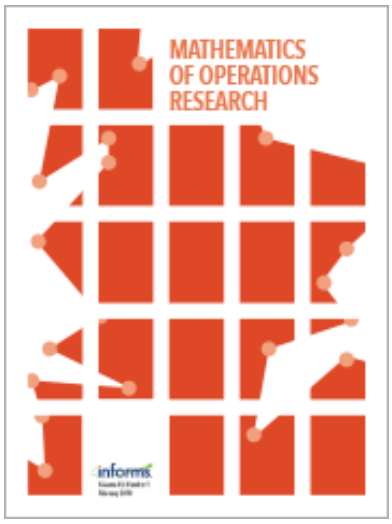

Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Cost of Nonconvexity in Deterministic Nonsmooth Optimization

Siyu Kong, A. S. Lewis

To cite this article:
Siyu Kong, A. S. Lewis (2023) The Cost of Nonconvexity in Deterministic Nonsmooth Optimization. Mathematics of Operations Research

Published online in Articles in Advance 29 Nov 2023

. https://doi.org/10.1287/moor.2022.0289

https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–17
ISSN 0364-765X (print), ISSN 1526-5471 (online)

The Cost of Nonconvexity in Deterministic Nonsmooth Optimization

Siyu Kong,[a] **A. S. Lewis**[a,*]

[a]School of Operations Research and Information Engineering, Cornell University, Ithaca, New York 14853
*Corresponding author
Contact: sk3333@cornell.edu, https://orcid.org/0009-0005-9856-1555 (SK); adrian.lewis@cornell.edu, https://orcid.org/0000-0002-1785-1106 (ASL)

Received: October 7, 2022
Revised: May 12, 2023; July 21, 2023
Accepted: September 19, 2023
Published Online in Articles in Advance: November 29, 2023

MSC2020 Subject Classifications: Primary: 90C56, 49J52, 65Y20

https://doi.org/10.1287/moor.2022.0289

Abstract. We study the impact of nonconvexity on the complexity of nonsmooth optimization, emphasizing objectives such as piecewise linear functions, which may not be weakly convex. We focus on a dimension-independent analysis, slightly modifying a 2020 black-box algorithm of Zhang-Lin-Jegelka-Sra-Jadbabaie that approximates an ϵ-stationary point of any directionally differentiable Lipschitz objective using $O(\epsilon^{-4})$ calls to a specialized subgradient oracle and a randomized line search. Seeking by contrast a *deterministic* method, we present a simple black-box version that achieves $O(\epsilon^{-5})$ for any difference-of-convex objective and $O(\epsilon^{-4})$ for the weakly convex case. Our complexity bound depends on a natural nonconvexity modulus that is related, intriguingly, to the negative part of directional second derivatives of the objective, understood in the distributional sense.

Funding: This work was supported by the National Science Foundation [Grant DMS-2006990].

Keywords: nonsmooth optimization • nonconvex • Goldstein subgradient • complexity • distributional derivative

1. Introduction

We consider the problem of minimizing a Lipschitz objective function $f : \mathbf{R}^n \to \mathbf{R}$. We suppose that we are given a Lipschitz constant, an initial point $x_0 \in \mathbf{R}^n$, and an upper bound on the gap $f(x_0) - \inf f$. We have access to f at input points $x \in \mathbf{R}^n$ through an oracle that outputs only local information, such as the function value $f(x)$ and a subgradient in $\partial f(x)$. It is easy to see that the problem of approximating the minimum value $\inf f$ within a given tolerance $\epsilon > 0$ suffers from the curse of dimensionality; it requires what amounts to grid search, needing a number of oracle calls growing like $O\left(\frac{1}{\epsilon^n}\right)$, so depending exponentially on the dimension n.

