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, care is needed when choosing the size of the artificially imposed bounds: large feasible sets tend to slow algorithms down, whereas small sets may lead to the exclusion of minimizers. 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 [15,16,18,19]. This contrasts with the enormous literature on combinatorial optimization where approximation algorithms are studied extensively.

In particular, Nesterov [15] showed that the above obstacles can be overcome when minimizing convex *homogeneous* functions over an affine subspace. The essence of his approach involves computing an *ellipsoidal rounding* of the subdifferential of the objective function at the origin. 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 [14,15] 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 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

*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 [17, 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).

and Eleutério [5] 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 a "nonrestarting" behavior. In Section 5 we propose more efficient methods, which are grounded in Nesterov's smoothing technique. These are of an order of magnitude faster than those based on the subgradient routine. Next follows Section 6 in which we briefly summarize the theoretical complexities and remark on the scaling invariance of the methods. In Section 7 we describe several special cases to which the methods of this paper apply. The final section is devoted to computational experiments.

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 [20]. A 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 [7]. For a more compact and up-to-date treatment see Borwein and Lewis [4, 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. *A 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 equivalent inequality $\varphi(x) - \varphi^* \leq \frac{\delta}{1+\delta}\varphi(x)$.

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

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

The function φ is clearly linear on every feasible ray leaving from the origin. In fact, it can be shown that φ is convex on its domain (Hiriart-Urruty and Lemaréchal [7, 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

$$\varphi(x) = \tau\phi(y/\tau) = \tau \max_{1\leq i\leq m} |\langle a_i, y/\tau\rangle + b^{(i)}| = \max_{1\leq i\leq m} |\langle a_i, y\rangle + b^{(i)}\tau| = \max_{1\leq i\leq m} |\langle a_i', x\rangle|,$$

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 [15] 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 using Khachiyan's algorithm [9], or the recent method of Ahipaşaoğlu, Sun and Todd [1]. We thus assume that G and $\rho \geq 1$ are available such that

$$\mathcal{B}(G,1) \subseteq \partial\varphi(0) \subseteq \mathcal{B}(G,\rho), \tag{3.1}$$

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 iteration complexities of the algorithms of this paper depend on the parameter ρ characterizing the quality of the ellipsoidal rounding (3.1). The following result, a celebrated theorem of John [8], gives lower bounds on the quality of rounding admitted by full-dimensional convex sets.

PROPOSITION 2 (John [8]). *Any convex body $Q \subset \mathbf{E}^*$ admits a rounding by concentric ellipsoids with $\rho \leq \dim \mathbf{E}^*$. If Q is centrally symmetric, then there exists a rounding with $\rho \leq \sqrt{\dim \mathbf{E}^*}$.*

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
(i) a regular n-simplex has quality $\rho = n$,
(ii) the n-cube has quality $\rho = \sqrt{n}$.

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, of radii 1 and ρ, 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. 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

$$\|x\|_G \leq \varphi(x) \leq \rho\|x\|_G, \quad x \in \mathbf{E}, \tag{3.5}$$

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

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

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

$$\tfrac{\varphi(x_0)}{\rho} \leq \|x_0\|_G \leq \|x^*\|_G \leq \varphi^* \leq \varphi(x_0) \leq \rho\|x_0\|_G. \tag{3.6}$$

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

$$\|x^* - x_0\|_G < \|x^*\|_G \leq \varphi^* \leq \varphi(x), \quad x \in \mathcal{L}. \tag{3.7}$$

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 we refer the reader to Shor [21] and Goffin [6]. 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 [12, Section 3.2.3].

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,
- $\gamma = \rho$ 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, for example, Nesterov [12, 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:*

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

Algorithm 1 (Subgrad) Constant step-length subgradient scheme

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

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

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 an upper bound R on $\|x^* - x_0\|$.

The particular choice of x_0 as the projection of the origin onto the feasible set of (P) makes sense for at least two reasons. First, notice that if the ellipsoidal rounding of $\partial\varphi(0)$ is perfectly tight ($\rho = 1$), 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 by (3.6),

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

and hence x_0 is a $(\rho-1)$-approximate solution of (P). The better the rounding, the better the approximation factor. Second, (3.7) offers the readily available upper bound $R = \varphi(x_0)$. Of course, φ^* would be better; the issue is that it is not known.

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

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

Available but bad upper bound. Since the previous upper bound is unknown, it seems reasonable to instead use the worse (but available) bound $R = \varphi(x_0)$. If we wish to guarantee a solution within relative error δ, we need to take $N = \lfloor \rho^4/\delta^2 \rfloor$ iterations. The argument is exactly the same and uses (4.2).

Iteratively updated upper bound. To move towards the better of the two extremes, Nesterov [15] proposed a scheme (Algorithm 2) which uses the subgradient method as a subroutine, iteratively decreasing the known upper bound. This algorithm starts by running the subgradient method for $O(\rho^2/\delta^2)$ iterations with the available upper bound $\varphi(x_0)$. In the case when the subgradient subroutine is doing well and manages to decrease the objective value by a constant factor, the previously available upper bound also decreases by the same factor. This improved bound is then used to run the next subgradient subroutine, again starting from x_0.

Algorithm 2 (SubSearch) Subgradient search scheme.

Input: $\varphi, \mathcal{L}, x_0, \rho, \beta > 0, \delta$;
$\hat{x}_0 = x_0,\ c = e^{\beta},\ k = 1$;
$N = \left\lfloor c^2\rho^2\left(1+\frac{1}{\delta}\right)^2 \right\rfloor$;
$\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, \varphi(\hat{x}_{k-1}), N)$;
while $\varphi(\hat{x}_k) < \varphi(\hat{x}_{k-1})/c$ **do**
  $k = k+1$;
  $\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, \varphi(\hat{x}_{k-1}), N)$;
end while
Output: $\hat{x}_k$

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. However, it underperforms the one-time application of the subgradient method with the good but unknown upper bound, by a factor of $O(\ln\rho)$. The performance of the method, as analyzed by Nesterov [15], 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 [15, Theorem 3]). *Algorithm 2 returns a δ-approximate solution of (P) and takes at most*

$$e^{2\beta}\rho^2\left(1+\frac{1}{\delta}\right)^2\left(1+\frac{1}{\beta}\ln\rho\right) \tag{4.3}$$

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

$$O\left(\tfrac{\rho^2}{\delta^2}\ln\rho\right). \tag{4.4}$$

The optimal choice is $\beta = \frac{1}{2}(\sqrt{t^2+2t}-t) \approx \frac{1}{2}$, with $t = \ln\rho$.

