. In view of (3.6) we have

$$\alpha\varphi(x_0) \le \varphi^* \le \varphi(\hat{x}_{k-1}) < \left(\frac{1}{c}\right)^{k-1}\varphi(x_0),$$

and by comparing the first and the last term in this chain of inequalities we conclude that the number of calls of the subgradient subroutine is at most $1 + \frac{1}{\beta}\ln\frac{1}{\alpha}$. The bound on the number of lower level steps is obtained by multiplying this by N from Step 3 of the algorithm. It remains to show that the output is as specified. Indeed, using the termination rule from Step 5 and applying Proposition 3 to the last call of the subgradient subroutine we get

$$\varphi(\hat{x}_k) - \varphi^* \le \frac{\gamma_1\frac{\varphi(\hat{x}_{k-1})}{\gamma_0}}{\sqrt{N+1}} \le \frac{\frac{e^\beta}{\alpha}\varphi(\hat{x}_k)}{\sqrt{N+1}} \le \frac{\delta}{1+\delta}\varphi(\hat{x}_k). \qquad \square$$

4.3. Bisection improvement. Each outer iteration of Algorithm 2, possibly except the last one, produces a *guaranteed* upper bound on the distance of x_0 from the set of minimizers of (P)—better by a constant factor than the one available before. Loosely speaking, we will show that by allowing for *guesswork* it is possible to get a theoretical and practical improvement in the performance of this algorithm (the same improvement was independently obtained by Chudak and Eleutério [4] in the context

of combinatorial applications). The key observation is formulated in the following lemma.

LEMMA 5. *If $\varphi^*/\gamma_0 \leq R$ and $N = \lfloor \alpha^{-2}\beta^{-2} \rfloor$ for some $\beta > 0$, then*

$$x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, R, N)$$

satisfies

$$\frac{\varphi(x)}{\gamma_0} \leq (1+\beta)R.$$

Proof. By Proposition 3 we have $\varphi(x) - \varphi^* \leq \gamma_1 R/\sqrt{N+1} \leq \gamma_0 \beta R$ and hence

$$\frac{\varphi(x)}{\gamma_0} \leq \frac{\varphi^*}{\gamma_0} + \beta R \leq R(1+\beta). \qquad \square$$

The above result essentially states that for *any* positive R we can, at the cost of $O(\alpha^{-2}\beta^{-2})$ iterations of the subgradient method (Algorithm 1), either get a certificate that $\varphi^*/\gamma_0 \leq (1+\beta)R$ or that $R \leq \varphi^*/\gamma_0$. In any case we either get a *new* upper or lower bound on φ^*/γ_0. The *initial* lower and upper bounds come from (3.7): if we set $L_0 := \|x_0\|_G$ and $R_0 := \varphi(x_0)/\gamma_0$ then

$$\frac{\varphi(x_0)}{\gamma_1} \leq L_0 \leq \frac{\varphi^*}{\gamma_0} \leq R_0,$$

with $q_0 := R_0/L_0 \leq \frac{1}{\alpha}$. Assuming $(1+\beta)R \leq R_0$, the new lower and upper bounds are either $(L_1, R_1) = (L_0, (1+\beta)R)$, or $(L_1, R_1) = (R, R_0)$, depending on the outcome of the procedure suggested in Lemma 5 (see Figure 1). This bisection step is then repeated until the ratio $q_k := R_k/L_k$ gets down to a sufficiently small value. It turns out that it is efficient to choose $\beta = \theta(1)$ and bisect only until q_k decreases down to a constant value and then "finish the job" by taking $O(\alpha^{-2}\delta^{-2})$ additional subgradient steps, much in the way as we have seen with the "good but unavailable" upper bound.

The following lemma states how much of improvement in q_k can be obtained by a single bisection step.

LEMMA 6. *Assume L_{k-1} and R_{k-1} are lower and upper bounds on φ^*/γ_0, respectively, with $q_{k-1} > 1+\beta$, and let*

$$R := \sqrt{\frac{L_{k-1}R_{k-1}}{1+\beta}}.$$

If we run the subgradient method as indicated in Lemma 5 and if L_k and R_k are the new bounds, then

$$q_k \leq \sqrt{1+\beta}\sqrt{q_{k-1}}. \tag{4.4}$$

Proof. First notice that the assumption $q_{k-1} > 1+\beta$ implies that $L_{k-1} < R < (1+\beta)R < R_{k-1}$. Recall that we either have $(L_k, R_k) = (L_{k-1}, (1+\beta)R)$ or $(L_k, R_k) = (R, R_{k-1})$ and observe that R is chosen so that the value of q_k is the same under both eventualities:

$$\frac{(1+\beta)R}{L_{k-1}} = \frac{R_{k-1}}{R}.$$

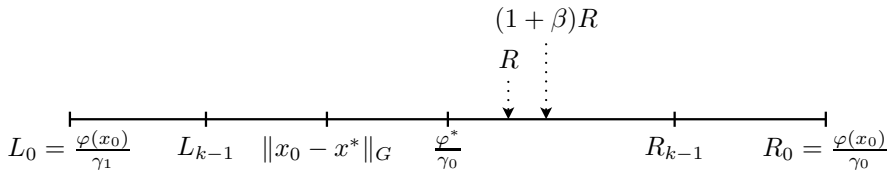

FIG. 1. *Bisection step k.*

Putting these observations together,

$$q_k = \frac{R_{k-1}}{R} = \sqrt{1+\beta}\sqrt{q_{k-1}}. \qquad \square$$

The ideas outlined above lead to Algorithm 3 whose performance is analyzed in the next theorem.

Algorithm 3 (SubBis) Subgradient bisection scheme.

Input: $\varphi, \mathcal{L}, x_0, \gamma_0, \gamma_1, \beta, \delta$;
$k = 0,\ \hat{x}_0 = x_0,\ L_0 = \|x_0\|_G,\ R_0 = \varphi(x_0)/\gamma_0$;
$\alpha = \gamma_0/\gamma_1,\ c = 2(1+\beta),\ N = \left\lfloor \frac{1}{\alpha^2\beta^2} \right\rfloor$;
while $R_k/L_k > c$ **do**
  $k = k+1,\ R = \sqrt{\frac{L_{k-1}R_{k-1}}{1+\beta}},\ x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, R, N)$;
  if $\varphi(x)/\gamma_0 \le (1+\beta)R$ **then**
    $R_k = \varphi(x)/\gamma_0,\ L_k = L_{k-1},\ \hat{x}_k = x$;
  else
    $L_k = R$;
    if $\varphi(x)/\gamma_0 \le R_{k-1}$ **then**
      $R_k = \varphi(x)/\gamma_0,\ \hat{x}_k = x$;
    else
      $R_k = R_{k-1},\ \hat{x}_k = \hat{x}_{k-1}$;
    end if
  end if
end while
$N = \left\lfloor \frac{c^2}{\alpha^2}\left(1+\frac{1}{\delta}\right)^2 \right\rfloor,\ \hat{x}_{k+1} = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, R, N)$;
Output: $\hat{x}_{k+1}$

THEOREM 7. *Algorithm 3 returns a δ-approximate solution of (P) and takes at most*

$$\frac{1}{\alpha^2\beta^2}\left(1 + \log_2 \log_2 \frac{1}{\alpha}\right) + \frac{4(1+\beta)^2}{\alpha^2}\left(1+\frac{1}{\delta}\right)^2$$

steps of the subgradient subroutine. If β is chosen to be a constant, then the number of steps is

$$O\left(\frac{1}{\alpha^2\delta^2} + \frac{1}{\alpha^2}\ln\ln\frac{1}{\alpha}\right). \tag{4.5}$$