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

$$\tfrac{\varphi(x_0)}{\rho} \le \varphi^* \le \varphi(\hat{x}_{k-1}) < \tfrac{\varphi(x_0)}{e^{\beta(k-1)}},$$

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 $1+\beta^{-1}\ln\rho$. The bound (4.3) is obtained by multiplying this by N from Step 3 of the algorithm. Minimizing (4.3) in β gives the final statement. 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

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

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 improve the theoretical performance of this algorithm (the same improvement was independently obtained by Chudak and Eleutério [5] in the context of combinatorial applications). The key observation is formulated in the following lemma.

LEMMA 5. *If $\varphi^* \le R$ and $N = \lfloor \rho^2/\beta^2 \rfloor$ for some $\beta > 0$, then*

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

satisfies

$$\varphi(x) - \beta R \le \varphi^* \quad \textit{and} \quad \varphi(x) \le (1+\beta)R. \tag{4.5}$$

Proof. By Proposition 3 we have $\varphi(x) - \varphi^* \le \rho R/\sqrt{N+1} \le \beta R$, and hence $\varphi(x) \le \varphi^* + \beta R \le (1+\beta)R$. $\square$

The above result essentially states that for *any* positive R we can, at the cost of $O(\rho^2/\beta^2)$ iterations of the subgradient method (Algorithm 1), either get a certificate that $\varphi^* \le (1+\beta)R$ (if x satisfies $\varphi(x) \le (1+\beta)R$) or that $R < \varphi^*$ (if $\varphi(x) > (1+\beta)R$). In any case, we either get an upper *or* lower bound on φ^*.

Note that thanks to (3.6), we are in the possession of an initial lower and upper bound on φ^*: if we set $L_0 = \|x_0\|_G$ and $R_0 = \varphi(x_0)$, then $\frac{\varphi(x_0)}{\rho} \le L_0 \le \varphi^* \le R_0$, with

$$\tfrac{R_0}{L_0} \le \rho. \tag{4.6}$$

Assume that at step k we have $L_k \le \varphi^* \le R_k$, with $q_k \stackrel{\text{def}}{=} R_k/L_k > 1+\beta$ (see Figure 1). Pick R so that

$$L_k < R < (1+\beta)R < R_k. \tag{4.7}$$

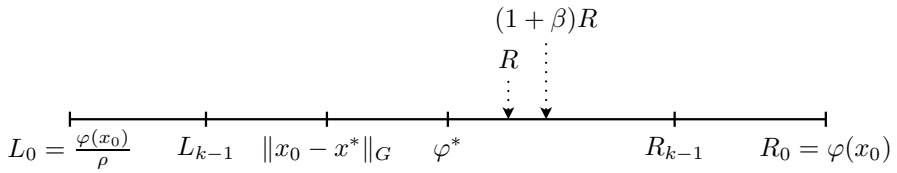

FIG. 1. *Bisection step k.*

For this R let x be given by Lemma 5. There are two possibilities. If $\varphi(x) \leq (1+\beta)R$, then in view of (4.5) we can update:

$$L_{k+1} = \max\{\varphi(x) - \beta R, L_k\}, \qquad R_{k+1} = \min\{R_k, \varphi(x)\} \leq (1+\beta)R. \tag{4.8}$$

If $\varphi(x) > (1+\beta)R$, then we can set

$$L_{k+1} = R, \qquad R_{k+1} = \min\{R_k, \varphi(x)\}. \tag{4.9}$$

This bisection procedure is then repeated until $q_k < (1+\tau)(1+\beta)$ for some $\tau > 0$. The following lemma states how much improvement in q_k can be obtained by a single bisection step.

LEMMA 6. *Assume $L_k \leq \varphi^* \leq R_k$, $q_k > 1+\beta$ and $\beta > 0$. After a single bisection step with $R = [L_k R_k/(1+\beta)]^{1/2}$ we obtain $L_{k+1} \leq \varphi^* \leq R_{k+1}$ satisfying*

$$q_{k+1} \leq (1+\beta)^{1/2} q_k^{1/2}. \tag{4.10}$$

Proof. It is easy to see that (4.7) holds. Observing that R is chosen so that $(1+\beta)R/L_k = R_k/R$, in view of (4.8) and (4.9) we obtain

$$q_{k+1} = \tfrac{R_{k+1}}{L_{k+1}} \leq \max\left\{\tfrac{(1+\beta)R}{L_k}, \tfrac{\min\{R_k,\varphi(x)\}}{R}\right\} \leq \tfrac{R_k}{R} = (1+\beta)^{1/2} q_k^{1/2}. \qquad \square$$

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

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

$$\tfrac{\rho^2}{\beta^2}\left[1 + \log_2\left(\tfrac{\ln \rho}{\ln(1+\tau)}\right)\right] + (1+\tau)^2(1+\beta)^2\rho^2\left(1+\tfrac{1}{\delta}\right)^2 \tag{4.11}$$

steps of the subgradient subroutine. If β is a constant, the number of steps is

$$O\left(\rho^2\left(\tfrac{1}{\delta^2} + \ln\ln\rho\right)\right). \tag{4.12}$$

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

$$q_k \leq (1+\beta)^{\frac{1}{2}} q_{k-1}^{\frac{1}{2}} \leq (1+\beta)^{\frac{1}{2}}(1+\beta)^{\frac{1}{4}} \cdots (1+\beta)^{\frac{1}{2^k}} q_0^{\frac{1}{2^k}} \overset{(4.6)}{\leq} (1+\beta)\rho^{\frac{1}{2^k}}.$$

The smallest integer k for which $(1+\beta)\rho^{\frac{1}{2^k}} \leq (1+\tau)(1+\beta)$ is $k^* = \lceil \log_2(\ln\rho/\ln(1+\tau))\rceil$ and hence the total number of lower-level subgradient method iterations of the bisection phase is at most $N_{\text{bis}} = \rho^2\beta^{-2}k^*$. The statement then follows by adding N_{bis}

Algorithm 3 (SubBis) Subgradient bisection scheme.

1: **Input:** $\varphi, \mathcal{L}, x_0, \rho, \beta, \tau, \delta$;
2: $k = 0,\ L_0 = \|x_0\|_G,\ R_0 = \varphi(x_0),\ c = (1+\tau)(1+\beta),\ N = \lfloor \rho^2/\beta^2 \rfloor$;
3: **while** $R_k/L_k > c$ **do**
4: $\quad R = \sqrt{\frac{L_k R_k}{1+\beta}},\ x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, R, N)$;
5: $\quad$ **if** $\varphi(x) \le (1+\beta)R$ **then**
6: $\quad\quad$ set R_{k+1}, L_{k+1} as in (4.8)
7: $\quad$ **else**
8: $\quad\quad$ set R_{k+1}, L_{k+1} as in (4.9)
9: $\quad$ **end if**
10: $\quad k = k+1$
11: **end while**
12: $N = \left\lfloor \frac{R_k^2}{L_k^2}\rho^2 \left(1+\frac{1}{\delta}\right)^2 \right\rfloor,\ x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x_0, R_k, N)$;
13: **Output:** x

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. Notice that $L_k \le \varphi(x)$ and Proposition 3 applied to the subgradient method call at Step 12 give

$$\varphi(x) - \varphi^* \le \frac{\rho R_k}{\sqrt{N+1}} \le \frac{\rho \frac{R_k}{L_k}\varphi(x)}{\sqrt{N+1}} \le \frac{\delta}{1+\delta}\varphi(x). \qquad \square$$

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 feasible set. This point is indeed special as it allows for the key inequalities (3.6) and (3.7), 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^*. 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^* is both the optimal value of (P) *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. Indeed, for any $x \in \mathcal{L}$,

$$\|x - x^*\|_G \leq \|x\|_G + \|x^*\|_G \overset{(3.7)}{\leq} \|x\|_G + \varphi^* \leq \|x\|_G + \varphi(x) \overset{(3.5)}{\leq} 2\varphi(x). \tag{4.13}$$

This means that whenever the subgradient method outputs some point x, we have an upper bound on $\|x - x^*\|_G$ available. Therefore, on the next call we can run the method starting at x with $R = \|x\|_G + \varphi(x)$.

Nonrestarting version of SubSearch. Algorithm 4 is a modified version of Algorithm 2 in the spirit of the preceding discussion. The theoretical performance is unchanged.

Algorithm 4 (SubSearchNR) Nonrestarting subgradient search scheme.

Input: $\varphi, \mathcal{L}, x_0, \rho, \delta$;
$\hat{x}_0 = x_0,\ c = \sqrt{e},\ k = 1$;
$N = \left\lfloor c^2\rho^2\left(1+\frac{1}{\delta}\right)^2 \right\rfloor,\ N' = \left\lfloor 4c^2\rho^2\left(1+\frac{1}{\delta}\right)^2 \right\rfloor,\ R = \varphi(\hat{x}_0)$;
$\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_0, R, N)$;
while $\varphi(\hat{x}_k) < \varphi(\hat{x}_{k-1})/c$ **do**
  $k = k + 1$;
  $R = \|\hat{x}_{k-1}\|_G + \varphi(\hat{x}_{k-1})$;
  $\hat{x}_k = \mathsf{Subgrad}(\varphi, \mathcal{L}, \hat{x}_{k-1}, R, N')$;
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 $1 + 2\ln\rho$ and the total number of lower-level subgradient steps is hence at most*

$$4e\rho^2\left(1+\tfrac{1}{\delta}\right)^2(1 + 2\ln\rho) = O\left(\tfrac{\rho^2}{\delta^2}\ln\rho\right). \tag{4.14}$$

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). For $k > 1$, the analysis is analogous:

$$\varphi(\hat{x}_k) - \varphi^* \leq \tfrac{\rho R}{\sqrt{N+1}} \overset{(4.13)}{\leq} \tfrac{\rho 2\varphi(\hat{x}_{k-1})}{2c\rho\left(1+\frac{1}{\delta}\right)} \leq \tfrac{\delta}{1+\delta}\varphi(\hat{x}_k). \qquad \square$$

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 $x' \in \mathcal{L}$ and assume $\varphi^* \leq R$. If we let $N = \lfloor \rho^2/\beta^2 \rfloor$ for some $\beta > 0$, then*

$$x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x', R + \|x'\|_G, N)$$

satisfies

$$\varphi(x) - \beta(\|x'\|_G + R) \leq \varphi^* \quad \text{and} \quad \varphi(x) \leq (1+\beta)R + \beta\|x'\|_G. \tag{4.15}$$

Proof. By (4.13) we have $\|x' - x^*\|_G \leq \|x'\|_G + R$ and hence by Proposition 3,

$$\varphi(x) - \varphi^* \leq \rho\tfrac{\|x'\|_G + R}{\sqrt{N+1}} \leq \beta(\|x'\|_G + R).$$

Rearranging the expression gives the first inequality in (4.15); the second inequality follows from (3.5). ∎

The idea of updating the lower and upper bounds is analogous to the restarting version of the algorithm. Assume that at step k we have $L_k \le \varphi^* \le R_k$, with $q_k = R_k/L_k > 1+\beta$. Pick R so that (4.7) holds and x' such that $R_k = \varphi(x')$, and let x be given by Lemma 9. Again, we have two possibilities. Notice that if $\varphi(x) \le (1+\beta)R + \beta\|x'\|_G$, then since $\|x'\|_G \le \varphi(x') = R_k$, we have

$$\varphi(x) \le (1+\beta)R + \beta R_k \overset{(4.7)}{\le} R_k, \tag{4.16}$$

as long as $\beta \le 1$. We can thus update

$$L_{k+1} = \max\{\varphi(x) - \beta(\|x'\|_G + R), L_k\}, \quad R_{k+1} = \varphi(x). \tag{4.17}$$

If $\varphi(x) > (1+\beta)R + \beta\|x'\|_G$, we can set

$$L_{k+1} = R, \qquad R_{k+1} = \min\{R_k, \varphi(x)\}. \tag{4.18}$$

The improvement in q_k after a single bisection step is given in the following result.