Proof. Let us first analyze the bisection phase (the **while** loop). Repeated use of Lemma 6 gives

$$q_k \le (1+\beta)^{\frac{1}{2}} q_{k-1}^{\frac{1}{2}} \le (1+\beta)^{\frac{1}{2}}(1+\beta)^{\frac{1}{4}} q_{k-2}^{\frac{1}{4}} \le \cdots \\ \le (1+\beta)^{\frac{1}{2}}(1+\beta)^{\frac{1}{4}} \cdots (1+\beta)^{\frac{1}{2^k}} q_0^{\frac{1}{2^k}} \le (1+\beta)\left(\tfrac{1}{\alpha}\right)^{\frac{1}{2^k}}.$$

The smallest integer k for which $(1+\beta)\left(\frac{1}{\alpha}\right)^{\frac{1}{2^k}} \le 2(1+\beta)$ is $k^* := \left\lceil \log_2 \log_2 \left(\frac{1}{\alpha}\right)\right\rceil$ and hence the total number of lower-level subgradient method iterations of the bisection phase is at most $N_{\text{bis}} = \frac{1}{\alpha^2\beta^2}\left(1+\log_2\log_2\left(\frac{1}{\alpha}\right)\right)$. The guarantee (4.5) follows by adding N_{bis} and the number of iterations needed for the finalization phase (Step 17). It remains to show that the output of the algorithm is as specified. Notice that $\varphi(\hat{x}_{k+1})/\gamma_0 \in [L_k, R_k] = [L_k, \varphi(\hat{x}_k)/\gamma_0]$ and hence

$$\frac{\varphi(\hat{x}_k)}{\varphi(\hat{x}_{k+1})} \le \frac{R_k}{L_k} = q_k \le c.$$

Now we just need to apply Proposition 3 to the subgradient subroutine call of Step 17 of the algorithm using the inequality above:

$$\varphi(\hat{x}_{k+1}) - \varphi^* \le \frac{\gamma_1}{\sqrt{N+1}}\frac{\varphi(\hat{x}_k)}{\gamma_0} \le \frac{1}{\sqrt{N+1}}\frac{c\varphi(\hat{x}_{k+1})}{\alpha} \le \frac{\delta}{1+\delta}\varphi(\hat{x}_{k+1}). \qquad \square$$

4.4. Nonrestarting algorithms. Algorithms SubSearch and SubBis (Algorithms 2 and 3) use the subgradient subroutine *always started from one point*, denoted x_0, which is defined as the projection of the origin onto the feasibility set. This point is indeed special as it allows for the key inequalities (3.7) and (3.8), which in turn drive the analysis in both algorithms. The first of these inequalities makes x_0 indispensable as the starting point of the very *first* subgradient subroutine call in both algorithms, making it possible to construct initial lower and upper bounds on φ^*/γ_0. It is hard to think of a different readily computable point that could serve the same purpose.

The issue we are going to touch upon in this subsection concerns the use of x_0 as the starting point in all *subsequent* calls of the subroutine. In our view, *restarting* from this particular point seems to be convenient for the sake of the proofs rather than efficient algorithmically. Let us elaborate on this a bit. Both algorithms mentioned above can be viewed as simultaneously optimizing (solving (P)) and searching for a good upper bound on $\|x_0 - x^*\|_G$ in order to look less like the "do-it-all-with-the-available-but-bad-upper-bound" and more like the "do-it-all-with-the-good-but-unavailable-upper-bound" algorithm. Combining these two goals is possible because φ^*/γ_0 is both the optimal value of (P) (up to the known constant factor γ_0) and an upper bound on $\|x_0 - x^*\|_G$. It seems likely that the optimization goal could be attained faster if we could use the current best point, as opposed to x_0, to start every call of the subroutine. Although both algorithms gather information about increasingly better iterates $\{\hat{x}_k\}$, this knowledge is used only to update the upper bound on $\|x_0 - x^*\|_G$ in the next call of the subgradient subroutine and not to start the subroutine itself from a better point. There is a good reason for that though. Even if some point $\hat{x}_k$ obtained along the way in one of the algorithms was much better than x_0 in terms of its objective value, there are no theoretical guarantees that $\|\hat{x}_k - x^*\|_G$ will be smaller. Starting the subgradient subroutine from such a point thus means combining a probable advantage with a possible disadvantage. A simple

observation reveals that the disadvantage factor is under control. Note that for any feasible $\hat{x}_k$,

$$\|\hat{x}_k - x^*\|_G \le \|\hat{x}_k\|_G + \|x^*\|_G \le \frac{\varphi(\hat{x}_k)}{\gamma_0} + \frac{\varphi^*}{\gamma_0} \le 2\frac{\varphi(\hat{x}_k)}{\gamma_0}.$$

This means that whenever the subgradient method outputs some point $\hat{x}_k$, we have an upper bound on $\|\hat{x}_k - x^*\|_G$ available. Therefore, on the next call we can run the method starting at $\hat{x}_k$ with $R = 2\varphi(\hat{x}_k)/\gamma_0$, which is exactly twice the upper bound we would use when restarting from x_0.

Nonrestarting version of SubSearch**.** Algorithm 4 is a modified version of Algorithm 2 in the spirit of the above discussion. The theoretical performance stays the same.

Algorithm 4 (SubSearchNR) Nonrestarting subgradient search scheme.

Input: $\varphi, \mathcal{L}, x_0, \gamma_0, \gamma_1, \delta$;
$\hat{x}_0 = x_0,\ \alpha = \gamma_0/\gamma_1,\ c = \sqrt{e},\ k = 1$;
$N = \left\lfloor \frac{c^2}{\alpha^2}\left(1 + \frac{1}{\delta}\right)^2 \right\rfloor,\ R = \varphi(\hat{x}_0)/\gamma_0$;
$\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_0, R, N)$;
while $\varphi(\hat{x}_k) < \frac{1}{c}\varphi(\hat{x}_{k-1})$ **do**
  $k = k + 1$;
  $N = \left\lfloor \frac{4c^2}{\alpha^2}\left(1 + \frac{1}{\delta}\right)^2 \right\rfloor,\ R = 2\varphi(\hat{x}_{k-1})/\gamma_0$;
  $\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_{k-1}, R, N)$;
end while
Output: $\hat{x}_k$

THEOREM 8. *Algorithm 4 outputs a δ-approximate solution of (P). The number of calls of the subgradient subroutine is at most $1 + 2\ln\frac{1}{\alpha}$ and the total number of lower-level subgradient steps is hence at most*

$$\frac{4e}{\alpha^2}\left(1 + \frac{1}{\delta}\right)^2\left(1 + 2\ln\frac{1}{\alpha}\right) = O\left(\frac{1}{\alpha^2\delta^2}\ln\frac{1}{\alpha}\right). \tag{4.6}$$

Proof. The proof of the upper bound on the number of the outer level iterations is exactly the same as for Algorithm 2. If the algorithm terminates with $k = 1$, it is identical to Nesterov's, and the result follows (we can drop the constant 4 in this case). If $k > 1$, the analysis is analogous except the $2c$ (instead of just c) in the definition of N and 2 in the definition of R cancel out:

$$\varphi(\hat{x}_k) - \varphi^* \le \frac{\gamma_1 R}{\sqrt{N+1}} \le \frac{\gamma_1}{\frac{2c}{\alpha}\left(1 + \frac{1}{\delta}\right)}\frac{2\varphi(\hat{x}_{k-1})}{\gamma_0} \le \frac{1}{c\left(1 + \frac{1}{\delta}\right)}c\varphi(\hat{x}_k) = \frac{\delta}{1+\delta}\varphi(\hat{x}_k). \qquad \square$$

Nonrestarting bisection algorithm. The following fact plays the role of Lemma 5 in the design and analysis of a nonrestarting bisection algorithm (Algorithm 5).