LEMMA 10. *Assume $L_k \le \varphi^* \le R_k$, $q_k > 1+\beta$ and $0 < \beta \le 1$. After a single bisection step with $R = [L_k R_k/(1+\beta)]^{1/2}$ we obtain $L_{k+1} \le \varphi^* \le R_{k+1}$ satisfying*

$$q_k \le \left(\beta + \tfrac{1}{\sqrt{2}}\right) q_{k-1}. \tag{4.19}$$

Proof. Note that (4.7) holds. In view of (4.17), (4.16) and (4.18) we get

$$\begin{aligned} q_{k+1} = \tfrac{R_{k+1}}{L_{k+1}} &\le \max\left\{\tfrac{(1+\beta)R+\beta R_k}{L_k}, \tfrac{\min\{R_k,\varphi(x)\}}{R}\right\} \\ &\le \tfrac{(1+\beta)R+\beta R_k}{L_k} \\ &= \sqrt{1+\beta}\sqrt{q_k} + \beta q_k = \sqrt{\tfrac{1+\beta}{q_k}} q_k + \beta q_k < \sqrt{\tfrac{1}{2}} q_k + \beta q_k. \end{aligned}$$ ∎

THEOREM 11. *Algorithm 5 run with β satisfying $0 < \beta < 1 - \frac{1}{\sqrt{2}}$ returns a δ-approximate solution of (P) and takes at most*

$$\tfrac{\rho^2}{\beta^2}\left[1 + \tfrac{\ln\rho - \ln[(1+\tau)(1+\beta)]}{-\ln(\beta+1/\sqrt{2})}\right] + 4(1+\tau)^2(1+\beta)^2\rho^2\left(1+\tfrac{1}{\delta}\right)^2 \tag{4.20}$$

steps of the subgradient subroutine. If τ and β are chosen as constants, this becomes

$$O\left(\tfrac{\rho^2}{\delta^2} + \rho^2 \ln\rho\right). \tag{4.21}$$

Proof. Let us first analyze the bisection phase. Repeated use of Lemma 10 gives $q_k \le (\beta + 1/\sqrt{2})^k q_0 \le (\beta + 1/\sqrt{2})^k \rho$. The smallest integer k for which the last expression drops below $c = (1+\tau)(1+\beta)$ is $k^* = \lceil \ln(\rho/c)/\ln(1/(\beta+1/\sqrt{2}))\rceil = O(\ln\rho)$, and hence the total number of lower-level subgradient method iterations of the bisection phase is $N_{\text{bis}} = \lfloor \rho^2/\beta^2 \rfloor k^*$. The guarantee (4.20) follows by adding N_{bis} and N from Step 13. The output of the algorithm is as specified; the analysis 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 (4.21) in comparison with (4.12). However, the bisection strategy separates the δ from the logarithmic term when compared to the bound (4.14) for the SubSearch algorithm.

Algorithm 5 (SubBisNR) Nonrestarting subgradient bisection scheme.

Input: $\varphi, \mathcal{L}, x_0, \rho, \beta, \tau, \delta$;
$k = 0,\ x' = x_0,\ L_0 = \|x_0\|_G,\ R_0 = \varphi(x_0)$;
$c = (1+\tau)(1+\beta),\ N = \left\lfloor \rho^2/\beta^2 \right\rfloor$;
while $R_k/L_k > c$ **do**
  $R = \sqrt{\frac{L_k R_k}{1+\beta}},\ x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x', \|x'\|_G + R, N)$;
  if $\varphi(x) \le (1+\beta)R + \beta\|x'\|_G$ **then**
    set L_{k+1}, R_{k+1} as in (4.17)
  else
    set L_{k+1}, R_{k+1} as in (4.18)
  end if
  $x' = x,\ k = k+1$
end while
$N = \left\lfloor 4\frac{R_k^2}{L_k^2}\rho^2\left(1+\frac{1}{\delta}\right)^2 \right\rfloor,\ x = \mathsf{Subgrad}(\varphi, \mathcal{L}, x', \|x'\|_G + R_k, N)$;
Output: x

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 $O(\rho^2\delta^{-2}\ln\rho)$ iterations, by incorporating a simple bisection idea and obtained Algorithm 3 with the slightly better $O(\rho^2\delta^{-2} + \delta^{-2}\ln\ln\rho)$ complexity. That is, we have improved the dependence on the rounding parameter ρ, but not on the error parameter δ.

We start in the following subsection by briefly describing Nesterov's smoothing technique [13] 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 at the origin only.

5.1. The setting. Nesterov [13] 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 [11, 12] 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.

1. $\mathbf{E}_2 = \mathbf{E}_2^* = \mathbf{R}^n$, $Q_2 = \operatorname{conv}\{\pm a_i \ : \ i = 1, 2, \ldots, m\}$ 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 $\theta(y) \stackrel{\text{def}}{=} \min_x \{\langle A^* y, x\rangle \ : \ x \in Q_1\}$, 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:

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

[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.

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 additive 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 $Q_1 = \mathcal{L}$):

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

We assume that d_1 vanishes at its center and hence the above properties imply

$$d_1(x) \geq \tfrac{1}{2}\sigma_1 \|x - x_0\|_1^2.$$

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. Notice that for $d_1(x) = \frac{1}{2}\|x\|_1^2 - \frac{1}{2}\|x_0\|_1^2$ we have $d_1(x) = \frac{1}{2}\|x - x_0\|_1^2$ for $x \in \mathcal{L}$.