LEMMA 9. *Let $\hat{x}_{k-1} \in \mathcal{L}$ be arbitrary. If $\varphi^*/\gamma_0 \le R$ and $N = \lfloor \alpha^{-2}\beta^{-2} \rfloor$ for some $\beta > 0$, then*

$$\hat{x}_k := \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_{k-1}, R + \|\hat{x}_{k-1}\|_G, N)$$

satisfies

$$\frac{\varphi(\hat{x}_k)}{\gamma_0} \leq (1+\beta)R + \beta\|\hat{x}_{k-1}\|_G \leq (1+\beta)R + \beta\frac{\varphi(\hat{x}_{k-1})}{\gamma_0}. \tag{4.7}$$

Proof. First notice that $\|\hat{x}_{k-1} - x^*\|_G \leq \|\hat{x}_{k-1}\|_G + \|x^*\|_G \leq \|\hat{x}_{k-1}\|_G + \varphi^*/\gamma_0 \leq \|\hat{x}_{k-1}\|_G + R$ and hence by Proposition 3,

$$\varphi(\hat{x}_k) - \varphi^* \leq \gamma_1 \frac{R + \|\hat{x}_{k-1}\|_G}{\sqrt{N+1}} \leq \gamma_0\beta(R + \|\hat{x}_{k-1}\|_G).$$

Dividing the above inequality by γ_0 and rearranging the expression gives the result. The second inequality follows from (3.5). ∎

The idea with updating lower and upper bounds is the same as in the restarting version of the algorithm. Let $q_k := R_k/L_k$, as before. The improvement guaranteed by a single bisection step is given in the following result.

LEMMA 10. *Assume L_{k-1} and $R_{k-1} = \varphi(\hat{x}_{k-1})/\gamma_0$ are lower and upper bounds on φ^*/γ_0, respectively, with $q_{k-1} > 2(1+\beta)$, and let*

$$R := \sqrt{\frac{L_{k-1}R_{k-1}}{1+\beta}}.$$

If we run the subgradient method as indicated in Lemma 9 and if L_k and R_k are the new bounds, then

$$q_k \leq \left(\sqrt{\tfrac{1}{2}} + \beta\right) q_{k-1}. \tag{4.8}$$

Proof. Because $q_{k-1} > 2(1+\beta) > 1+\beta$, we are in the same situation as in Lemma 6 and so $L_{k-1} < R < (1+\beta)R < R_{k-1}$. Notice that the upper bound gets always updated to the value corresponding to the best point found so far, that is, $R_k = \varphi(\hat{x}_k)/\gamma_0$. So we either have $R_k \leq (1+\beta)R + \beta R_{k-1}$, in which case the lower bound stays unchanged, or $L_k = R$ (and $R_k \leq R_{k-1}$, possibly with equality). Therefore

$$q_k = \frac{R_k}{L_k} \leq \max\left\{\frac{(1+\beta)R + \beta R_{k-1}}{L_{k-1}}, \frac{R_{k-1}}{R}\right\}.$$

Notice that R is chosen so that the two expressions in the maximum above are equal, neglecting the βR_{k-1} portion of the first. The first expression must therefore be bigger and hence

$$\begin{aligned} q_k \leq \frac{(1+\beta)R + \beta R_{k-1}}{L_{k-1}} &= \sqrt{1+\beta}\sqrt{q_{k-1}} + \beta q_{k-1} \\ &= \left(\sqrt{\tfrac{1+\beta}{q_{k-1}}} + \beta\right) q_{k-1} < \left(\sqrt{\tfrac{1}{2}} + \beta\right) q_{k-1}. \end{aligned}$$ ∎

THEOREM 11. *Algorithm 5 run with β chosen to be a constant such that $\hat{\beta} := \sqrt{\frac{1}{2}} + \beta < 1$ returns a δ-approximate solution of (P) and takes*

$$O\left(\frac{1}{\alpha^2\delta^2} + \frac{1}{\alpha^2}\ln\frac{1}{\alpha}\right) \tag{4.9}$$

steps of the subgradient subroutine.

Proof. Let us first analyze the bisection phase. Repeated use of Lemma 10 gives

$$q_k \le \hat{\beta}^k q_0 \le \hat{\beta}^k \tfrac{1}{\alpha}.$$

The smallest integer k for which the last quantity drops below $c = 2(1+\beta)$ is

$$k^* := \left\lceil \frac{\ln(\alpha^{-1}c^{-1})}{\ln \hat{\beta}^{-1}} \right\rceil = O\left(\ln \frac{1}{\alpha}\right),$$

and hence the total number of lower-level subgradient method iterations of the bisection phase is $N_{\text{bis}} = O\left(\frac{1}{\alpha^2} \ln \frac{1}{\alpha}\right)$. The guarantee (4.9) follows by adding N_{bis} and the number of iterations needed for the finalization phase (Step 12). It remains to show that the output of the algorithm is as specified. The analysis, however, is identical to that in Theorem 8. ∎

Note that the nonrestarting version of the bisection algorithm has a slightly worse complexity bound—we have lost one logarithm in (4.9) in comparison with (4.5). However, the bisection strategy still manages to separate the δ from the logarithmic term as compared to the bound (4.6) for the SubSearch algorithm.

Algorithm 5 (SubBisNR) Nonrestarting subgradient bisection scheme.

1: **Input:** $\varphi, \mathcal{L}, x_0, \gamma_0, \gamma_1, \beta, \delta$;
2: $k = 0,\ \hat{x}_0 = x_0,\ L_0 = \|x_0\|_G,\ R_0 = \varphi(x_0)/\gamma_0$;
3: $\alpha = \gamma_0/\gamma_1,\ c = 2(1+\beta),\ N = \left\lfloor \frac{1}{\alpha^2\beta^2} \right\rfloor$;
4: **while** $R_k/L_k > c$ **do**
5: $\quad k = k+1,\ R = \sqrt{\frac{L_{k-1}R_{k-1}}{1+\beta}},\ \hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_{k-1}, R, N)$;
6: $\quad$ **if** $\varphi(\hat{x}_k)/\gamma_0 \le (1+\beta)R + \beta\varphi(\hat{x}_{k-1})/\gamma_0$ **then**
7: $\quad\quad L_k = L_{k-1},\ R_k = \varphi(\hat{x}_k)/\gamma_0$;
8: $\quad$ **else**
9: $\quad\quad L_k = R,\ R_k = \varphi(\hat{x}_k)/\gamma_0$;
10: $\quad$ **end if**
11: **end while**
12: $N = \left\lfloor \frac{4c^2}{\alpha^2}\left(1+\frac{1}{\delta}\right)^2 \right\rfloor,\ R = \frac{2\varphi(\hat{x}_k)}{\gamma_0},\ \hat{x}_{k+1} = \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_k, R, N)$;
13: **Output:** $\hat{x}_{k+1}$

5. Algorithms based on smoothing. We have seen in Section 4 that problem (P) allows for simple algorithms that require $O(\delta^{-2})$ iterations of the subgradient method. We have improved Nesterov's subgradient search algorithm (Algorithm 2), which needs $O(\alpha^{-2}\delta^{-2}\ln\frac{1}{\alpha})$ iterations, by incorporating a simple bisection idea and obtained Algorithm 3 with the slightly better $O(\alpha^{-2}\delta^{-2}+\delta^{-2}\ln\ln\frac{1}{\alpha})$ guarantee. That is, we have improved the dependence on the rounding parameter α, but not on the error parameter δ.