In an analogous fashion we define the center y_0 of Q_2 and assume that d_2 vanishes at y_0. Therefore

$$d_2(y) \geq \tfrac{1}{2}\sigma_2 \|y - y_0\|_2^2.$$

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 [13], 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

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

The smooth version of (P') therefore is

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

The main result of [13] is the following:

THEOREM 15 (Nesterov [13, Theorem 3]). *If we apply Algorithm 6 to problem (P'_{sm}) with smoothing parameter*

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

and if $x = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1, x_0, N)$, *then*[3]

$$\varphi(x) - \varphi^* \le \tfrac{4\|A\|_{1,2}}{N+1}\sqrt{\tfrac{D_1 D_2}{\sigma_1\sigma_2}}. \tag{5.11}$$

Algorithm 6 (Smooth) Efficient smooth method.

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

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 proposed already by Nesterov [14,15]. These methods are similar in spirit to Algorithm 2, recursively updating an upper bound on φ^*. We give an improved version of this algorithm applicable to the problems considered in the cited papers. Our contribution lies mainly in improving the theoretical complexity by incorporating a 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. Notice that the inequalities (3.5), (3.6) and (3.7) are implied by (3.1). To be able to obtain an algorithm guaranteeing a δ-approximate solution in *relative* scale, it is crucial to choose $\|x\|_1 \equiv \|x\|_G$, $x \in \mathbf{E}_1$.

If we wish to apply Algorithm 6, we need to supply to it a *bounded* subset of $\mathcal{L}$ containing the minimizer. Observe that as long as we are in the possession of an upper bound R on φ^*, (3.7) guarantees that all minimizers of (P) lie in the set

$$Q_1(R) \stackrel{\text{def}}{=} \mathcal{L} \cap \{x \ : \ \|x - x_0\|_G \le R\}.$$

The point 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.

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

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

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

and γ_μ as in (5.9), then

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

satisfies

$$\varphi(x) - \beta R \leq \varphi^* \quad \textit{and} \quad \varphi(x) \leq (1+\beta)R.$$

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) Smooth bisection scheme.

Input: $\varphi, x_0, \rho, \beta, \tau, \delta$;
$k = 0,\ x = x_0,\ L_0 = \|x_0\|_G,\ R_0 = \varphi(x_0)$;
$c = (1+\tau)(1+\beta),\ N = \left\lfloor \frac{2\sqrt{2}\|A\|_{1,2}}{\beta}\sqrt{\frac{D_2}{\sigma_2}} \right\rfloor$;
while $R_k/L_k > c$ **do**
  $R = \sqrt{\frac{L_k R_k}{1+\beta}},\ \mu = \frac{\sqrt{2}\|A\|_{1,2}R}{N+1}\sqrt{\frac{1}{\sigma_2 D_2}},\ \gamma_\mu = \frac{\|A\|^2_{1,2}}{\mu\sigma_2}$;
  $x = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1(R), x_0, N)$;
  if $\varphi(x) \leq (1+\beta)R$ **then**
    set L_{k+1}, R_{k+1} as in (4.8)
  else
    set L_{k+1}, R_{k+1} as in (4.9)
  end if
  $k = k+1$;
end while
$N = \left\lfloor 2\sqrt{2}\frac{R_k}{L_k}\|A\|_{1,2}(1+\frac{1}{\delta})\sqrt{\frac{D_2}{\sigma_2}} \right\rfloor,\ \mu = \frac{\sqrt{2}\|A\|_{1,2}R_k}{N+1}\sqrt{\frac{1}{\sigma_2 D_2}},\ \gamma_\mu = \frac{\|A\|^2_{1,2}}{\mu\sigma_2}$;
$x = \mathsf{Smooth}(\varphi_\mu, \gamma_\mu, Q_1(R_k), x_0, N)$;
Output: x

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

$$2\sqrt{2}\|A\|_{1,2}\sqrt{\frac{D_2}{\sigma_2}}\left[\frac{1}{\beta}\left\lceil \log_2\left(\frac{\ln\rho}{\ln(1+\tau)}\right)\right\rceil + (1+\tau)(1+\beta)\left(1+\frac{1}{\delta}\right)\right] \tag{5.12}$$

steps of the smooth optimization subroutine. If β and τ are constants, this becomes

$$O\left(\|A\|_{1,2}\sqrt{\frac{D_2}{\sigma_2}}\left(\ln\ln\rho + \frac{1}{\delta}\right)\right). \tag{5.13}$$

A reasonable practical choice of the paramaters β and τ is

$$\beta = \sqrt{\delta}, \quad \tau = \tfrac{1}{2}\left(\sqrt{1+\tfrac{4\beta}{\ln 2}} - 1\right). \tag{5.14}$$

5.4. A direct representation of the objective function. We can get rid of the dependence on $\|A\|_{1,2}$ in (5.13) 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 $D_2 \leq \frac{1}{2}\rho^2$; the second inequality follows from the ellipsoidal rounding inclusion (3.1). 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 the values of these parameters into (5.13) gives the complexity

$$O\left(\rho\left(\tfrac{1}{\delta} + \ln\ln\rho\right)\right).$$

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 (5.7)—a concave quadratic maximization problem over Q_2. If this problem can not be solved efficiently (say in a closed form), the method will likely be impractical.

6. Scaling and complexity.

6.1. Scaling. It is natural to ask the following question: how do the "relative-scale" algorithms developed in this paper perform when we scale the objective function? Note that if we replace A by tA for some $t > 0$ (effectively scaling φ by $t > 0$), then (3.1) holds with G replaced by t^2G. Therefore, the inequalities in Section 3.3 are valid and so are all the results of this paper. Note that, in particular, the values of ρ and $\|A\|_{1,2}$ remain unchanged. Looking at (4.4), (4.12), (4.14), (4.21) and (5.13) we see that the iteration complexities of the algorithms discussed in the paper are not affected by scaling.