We start in the following subsection by briefly describing Nesterov's smoothing technique [12] and the implied algorithm for smooth minimization of nonsmooth functions. It is not our intention to describe the approach in full generality; rather, we will adapt the results to the setting of problem (P)—the minimization of a nonnegative sublinear (convex and homogeneous) function vanishing only at the origin.

5.1. The setting. Nesterov [12] considers a rather general *nonsmooth* convex optimization problem and shows that it is possible to solve it in $O(\epsilon^{-1})$ iterations of a gradient-type method, if a solution within absolute error ϵ is sought. His novel approach involves two phases. The first is a pre-processing phase in which one approximates the objective function by a smooth function with Lipschitz continuous gradient. The second phase amounts to running an optimal smooth method [10, 11] with complexity $O(\epsilon^{-1/2})$ applied to the smooth function.

We will describe the model for sublinear functions. Consider the following more general version of problem (P), with φ replaced by an arbitrary sublinear function and $\mathcal{L}$ (or $\mathcal{L}$ intersected with a large ball) replaced by a compact and convex subset Q_1 of $\mathbf{E}_1 := \mathbf{E}$:

$$(P') \qquad \boxed{\varphi^* := \min_x \{\varphi(x) \ : \ x \in Q_1\}.}$$

Notice that φ can be written as

$$\varphi(x) = \max_g \{\langle g, x\rangle \ : \ g \in \partial\varphi(0)\}, \tag{5.1}$$

To allow for some modeling flexibility, the purpose of which will be clear later, we will instead consider the following family of representations of the objective function:

$$\varphi(x) = \max_y \{\langle Ax, y\rangle \ : \ y \in Q_2\}. \tag{5.2}$$

Here we are introducing a new finite-dimensional real vector space $\mathbf{E}_2$, a linear operator $A\colon \mathbf{E}_1 \to \mathbf{E}_2^*$ and a compact and convex set $Q_2 \subset \mathbf{E}_2$.

DEFINITION 12. *The* adjoint *of A is the operator $A^* : \mathbf{E}_2 \to \mathbf{E}_1^*$ defined via*

$$\langle Ax, y\rangle = \langle A^* y, x\rangle \qquad \forall\, x \in \mathbf{E}_1,\ y \in \mathbf{E}_2.$$

We assume that the spaces $\mathbf{E}_1$ and $\mathbf{E}_2$ are equipped with norms $\|\cdot\|_1$ and $\|\cdot\|_2$ respectively[2], and the dual spaces $\mathbf{E}_1^*$ and $\mathbf{E}_2^*$ with the corresponding dual norms

$$\|g\|_1^* := \max\{\langle g, x\rangle \ : \ \|x\|_1 \le 1\} \quad \text{and} \quad \|h\|_2^* := \max\{\langle h, y\rangle \ : \ \|y\|_2 \le 1\}, \tag{5.3}$$

for $g \in \mathbf{E}_1^*$ and $h \in \mathbf{E}_2^*$.

DEFINITION 13. *The norm of A is defined by*

$$\|A\|_{1,2} := \max_{x,y} \{\langle Ax, y\rangle \ : \ \|x\|_1 = 1,\ \|y\|_2 = 1\}. \tag{5.4}$$

One can similarly define $\|A^*\|_{2,1}$. It follows easily from the definition that

$$\|A\|_{1,2} = \max_x \{\|Ax\|_2^* \ : \ \|x\|_1 = 1\} = \|A^*\|_{2,1} = \max_y \{\|A^* y\|_1^* \ : \ \|y\|_2 = 1\}. \tag{5.5}$$

Example 3. Consider the function

$$\varphi_\infty(x) := \max_i \{|\langle a_i, x\rangle| \ : \ i = 1, 2, \ldots, m\},$$

where $x \in \mathbf{E}_1 = \mathbf{R}^n$, $a_i \in \mathbf{E}_1^* = \mathbf{R}^n$ and $\langle g, x\rangle = \sum_{i=1}^n g^{(i)} x^{(i)}$. Note that in the following three representations of φ_∞ the structure of the set Q_2 gets simpler as the dimension of the space $\mathbf{E}_2$ increases.

[2]The numbers are subscripts referring to the spaces in which the norms are defined and are not intended to suggest the use of the ℓ_1 and ℓ_2 norms.

1. $\mathbf{E}_2 = \mathbf{E}_2^* = \mathbf{R}^n$, $Q_2 = \mathrm{conv}\{\pm a_i \ : \ i = 1, 2, \ldots, m\}$ and $A = I$. This seems to be the most natural and straightforward representation.
2. $\mathbf{E}_2 = \mathbf{E}_2^* = \mathbf{R}^m$, $Q_2 = \{y \in \mathbf{R}^m \ : \ \sum_{i=1}^m |y^{(i)}| \le 1\}$ and A is the $m \times n$ matrix with rows $a_1, \ldots, a_m$. In this case we have
$$\varphi_\infty(x) = \max\left\{\sum_{i=1}^m y^{(i)}\langle a_i, x\rangle \ : \ \sum_{i=1}^m |y^{(i)}| \le 1\right\}.$$
3. $\mathbf{E}_2 = \mathbf{E}_2^* = \mathbf{R}^{2m}$, Q_2 is the unit simplex in $\mathbf{R}^{2m}$ and A is the $2m \times n$ matrix with rows composed of $a_1, \ldots, a_m$ and $-a_1, \ldots, -a_m$:
$$\varphi_\infty(x) = \max\left\{\sum_{i=1}^m (y_1^{(i)} - y_2^{(i)})\langle a_i, x\rangle \ : \ \sum_{i=1}^m y_1^{(i)} + y_2^{(i)} = 1, \ y_1^{(i)}, y_2^{(i)} \ge 0\right\}.$$

If we let
$$\theta(y) \stackrel{\mathrm{def}}{=} \min_x\{\langle A^* y, x\rangle \ : \ x \in Q_1\},$$
then because both Q_1 and Q_2 are convex and compact and $\langle A^* y, x\rangle \equiv \langle y, Ax\rangle$ is bilinear, we can apply a standard minimax result and rewrite (P') as follows:

$$(P'') \qquad \boxed{\varphi^* = \theta^* \stackrel{\mathrm{def}}{=} \max_y\{\theta(y) \ : \ y \in Q_2\}.}$$

5.2. Smoothing and an efficient smooth method. In the first phase of Nesterov's approach, the objective function of (P') is approximated by a smooth convex function with Lipschitz continuous gradient. An approximation with error $O(\epsilon)$ has gradient with Lipschitz constant of $O(1/\epsilon)$. The second phase consists of applying to (P) (with the objective function replaced by its smooth approximation) an efficient smooth method (Algorithm 6) requiring $O(1/\sqrt{\epsilon})$ iterations of a gradient type. The smooth algorithm is capable of producing points $\hat{x}$ and $\hat{g}$ feasible to *both* (P') and (P''), respectively, such that $\varphi(\hat{x}) - \theta(\hat{g}) = O(1/\epsilon)$. Because $\varphi^* = \theta^*$, these points are approximate optimizers in their respective problems (in the absolute sense).

The smoothing approach assumes the availability of *prox-functions* d_1 and d_2 for the sets Q_1 and Q_2, respectively. These are continuous and strongly convex nonnegative functions defined on these sets, with convexity parameters σ_1 and σ_2, respectively. Let x_0 be the *center* of the set Q_1 (think $Q_1 = \mathcal{L}$):

$$x_0 := \arg\min_x\{d_1(x) \ : \ x \in Q_1\}. \tag{5.6}$$

For example, if $d_1(x) := \frac{1}{2}\|x\|_1^2$ (so $\sigma_1 = 1$) and Q_1 is the intersection of $\mathcal{L}$ and a large-enough ball centered at the origin, then x_0 coincides with its earlier definition. We assume that d_1 vanishes at its center and hence the above properties imply
$$d_1(x) \ge \tfrac{1}{2}\sigma_1\|x - x_0\|_1^2.$$
In the example above, we subtract $\|x_0\|_1^2/2$ from d_1 and then the inequality holds as an equation. In an analogous fashion we define the center y_0 of Q_2 and assume that d_2 vanishes at y_0. Therefore
$$d_2(y) \ge \tfrac{1}{2}\sigma_2\|y - y_0\|_2^2.$$

Finally, let D_1 and D_2 satisfy

$$D_1 \geq \max_x \{d_1(x) \ : \ x \in Q_1\}$$

and

$$D_2 \geq \max_y \{d_2(y) \ : \ y \in Q_2\}.$$

PROPOSITION 14 (Nesterov [12], Theorem 1). *For $\mu > 0$, the function*

$$\varphi_\mu(x) := \max_y \{\langle Ax, y\rangle - \mu d_2(y) \ : \ y \in Q_2\}, \tag{5.7}$$

is a continuously differentiable uniform approximation of φ:

$$\varphi_\mu(x) \leq \varphi(x) \leq \varphi_\mu(x) + \mu D_2 \qquad \forall\, x \in \mathbf{E}_1. \tag{5.8}$$

Moreover, if we denote by $y_\mu(x)$ the (unique) maximizer from (5.7), then the gradient of $\varphi_\mu(x)$ is given by $\nabla\varphi_\mu(x) = A^* y_\mu(x)$ and is Lipschitz continuous with constant

$$\gamma_\mu = \frac{1}{\mu\sigma_2}\|A\|_{1,2}^2. \tag{5.9}$$

The smooth version of (P') therefore is

$$\boxed{\min_x \{\varphi_\mu(x) \ : \ x \in Q_1\}.} \tag{$P'_{\rm smooth}$}$$

The main result of [12] is the following:

THEOREM 15 (Nesterov [12, Theorem 3]). *If we apply Algorithm 6 to problem (P'_{smooth}) with smoothing parameter*

$$\mu = \frac{2\|A\|_{1,2}}{N+1}\sqrt{\frac{D_1}{\sigma_1\sigma_2 D_2}} \tag{5.10}$$

and if

$$x = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1, x_0, N),$$

then[3]

$$\varphi(x) - \varphi^* \leq \frac{4\|A\|_{1,2}}{N+1}\sqrt{\frac{D_1 D_2}{\sigma_1\sigma_2}}.$$

5.3. The main result. We will use the above theorem in the same way as Proposition 3 to devise a $O(1/\delta)$-algorithm for finding a δ-approximate solution of (P). Algorithms of this type, formulated for several specific choices of objective functions, were suggested already by Nesterov [13, 14]. These methods are similar in spirit to Algorithm 2, recursively updating an upper bound on φ^*. We give a single and faster algorithm applicable to the problems considered in the cited papers. Our contribution lies mainly in improving the theoretical complexity by incorporating a

[3]The original theorem states the result as a gap between $\varphi(x)$ and $\theta(y)$ for a certain $y \in Q_2$.

Algorithm 6 (Smooth) Efficient smooth method.

Input: $\psi, \gamma, Q_1, x_0, N$;
for $k = 0$ **to** N **do**
  Compute $\psi(x_k)$ and $\nabla\psi(x_k)$;
  $y_k = \arg\min\{\langle\nabla\psi(x_k), x - x_k\rangle + \frac{\gamma}{2}\|x - x_k\|_1^2 \ : \ x \in Q_1\}$;
  $z_k = \arg\min\{\sum_{i=0}^{k} \frac{i+1}{2}\langle\nabla\psi(x_i), x - x_i\rangle + \frac{\gamma}{\sigma_1}d_1(x) \ : \ x \in Q_1\}$;
  $x_{k+1} = \frac{2}{k+3}z_k + \frac{k+1}{k+3}y_k$;
end for
Output: y_N

bisection speedup. As in the previous section, it is possible to formulate a nonrestarting version of our algorithm by sacrificing the double logarithm in the theoretical complexity for a single one.

Preliminaries. Let us return to problem (P), using the representation (5.2) for the objective function (hence $Q_1 = \mathcal{L}$), and approach it with the tools described in the previous subsections. Let $\mathbf{E}_1 := \mathbf{E}$ and assume that $G\colon \mathbf{E}_1 \to \mathbf{E}_1^*$ defines an ellipsoidal rounding of $\partial\varphi(0) = A^*Q_2$ such that (3.1) holds with $\gamma_0 = 1$. Notice that the inequalities (3.5), (3.7) and (3.8) are implied by the former. To be able to obtain an algorithm guaranteeing a δ-approximate output in relative scale, the choice of the primal norm as the norm coming from the rounding is crucial:

$$\|x\|_1 := \|x\|_G \qquad \forall\, x \in \mathbf{E}_1.$$

If we wish to apply Algorithm 6, we need to supply it a *bounded* subset of $\mathcal{L}$ (which is unbounded) containing the minimizer. Observe that as long as $\varphi^* \le R$ for some positive number R, (3.8) guarantees that all minimizers of (P) lie in the set

$$Q_1(R) := \mathcal{L} \cap \{x \ : \ \|x - x_0\|_G \le R\},$$

where x_0—the projection of the origin onto $\mathcal{L}$ in the G-norm—is the center of $Q_1(R)$ as defined in (5.6) if we choose the prox-function for $Q_1(R)$ to be

$$d_1(x) := \tfrac{1}{2}\|x - x_0\|_G^2.$$

In this case $\sigma_1 = 1$ and $D_1 = \max\{d_1(x) \ : \ x \in Q_1(R)\} = \frac{1}{2}R^2$. We leave the choice of d_2 purposely open to allow for fine-tuning for particular applications.

A direct consequence of Theorem 15 with the settings described above is the following analogue of Lemma 5:

LEMMA 16. *If $\varphi^* \le R$, $\beta > 0$ and we set*

$$N = \left\lfloor \frac{2\sqrt{2}\|A\|_{1,2}}{\beta}\sqrt{\frac{D_2}{\sigma_2}} \right\rfloor$$

for some $\beta > 0$,

$$\mu = \frac{\sqrt{2}\|A\|_{1,2}R}{N+1}\sqrt{\frac{1}{\sigma_2 D_2}}$$

and γ_μ as in (5.9), then

$$x = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1(R), x_0, N)$$

satisfies

$$\varphi(x) \leq (1+\beta)R.$$

The above lemma leads to a bisection algorithm (Algorithm 7) in the same way as we have seen it in the section on subgradient algorithms. The main result follows:

Algorithm 7 (SmoothBis) Smoothed bisection scheme.