6.2. Complexity comparison. Table 1 compares the iteration complexities of the algorithms discussed in this paper.

TABLE 1
Summary of iteration complexities

Method Name	Algorithm #	Number of iterations
SubSearch	2	$O(\rho^2\delta^{-2}\ln\rho)$
SubBis	3	$O(\rho^2\delta^{-2} + \rho^2\ln\ln\rho)$
SubSearchNR	4	$O(\rho^2\delta^{-2}\ln\rho)$
SubBisNR	5	$O(\rho^2\delta^{-2} + \rho^2\ln\rho)$
SmoothBis	7	$O(\rho\delta^{-1} + \rho\ln\ln\rho)$

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 having the smallest ℓ_1 norm.

In all the examples above the feasible set $\mathcal{L}$ is a hyperplane. 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 [17, 19]. The iteration complexity is also $O(\frac{1}{\delta})$, 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:

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

with Q_2 being the unit simplex in $\mathbf{R}^{2m}$ and A the $2m \times n$ matrix with rows $a_i, -a_i, \ i = 1, \dots, m$. In addition, assume that the vectors $a_i, \ i = 1, 2, \dots, m$, 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 $y_0 := (\frac{1}{2m}, \dots, \frac{1}{2m})$.

LEMMA 18. *The prox-function 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 [4, Exercise 3.3.25(d)]) or, using differentiation and properties of convex functions (Nesterov [13, 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:

$$\|A\|_{1,2} = \max_{\|x\|_1 = 1} \|Ax\|_2^* = \max_{\|x\|_G = 1} \|Ax\|_\infty = \max_{\|x\|_G = 1} \varphi(x) \overset{(3.5)}{\leq} \rho.$$

In view of (5.9) we have $\gamma_\mu \leq \frac{\rho^2}{\mu}$. It is shown in Nesterov [13, Lemma 4] that

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

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 $\rho = O(\sqrt{n})$, 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, 10, 14, 22, 23].

Complexity. If follows from (5.13) that Algorithm 7 has the complexity

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

This improves the result of Nesterov [14], where the author gives the bound

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

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

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 $D_2 = \max\{d_2(y) \ : \ y \in Q_2\} \le \frac{1}{2}m$. Finally,

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

Complexity. It follows from (5.13) that Algorithm 7 has the complexity

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

This improves the following bound of Nesterov [15]

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

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 [24] 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 Nesterov [14]. Notice that we are interested in φ as defined on Δ_n only, which is a subset of the nonnegative orthant. Let us therefore define

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

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

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

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 centrally-symmetric sets mainly due to the possibility of working only with *diagonal* positive definite matrices defining the rounding.

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

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 $\rho = O(\sqrt{n})$ and then deduce inequality (3.5), which holds for all $x \in \mathbf{R}^n_+$ (Nesterov [14, 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 iteration complexity of Algorithm 3 as applied to the problem of computing the value of a two-person matrix game with nonnegative coefficients is:

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

This improves the result of Nesterov [14, Algorithm 4.4], where the author gives the bound

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

8. Computational experiments. In this section we perform computational tests on problems of the structure described in Section 7.1:

$$\min\{\varphi_\infty(x) \equiv \max\{|\langle a_i, x\rangle|,\ i = 1, \dots, m\}\ :\ \langle d, x\rangle = 1\}. \tag{8.1}$$

All experiments were done on a Windows XP desktop with Intel Core 2 Quad Q8300 CPU @2.5GHz with 3.46GB of RAM. Algorithm 7 is run with constants β, τ as given in (5.14). Rounding of the centrally symmetric set $\partial\varphi_\infty(0) = \mathrm{conv}\{\pm a_i, i = 1, \dots, m\}$ was in all cases done by Khachyian's algorithm [9] with $\rho = 1.1\sqrt{n}$.

8.1. Data: truss topology design. The data $A = [a_1, \dots, a_m] \in \mathbf{R}^{n\times m}$ and $d \in \mathbf{R}^n$ was generated using a formulation of the truss topology design (TTD) problem in the form (8.1). For details of the the derivation we refer to [3, Section 1.3.5] and [15, 19]. A brief description of the problem will suffice for our purposes: A 2D rectangle of size $a \times b$ is discretized into $a \times b$ equidistant nodes. The a nodes "on the left" are attached to a wall and a 2D force is applied at all the remaining nodes. There are a total of $a(b-1)$ free nodes, the vector of forces is thus of dimension $n = 2a(b-1)$. In our formulation this vector is d (and always represents a single horizontal unit force applied at the right middle node in the rightward direction). The TTD problem $ttd(a, b)$ is the problem of designing a structure of bars with endpoints in the nodes, with total weight of all the bars limited, such that the total *compliance* of the truss is minimized. Compliance is a quantity proportial to the work performed by the system after the forces are applied until the nodes and bars are displaced to equilibrium. Dimension m represents the total number of potential bars. In all of the problems we allow any two nodes to be connected, overlapping bars are not allowed.

8.2. Rounding. Table 2 lists six $ttd(a, b)$ test problems and quantities L_0, R_0, q_0 and ρ obtained after the initial rounding phase (i.e., computation of G satisfying (3.1)). For convenience, matrix A has in each case been scaled so that the optimal value of each problem is 1. This does not affect the algorithms (see Section 6.1) and allows for straightforward comparison between methods that work in relative scale and absolute scale due to the fact these two notions then coincide.

Note that for all problems $\varphi(x_0)$ is already quite close to the optimal value. It is within 2.7% of optimum in the $ttd(5, 5)$ case, and within a factor of 2.33 in the $ttd(9, 9)$

TABLE 2
Initialization by ellipsoidal rounding ($\varphi^* = 1$)

problem	n	m	$L_0 = \|x_0\|_G$	$R_0 = \varphi(x_0)$	$q_0 = R_0/L_0$	$\rho = 1.1\sqrt{n}$
$ttd(3,3)$	12	28	0.4005	1.4188	3.5429	3.8105
$ttd(5,5)$	40	200	0.4053	1.0266	2.5332	6.9570
$ttd(7,7)$	84	748	0.3855	1.9134	4.9634	10.0817
$ttd(9,9)$	144	2040	0.3717	2.3301	6.2690	13.2000
$ttd(21,5)$	200	3332	0.5257	2.1363	4.0637	15.5563

case. We know from (4.2) that $\varphi(x_0) \leq \rho$ must hold, the actual initial function values for our problems are much better than this bound. This suggests that the rounding stage does a very good job in pre-processing the problem for the optimization stage[5].