Input: $\varphi, \alpha, \beta, \delta, x_0$;
$k=0,\ \hat{x}_0 = x_0,\ L_0 = \|x_0\|_G,\ R_0 = \varphi(x_0)$;
$c = 2(1+\beta),\ N = \left\lfloor \frac{2\sqrt{2}\|A\|_{1,2}}{\beta}\sqrt{\frac{D_2}{\sigma_2}} \right\rfloor$;
while $R_k/L_k > c$ **do**
  $k = k+1$;
  $R = \sqrt{\frac{L_{k-1}R_{k-1}}{1+\beta}},\ \mu = \frac{\sqrt{2}\|A\|_{1,2}R}{N+1}\sqrt{\frac{1}{\sigma_2 D_2}},\ \gamma_\mu = \frac{\|A\|_{1,2}^2}{\mu\sigma_2}$;
  $x = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1(R), x_0, N)$;
  if $\varphi(x) \leq (1+\beta)R$ **then**
    $R_k = \varphi(x),\ L_k = L_{k-1},\ \hat{x}_k = x$;
  else
    $L_k = R$;
    if $\varphi(x) \leq R_{k-1}$ **then**
      $R_k = \varphi(x),\ \hat{x}_k = x$;
    else
      $R_k = R_{k-1},\ \hat{x}_k = \hat{x}_{k-1}$;
    end if
  end if
end while
$R = \varphi(\hat{x}_k)$;
$N = \left\lfloor 2\sqrt{2}c\|A\|_{1,2}(1+\frac{1}{\delta})\sqrt{\frac{D_2}{\sigma_2}} \right\rfloor,\ \mu = \frac{\sqrt{2}\|A\|_{1,2}R}{N+1}\sqrt{\frac{1}{\sigma_2 D_2}},\ \gamma_\mu = \frac{\|A\|_{1,2}^2}{\mu\sigma_2}$;
$\hat{x}_{k+1} = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1(R), x_0, N)$;
Output: $\hat{x}_{k+1}$

THEOREM 17. *Algorithm 7 returns a δ-approximate solution of (P) and takes at most*

$$\frac{2\sqrt{2}\|A\|_{1,2}}{\beta}\sqrt{\frac{D_2}{\sigma_2}}\left(\log_2\log_2\frac{1}{\alpha}\right) + 2\sqrt{2}(1+\beta)\|A\|_{1,2}\left(1+\frac{1}{\delta}\right)\sqrt{\frac{D_2}{\sigma_2}}$$

steps of the smooth optimization subroutine. If β is chosen to be a constant, then the number of steps is

$$O\left(\|A\|_{1,2}\sqrt{\frac{D_2}{\sigma_2}}\left(\frac{1}{\delta}+\ln\ln\frac{1}{\alpha}\right)\right). \tag{5.11}$$

5.4. A direct representation of the objective function. We can get rid of the dependence on $\|A\|_{1,2}$ in (5.11) by identifying $\mathbf{E}_2$ with $\mathbf{E}_1^*$ (and consequently $\mathbf{E}_1$ with $\mathbf{E}_2^*$). In this case we can simply choose $A = I$ and consider the following structural model for the objective function:

$$\varphi(x) = \max_g\{\langle g, x\rangle \ :\ g \in Q_2\}.$$

Let us set $\|g\|_2 = \|g\|_1^* = \|g\|_G^*$ and select the following prox-function for Q_2 (with center at the origin):

$$d_2(g) = \tfrac{1}{2}(\|g\|_G^*)^2.$$

Clearly $\sigma_2 = 1$ and $D_2 \leq \frac{1}{2\alpha^2}$—the second inequality follows from the ellipsoidal rounding inclusion (3.1) and the assumption $\gamma_0 = 1$. Also observe that since $\|\cdot\|_2^* \equiv \|\cdot\|_1$, we have

$$\|A\|_{1,2} = \max\{\|Ax\|_2^* \ : \ \|x\|_1 = 1\} = \max\{\|x\|_1 \ : \ \|x\|_1 = 1\} = 1.$$

Substituting for the values of these parameters into (5.11) gives the following guarantee:

$$O\left(\frac{1}{\alpha\delta} + \frac{1}{\alpha}\ln\ln\frac{1}{\alpha}\right).$$

Remark 1. Observe that, in principle, we do not lose generality by "excluding" A because we can simply set the "new" Q_2 to be equal to the "old" A^*Q_2. However, this sacrifice in modeling flexibility means that Q_2 always coincides with $\partial\varphi(0)$, which has to be of a simple structure for the algorithm to work efficiently. This is mainly due to the need to compute derivatives of φ_μ, which amounts to solving a concave quadratic maximization problem over Q_2 (5.7). If this problem can not be solved efficiently (say in a closed form), the method will likely be impractical.

6. Comparison of complexities. Table 1 compares the iteration complexities of the algorithms we have discussed in this paper. The per-iteration work of all methods is $O(mn)$.

TABLE 1
Summary of iteration complexities of all algorithms.

Method	Number of iterations
SubSearch	$O(\frac{1}{\alpha^2\delta^2}\ln\frac{1}{\alpha})$
SubBis	$O(\frac{1}{\alpha^2\delta^2} + \frac{1}{\alpha^2}\ln\ln\frac{1}{\alpha})$
SubSearchNR	$O(\frac{1}{\alpha^2\delta^2}\ln\frac{1}{\alpha})$
SubBisNR	$O(\frac{1}{\alpha^2\delta^2} + \frac{1}{\alpha^2}\ln\frac{1}{\alpha})$
SmoothBis	$O(\frac{1}{\alpha\delta} + \frac{1}{\alpha}\ln\ln\frac{1}{\alpha})$

We will now briefly put the above results in perspective with interior-point methods (IPM). While IPMs, in theory, need only $O(\ln(1/\epsilon))$ iterations to find a point within the (absolute) error ϵ of the optimum, each iteration is considerably more expensive because of the need to work with second-order information. In this sense, the fastest methods presented in this paper are promising for problems where the desired accuracy is not too high, and the dimension of the problem is huge so that performing even a single iteration of an IPM is impossible.

7. Applications. In this section we apply the fastest of the algorithms developed in this paper—the bisection algorithm based on smoothing SmoothBis —to several problems of the form (P).

7.1. Minimizing the maximum of absolute values of linear functions. In this subsection we consider problem (P) with the objective function from Example 3:

$$\min\{\varphi_\infty(x) \ : \ x \in \mathcal{L}\}. \tag{7.1}$$

Many seemingly unrelated problems can be reformulated in the above form. For example, by (7.1) one can model

- the truss topology design problem,
- the problem of the construction of a c-optimal statistical design, and
- the problem of finding a solution of an underdetermined linear system with the smallest ℓ_1 norm.

In all the examples above the feasible set $\mathcal{L}$ is one-dimensional. We will now show how one can solve problem 7.1 using the results of Section 5. A different approach for solving the problems above, simultaneously and in relative scale, was recently proposed by Richtárik [16, 18]. The iteration complexity is also $O(1/\delta)$, but the approach uses very different techniques.

Applying the algorithm. We will work with the last of the three representations for the objective function from Example 3:

$$\varphi_\infty(x) = \max\{|\langle a_i, x\rangle| \ : \ i = 1, 2, \ldots, m\} = \max_y\{\langle Ax, y\rangle \ : \ y \in Q_2\},$$

with Q_2 being the unit simplex in $\mathbf{R}^{2m}$ and A the $2m \times n$ matrix with rows $a_i, -a_i, \ i = 1, \ldots, m$. In addition, assume that the vectors $a_i, \ i = 1, 2, \ldots, m$, span $\mathbf{E}_1^* = \mathbf{R}^n$. It seems natural to choose $\|y\|_2 := \sum_i |y^{(i)}|$ so that $\|y\|_2 = 1$ for all $y \in Q_2$. If we let

$$d_2(y) := \ln 2m + \sum_{i=1}^{2m} y^{(i)} \ln y^{(i)}$$

and define $0 \times \ln 0 := \lim_{\tau \downarrow 0} \tau \ln \tau = 0$, then by the following lemma, d_2 is a prox-function on Q_2 with center $y_0 := (\frac{1}{2m}, \ldots, \frac{1}{2m})$.

LEMMA 18. *d_2 is strongly convex on Q_2, with respect to $\|\cdot\|_2$, with convexity parameter $\sigma_2 = 1$.*

Proof. It suffices to show that $d_2(y) \geq \frac{1}{2}\|y - y_0\|_2^2$. This can be proved by elementary means using only the Cauchy-Schwarz inequality (see, eg. Borwein and Lewis [3, Exercise 3.3.25(d)]) or, using differentiation and a certain knowledge about convex functions (Nesterov [12, Lemma 3]). □

It is easy to see that $D_2 = \sup\{d_2(y) \ : \ y \in Q_2\} = \ln 2m$ (the supremum is attained at each of the boundary vertices). Finally, let us compute the norm of the linear operator A:

$$\begin{aligned} \|A\|_{1,2} &= \max\{\|Ax\|_2^* \ : \ \|x\|_1 = 1\} \\ &= \max\{\|Ax\|_\infty \ : \ \|x\|_G = 1\} \\ &= \max\{\varphi(x) \ : \ \|x\|_G = 1\} = \tfrac{1}{\alpha}. \end{aligned}$$

The last step follows from inequality (3.5) in view of our assumption that $\gamma_0 = 1$. It is shown in Lemma 4 of Nesterov [12] that the smooth approximation of φ is given by

$$\varphi_\mu(x) = \mu \ln\left(\frac{1}{2m}\sum_{i=1}^{m}\left[e^{\langle a_i, x\rangle/\mu} + e^{\langle -a_i, x\rangle/\mu}\right]\right).$$

Since $\partial\varphi(0) = \operatorname{conv}\{\pm a_i, i = 1, 2, \ldots, m\}$ is a centrally symmetric subset of $\mathbf{R}^n$, we may assume that a good rounding, with $\frac{1}{\alpha} = O(\sqrt{n})$, is available to us. It can be computed efficiently, in $O(n^2 m \ln m)$ arithmetic operations. For details about algorithms we refer to [1, 9, 13, 21, 22].

Complexity. The performance of Algorithm 7 for this problem then by substituting into (5.11) is

$$O\left(\sqrt{n \ln m}\left(\ln\ln n + \frac{1}{\delta}\right)\right).$$

This improves on the result of Nesterov [13], where the author gives the bound

$$O\left(\frac{\sqrt{n \ln m}}{\delta} \ln n\right).$$

7.2. Minimizing the sum of absolute values of linear functions. Consider problem (P) with the following objective function:

$$\varphi_1(x) = \sum_{i=1}^{m} |\langle a_i, x\rangle|.$$

As usual, we assume that the vectors $a_1, a_2, \ldots, a_m$ span $\mathbf{E}_1^*$.

Applying the algorithm. Let $\mathbf{E}_1 = \mathbf{E}_1^* = \mathbf{R}^n$ and $\mathbf{E}_2 = \mathbf{E}_2^* = \mathbf{R}^m$ and let us represent φ_1 as

$$\varphi_1(x) = \max_y \{\langle Ax, y\rangle \ : \ y \in Q_2\}, \tag{7.2}$$

where $Q_2 = \{y \in \mathbf{R}^m \ : \ |y^{(i)}| \le 1, \ i = 1, 2, \ldots, m\}$ and A is the $m \times n$ matrix with rows $a_1, \ldots, a_m$. Usually we first find a rounding of $\partial\varphi_1(0)$ and using the rounding operator define a norm on $\mathbf{E}_1$. Because of the simple structure of Q_2, we will instead start by defining $\|y\|_2 := (\sum_i (y^{(i)})^2)^{1/2}$ and noting that this leads to a $\sqrt{m}$-rounding of Q_2:

$$\mathcal{B}(I, 1) \subseteq Q_2 \subseteq \mathcal{B}(I, \sqrt{m}), \tag{7.3}$$

with $I\colon \mathbf{R}^m \to \mathbf{R}^m$ denoting the identity operator. We will show now how this naturally leads to a rounding operator defined on $\mathbf{E}_1$ enjoying the same quality of rounding.

LEMMA 19 (Nesterov [14, Lemma 2]). *If the vectors $a_1, \ldots, a_m$ span $\mathbf{R}^m$, then $\|x\|_1 := \|Ax\|_2^*$ defines a norm on $\mathbf{R}^n$. Moreover, if we let $G := A^T A$ (a positive definite matrix), then $\|\cdot\|_1 \equiv \|\cdot\|_G$ and*

$$\mathcal{B}(G, 1) \subseteq \partial\varphi(0) = A^T Q_2 \subseteq \mathcal{B}(G, \sqrt{m}).$$

Proof. Note that $\|x\|_1 = \|Ax\|_2^* = \langle Ax, Ax\rangle^{1/2} = \langle Gx, x\rangle^{1/2} = \|x\|_G$. The equality $\partial\varphi(0) = A^T Q_2$ follows from (7.2). In view of (7.3) we obtain

$$\varphi(x) = \max_{y \in Q_2} \langle Ax, y\rangle \le \max_{y \in \mathcal{B}(I, \sqrt{m})} \langle Ax, y\rangle = \max_{\|y\|_2 \le \sqrt{m}} \langle Ax, y\rangle = \sqrt{m}\|Ax\|_2^* = \sqrt{m}\|x\|_1$$

and

$$\varphi(x) = \max_{y \in Q_2} \langle Ax, y\rangle \ge \max_{y \in \mathcal{B}(I, 1)} \langle Ax, y\rangle = \max_{\|y\|_2 \le 1} \langle Ax, y\rangle = \|Ax\|_2^* = \|x\|_1. \qquad \square$$

Let us define $d_2(y) := \frac{1}{2}\|y\|_2^2$, so that the convexity parameter of this prox-function is $\sigma_2 = 1$. It follows from (7.3) that $D_2 = \max\{d_2(y) \ : \ y \in Q_2\} \leq \frac{1}{2}m$. Finally,

$$\|A\|_{1,2} = \max\{\|Ax\|_2^* \ : \ \|x\|_1 = 1\} = \max\{\|x\|_1 \ : \ \|x\|_1 = 1\} = 1.$$