8.3. Nonrestarting vs restarting. In this test we will contrast the behavior of the nonrestarting variant of the subgradient search scheme (SubSearchNR) against its restarting version (SubSearch). Let us equip both methods with the additional ability to quit the subgradient subroutine at step k in case a point x is found for which $\varphi(x) < \varphi(\hat{x}_{k-1})/c$. It is particularly interesting to see what happens for extremely small values of the decrease factor: we will choose $c = 1 + 10^{-8}$. This choice essentially means that the subgradient subroutine will be left immediately after a point is found which is better than the current best point (it also means that the theoretical complexity of both methods blows up). The nonrestarting method should have a clear advantage: it starts the next subgradient subroutine from the current best point, and hence it should not take too long before a new better point is found. In contrast, the restarting version starts the whole process again from x_0. Both methods will run their next call of the subgradient subroutine with smaller stepsizes. Method SubSearchNR with extremely small c can in view of (4.13) be interpreted as a subgradient method which adjusts its stepsize as soon as it gets new information about the distance of the current best point to the set of minimizers, which happens everytime a new best point is found.

TABLE 3
Nonrestarting vs restarting: $ttd(9,9)$

	SubSearchNR		SubSearch	
$\hat{\delta}$	$\delta = 0.1$	$\delta = 0.01$	$\delta = 0.1$	$\delta = 0.01$
10%	117 (3)	631 (6)	2,821 (69)	107,722 (433)
9%	134 (3)	703 (9)	3,666 (98)	122,493 (495)
8%	181 (31)	827 (15)	5,473 (116)	150,703 (592)
7%	207 (4)	1084 (12)	6,830 (167)	201,754 (862)
6%	223 (2)	1387 (9)	7,591 (213)	292,715 (1,125)
5%	420 (33)	1744 (5)	8,572 (278)	374,557 (1,468)

The results of this comparison, for the $ttd(9,9)$ problem, are given in Table 3. For both methods we list the number of lower level subgradient iterations N it takes to achieve a certain relative accuracy level $\hat{\delta}$ (this is not the target accuracy δ that enters the method as an input). The number in the parentheses is the number of subgradient steps in the *last* call of the subgradient subroutine, the one in which the $\hat{\delta}$-approximate point was found. The difference between the two methods is clear. For any given accuracy $\hat{\delta}$, the total number of subgradient steps of the nonrestarting method (SubSearchNR) is approximately equal to the number of subgradient steps of *the last* call of the subgradient subroutine in the restarting version (SubSearch).

[5]For an $O(1/\delta)$ algorithm in which the rounding and optimization stages coincide, see [19].

Notice that the choice of δ has huge effect on the performance of both methods as it directly affects the stepsize of the subgradient subroutines. Smaller δ leads to bigger $N \sim (1+\frac{1}{\delta^2})$, which in turn leads to smaller stepsize $\kappa = \frac{R}{\sqrt{N+1}}$. A similar but milder effect occurs for the smooth methods as well: small δ increases N, which decreases the smoothing parameter μ, which increases the Lipschitz constant L_μ of $\nabla\varphi_\mu$, which in turn leads to smaller steps via (8.3) ($t = L_\mu/2$).

8.4. Relativity speedup. In this test we compare the fastest of our methods, SmoothBis (Algorithm 7), to Smooth (Algorithm 6) applied to the smoothed version of each problem *directly*, in the spirit of Theorem 15. We set $\delta = 0.01$ for SmoothBis and $\epsilon = 0.01$ for Smooth (these settings have the same meaning as $\varphi^* = 1$ in all test problems). For each of the test problems and both algorithms we list the number of iterations N of the smooth subroutine (each comprising two subproblems of the form (8.2)), time t in seconds, and the accuracy $\hat{\delta}, \hat{\epsilon}$ at termination. Both algorithms are run for the full number of iterations, as prescribed by theory. The results are given in Table 4 and the bar structure of the resulting optimal trusses in Figure 8.4. Note that the method working in relative scale (SmoothBis) is faster on all test problems except $ttd(3,3)$, both in terms of speed and iteration count. That is, we do not pay for obtaining a result in relative scale; quite to the contrary, we benefit from it.

TABLE 4
Relativity speedup

	Smooth (Alg 6), $\epsilon = 0.01$			SmoothBis (Alg 7), $\delta = 0.01$		
problem	N	t	$\hat{\epsilon}$	N (saving)	t (saving)	$\hat{\delta}$
$ttd(3,3)$	2,594	0.2"	$< 10^{-20}$	2,990	0.3"	$< 10^{-20}$
$ttd(5,5)$	7,863	1.1"	5.6×10^{-4}	6,030 (23.3%)	0.9" (21.0%)	4.9×10^{-4}
$ttd(7,7)$	15,091	7.3"	3.6×10^{-4}	9,344 (38.2%)	4.2" (42.5%)	3.2×10^{-4}
$ttd(9,9)$	22,245	54.4"	6.3×10^{-4}	13,053 (41.3%)	32.2" (40.8%)	5.7×10^{-4}
$ttd(21,5)$	27,891	140.5"	4.8×10^{-4}	15,961 (42.8%)	77.7" (44.7%)	4.3×10^{-4}