Complexity. The performance of Algorithm 7 on this problem then by substituting into (5.11) is

$$O\left(\sqrt{m}\left(\frac{1}{\delta} + \ln\ln m\right)\right).$$

This improves on the following bound of Nesterov [14]

$$O\left(\frac{\sqrt{m}\ln m}{\delta}\right).$$

7.3. Minimizing the maximum of linear functions over a simplex. The motivation for this problem is the computation of the value of a two-person zero-sum matrix game with nonnegative coefficients: Let $\hat{A} \in \mathbf{R}^{m\times n}$ be a real matrix with nonnegative entries and rows $a_1, \ldots, a_m$. Consider the following game. There are two players: a row player (R) and a column player (C). Player R chooses a probability distribution y over the rows of matrix $\hat{A}$ and C chooses a probability distribution x over the columns. After that, C pays $y^T\hat{A}x$ dollars to R. Assume the players are *conservative*, that is, C wishes to minimize his worst-case loss and R wants to maximize his worst-case win. That is, C prefers to choose strategy

$$x^* \in \arg\min_{x\in\Delta_n}\max_{y\in\Delta_m} y^T\hat{A}x,$$

and similarly, R wishes to choose strategy

$$y^* \in \arg\max_{y\in\Delta_m}\min_{x\in\Delta_n} y^T\hat{A}x.$$

The set Δ_n (resp. Δ_m) denotes the unit simplex in $\mathbf{R}^n$ (resp. $\mathbf{R}^m$). A classical result by von Neumann [23] says that[4]

$$\varphi^* := \min_{x\in\Delta_n}\max_{y\in\Delta_m} y^T\hat{A}x = \max_{y\in\Delta_m}\min_{x\in\Delta_n} y^T\hat{A}x.$$

The value φ^* is called the *value of the game*. Note that if we let $Q_1 := \Delta_n$ and

$$\varphi(x) = \max\{\langle a_i, x\rangle \ : \ i = 1, 2, \ldots, m\},$$

then we can write $\varphi^* = \min_x\{\varphi(x) \ : \ x \in Q_1\}$.

Applying the algorithm. First observe that

$$\partial\varphi(0) = \mathrm{conv}\{a_i \ : \ i = 1, 2, \ldots, m\},$$

which fails to satisfy (2.1) due to the assumption on nonnegativity of the entries of $\hat{A}$. To remedy this situation, we will follow a trick suggested in Nesterov [13]. Notice

[4]For a modern proof based on Fenchel duality, we refer to, for example, Exercise 4.2.16 in Borwein and Lewis [3].

that we are interested in φ as defined on Δ_n only, which is a subset of the nonnegative orthant. Let us therefore define

$$\hat{\varphi}(x) \stackrel{\text{def}}{=} \max\{\langle a_i, |x|\rangle \ : \ i = 1, 2, \ldots, m\},$$

where $|x| = (|x_1|, \ldots, |x_n|)$ and observe that $\hat{\varphi}(x) = \varphi(x)$ for all $x \in \mathbf{R}^n_+$, and

$$\partial\hat{\varphi}(0) = \operatorname{conv} \bigcup_{i=1}^{m} \{g \ : \ -a_i \leq g \leq a_i\}.$$

It is particularly interesting to note that $\partial\hat{\varphi}(0)$ is a *sign-invariant* set, one that with every point g contains all points obtained by arbitrarily changing the signs of the coordinates of g. In fact, $\partial\hat{\varphi}(0)$ is the smallest sign-invariant set containing $\partial\varphi(0)$. Nesterov shows that sign-invariant convex bodies admit a more efficient rounding algorithm than the more general central-symmetric sets mainly due to the possibility of working only with *diagonal* positive definite matrices defining the rounding.

Instead of rounding $\partial\varphi(0)$ one can therefore find an ellipsoidal rounding of $\partial\hat{\varphi}(0)$ (defined by a diagonal positive definite matrix G) with $\frac{1}{\alpha} = O(\sqrt{n})$ and then deduce inequality (3.5), which holds for all $x \in \mathbf{R}^n_+$ (Nesterov [13, Lemma 5]). Smoothing of φ (and hence of $\hat{\varphi}$ on the domain of interest) can be performed in complete analogy with the situation in Subsection 7.1. The choice of the representation of the objective function, the choice of the prox-function for Q_2 and the implied bounds are all identical (the only change is that the dimension drops from $2m$ to m).

Complexity. The complexity guarantee of Algorithm 3 as applied to the problem of computing the value of a two-person matrix game with nonnegative coefficients is:

$$O\left(\sqrt{n \ln m}\left(\frac{1}{\delta} + \ln\ln n\right)\right).$$

This improves on the result in Nesterov [13, Algorithm 4.4], where the author gives the bound

$$O\left(\frac{\sqrt{n \ln m}}{\delta} \ln n\right).$$

Acknowledgments. The author is very grateful to Mike Todd for numerous enlightening discussions and encouragement to publish these results.

REFERENCES

[1] D. Ahipaşaoğlu, P. Sun, and M. J. Todd, *Linear convergence of a modified Frank-Wolfe algorithm for computing minimum-volume enclosing ellipsoids*, Tech. Report TR1452, Cornell University, School of Operations Research and Information Engineering, 2006.

[2] Alexandre Belloni and Robert M. Freund, *On the symmetry function of a convex set*, Math. Program., 111 (2007), pp. 57–93.

[3] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization*, Advanced Books in Mathematics, Canadian Mathematical Society, 2000.

[4] F. A. Chudak and V. Eleutério, *Improved approximation schemes for linear programming relaxations of combinatorial optimization problems.*, in IPCO'05, Berlin, 2005.

[5] J.-L. Goffin, *On convergence rates of subgradient optimization methods*, Mathematical Programming, 13 (1977), pp. 329–347.

[6] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[7] F. John, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays, Presented to R. Courant on his 60th Birthday January 8, 1948, New York, 1948, Wiley Interscience, pp. 187–204.
[8] L. G. Khachiyan, *Rounding of polytopes in the real number model of computation*, Mathematics of Operations Research, 21 (1996), pp. 307–320.
[9] P. Kumar and E. A. Yildirim, *Minimum volume enclosing ellipsoids and core sets*, Journal of Optimization Theory and Applications, 126 (2005), pp. 1–21.
[10] Yu. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence* $O(\frac{1}{k^2})$., Doklady AN SSSR (translated as Soviet. Math. Docl.), 269 (1983), pp. 543–547.
[11] ———, *Introductory Lectures on Convex Optimization. A Basic Course*, vol. 87 of Applied Optimization, Kluwer Academic Publishers, Boston, 2004.
[12] ———, *Smooth minimization of non-smooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.
[13] ———, *Rounding of convex sets and efficient gradient methods for linear programming problems*, CORE Discussion Paper #2004/04, (January 2004).
[14] ———, *Unconstrained convex minimization in relative scale*, CORE Discussion Paper #2003/96, (November 2003).
[15] ———, *Barrier subgradient method*, CORE Discussion Paper #2008/60, (October 2008).
[16] P. Richtárik, *Some Algorithms for Large-Scale Linear and Convex Minimization in Relative Scale*, PhD thesis, Cornell University, School of Operations Research and Information Engineering, August 2007.
[17] ———, *Approximate level method*, CORE Discussion Paper #2008/83, (December 2008).
[18] ———, *Simultaneously solving seven optimization problems in relative scale*, Manuscript, (December 2008).
[19] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1997. Reprint of the 1970 original, Princeton Paperbacks.
[20] N. Z. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, 1985.
[21] P. Sun and R. M. Freund, *Computation of minimum-volume covering ellipsoids*, Oper. Res., 52 (2004), pp. 690–706.
[22] M. J. Todd and E. A. Yildirim, *On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids*, Tech. Report TR1435, Cornell University, School of Operations Research and Information Engineering, 2005.
[23] J. von Neumann and O. Morgenstern, *The Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, USA, 1948.