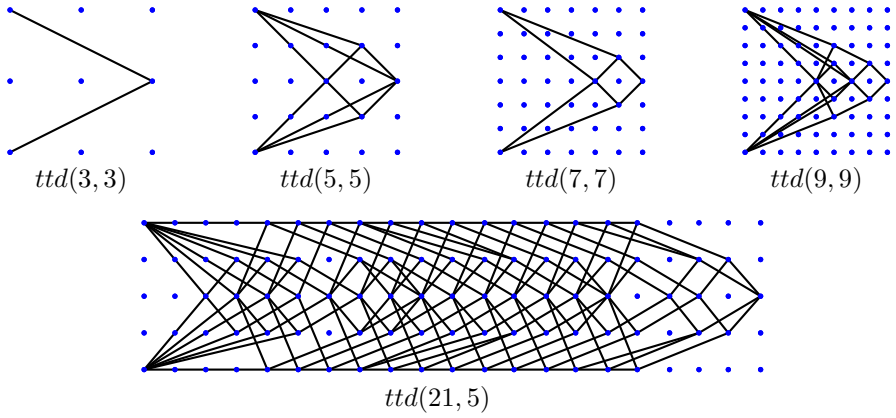

FIG. 2. *Bar structure of optimal trusses ($\delta = 0.001$)*

8.5. Bisection speedup. Let us now compare the smooth bisection scheme, SubBis (Algorithm 7), with Algorithm 3.9 of Nesterov [14] (let us call it SmoothSearch)—a smooth analogue of Algorithm 2 in which the role of the subgradient subroutine

Subgrad is replaced by Smooth. The iteration complexity of SmoothSearch is

$$\sqrt{8}e\rho(1+\ln\rho)\sqrt{\ln(2m)}(1+\tfrac{1}{\delta}),$$

with each step comprising of two operations of type (8.2). Table 5 compares the methods on a single problem, $ttd(9,9)$, for several target relative accuracies δ. Results similar to these were observed also for the other test problems and we therefore do not list them. In particular, for each δ and both methods, we list the number of iterations N, running time t in seconds and the accuracy level $\hat{\delta}$ at termination. Both methods are run for the full number of iterations as prescribed by the iteration complexity analysis. For the faster method (SmoothBis) we also list the percentage savings in iteration count and time as compared to the slower method. Notice that for both algorithms, the number of iterations increases linearly with decreasing δ, as predicted by the theory. In all cases the termination accuracy is higher than the target accuracy by a bit more than an order of magnitude. Finally, observe that the advantage of SmoothBis *grows*, both in speed and number of iterations, with increasing accuracy demand.

TABLE 5
Bisection speedup: $ttd(9,9)$

	SmoothSearch			SmoothBis		
δ	N	t	real δ	N (saving)	t (saving)	real δ
0.05	6,145	18.1"	2.2×10^{-3}	3,289 (46.5%)	12.3" (32.0%)	2.4×10^{-3}
0.01	29,555	68.9"	4.8×10^{-4}	13,053 (55.8%)	33.7" (51.1%)	5.7×10^{-4}
0.005	58,818	125.0"	2.4×10^{-4}	24,694 (58.0%)	57.3" (54.2%)	3.0×10^{-4}
0.001	292,919	575.3"	4.8×10^{-5}	116,153 (60.4%)	225.9" (60.7%)	6.0×10^{-5}
0.0005	585,546	1078.1"	2.4×10^{-5}	229,065 (60.9%)	440.7" (59.1%)	3.0×10^{-5}

8.6. Algorithm 6. Steps 4–5 of Algorithm 6 are of the form

$$\min\{\langle s,x\rangle + t\|x-\bar{x}\|_G^2 \;:\; \langle d,x\rangle = 1,\ \|x-x_0\|_G \le R\}, \tag{8.2}$$

where $\langle d,\bar{x}\rangle = 1$ and $t>0$. The solution is given by

$$x = x_0 - \tfrac{1}{2(t+\alpha)}G^{-1}(s'+\lambda d), \quad \text{where} \tag{8.3}$$

$$s' = s + 2tG(x_0-\bar{x}), \quad \lambda = \langle s', x_0\rangle, \quad v = \|s'+\lambda d\|_G^*, \quad \alpha = \begin{cases} \frac{v}{2R} - t, & v \ge 2Rt, \\ 0, & \text{otherwise.} \end{cases}$$

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*, Optimization Methods and Software, 23 (2008), pp. 5–19.

[2] ALEXANDRE BELLONI AND ROBERT M. FREUND, *On the symmetry function of a convex set*, Math. Program., 111 (2007), pp. 57–93.

[3] A. BEN-TAL AND A. NEMIROVSKI, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[4] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization*, Advanced Books in Mathematics, Canadian Mathematical Society, 2000.
[5] F. A. Chudak and V. Eleutério, *Improved approximation schemes for linear programming relaxations of combinatorial optimization problems.*, in IPCO'05, Berlin, 2005.
[6] J.-L. Goffin, *On convergence rates of subgradient optimization methods*, Mathematical Programming, 13 (1977), pp. 329–347.
[7] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.
[8] 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.
[9] L. G. Khachiyan, *Rounding of polytopes in the real number model of computation*, Mathematics of Operations Research, 21 (1996), pp. 307–320.
[10] P. Kumar and E. A. Yildirim, *Minimum volume enclosing ellipsoids and core sets*, Journal of Optimization Theory and Applications, 126 (2005), pp. 1–21.
[11] 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.
[12] ———, *Introductory Lectures on Convex Optimization. A Basic Course*, vol. 87 of Applied Optimization, Kluwer Academic Publishers, Boston, 2004.
[13] ———, *Smooth minimization of non-smooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.
[14] ———, *Rounding of convex sets and efficient gradient methods for linear programming problems*, Optimization Methods and Software, (2008), pp. 109–128.
[15] ———, *Unconstrained convex minimization in relative scale*, Mathematics of Operations Research, 34 (2009), pp. 180–193.
[16] ———, *Barrier subgradient method*, CORE Discussion Paper #2008/60, (October 2008).
[17] 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.
[18] ———, *Approximate level method*, CORE Discussion Paper #2008/83, (December 2008).
[19] ———, *Simultaneously solving seven optimization problems in relative scale*, Manuscript, (December 2008).
[20] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1997. Reprint of the 1970 original, Princeton Paperbacks.
[21] N. Z. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, 1985.
[22] P. Sun and R. M. Freund, *Computation of minimum-volume covering ellipsoids*, Oper. Res., 52 (2004), pp. 690–706.
[23] M. J. Todd and E. A. Yildirim, *On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids*, Discrete Applied Mathematics, 155 (2007), pp. 1731–1744.
[24] J. von Neumann and O. Morgenstern, *The Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, USA, 1948.