Relaxing our goals, rather than minimization, we may instead seek points that are, in some sense, nearly critical. A well-known example is the case of a smooth but nonconvex objective function $f : \mathbf{R}^n \to \mathbf{R}$, for which finding a point $x \in \mathbf{R}^n$ satisfying $|\nabla f(x)| \le \epsilon$ is relatively easy. Assuming that f is bounded below and L-smooth, meaning that its gradient has a known Lipschitz constant L, elementary calculus shows that the gradient descent iteration $x \leftarrow x - \frac{1}{L}\nabla f(x)$ always reduces the objective value $f(x)$ by at least $\frac{1}{2L}|\nabla f(x)|^2$. Assuming a bound M on the gap between the initial objective value and $\inf f$, the algorithm succeeds after no more than $\frac{2LM}{\epsilon^2}$ iterations, independent of the dimension n.

Less well known is an interesting analogous result for objectives f that are nonsmooth but convex; Davis and Drusvyatskiy [4] present a randomized algorithm that finds a point within a distance ϵ of some point at which f has a subgradient with norm no larger than ϵ using $\tilde{O}\left(\frac{1}{\epsilon^2}\right)$ subgradient evaluations. The $\tilde{O}(\cdot)$ notation suppresses logarithmic factors, but again, the complexity estimate is dimension independent. When f is just *weakly* convex, meaning that the function $f + \frac{\rho}{2}|\cdot|^2$ is convex for some constant $\rho > 0$, the analogous algorithm (see Davis and Drusvyatskiy [5]) still has a dimension-independent complexity bound, now of the form $O\left(\frac{1}{\epsilon^4}\right)$.

For general Lipschitz functions f, however, the analogous problem is intractable (see Kornowski and Shamir [15]). More precisely, any algorithm guaranteed to approximate within a distance ϵ a point with a Clarke subgradient of norm less than ϵ must suffer from the same curse of dimensionality as grid search, requiring a number of subgradients growing like $O\left(\frac{1}{\epsilon^n}\right)$.

Although that intractability might seem the end of the question, there is a more relaxed proxy approach to minimizing Lipschitz functions, dating back to work of Goldstein [10] in the 1970s. This method can be viewed as seeking a point in $\mathbf{R}^n$ around which f is differentiable on some cluster of nearby points at which the gradients have a small *convex combination*—a small "Goldstein subgradient" in the language of Goldstein [10]. Although

some published algorithms, such as that of Mahdavi-Amiri and Yousefpour [16], accomplished this goal, no complexity analysis existed until recently.

A 2020 breakthrough of Zhang et al. [24] presented an algorithm for this Goldstein subgradient problem with a complexity analysis depending on the radius δ of the cluster and the size ϵ of the subgradient but independent of the dimension n. The algorithm assumes directional differentiability of the objective f and relying on an associated directional subgradient oracle, uses an innovative randomized line search to achieve an efficiency guarantee of essentially $O(\epsilon^{-3}\delta^{-1})$. The development of Zhang et al. [24] may have practical as well as theoretical interest. The basic algorithm in Zhang et al. [24] employs "null" steps, rather like the traditional bundle methods that have long enjoyed considerable success in large-scale convex optimization (see Sagastizábal [21]). The practicality of the basic algorithm of Zhang et al. [24] remains unclear, but a related algorithm performs at least comparably with stochastic gradient descent in the authors' preliminary experiments. Two subsequent papers, Davis et al. [6] and Tian and So [22], point out that small random perturbations allow a standard subgradient oracle to replace the directional version of Zhang et al. [24].

Two recent developments, Jordan et al. [13] and Kornowski and Shamir [15] (see also Jordan et al. [14][1]), raise the question of *deterministic* algorithms for this problem. Although both papers prove positive results in the smooth case and Jordan et al. [14] thereby develop a "white-box" deterministic smoothing approach to the nonsmooth problem, both manuscripts also prove negative results for the general dimension-independent question.

However, the negative results of Jordan et al. [13] and Kornowski and Shamir [15] concern *general* Lipschitz optimization. In contrast, by modestly restricting the class of nonsmooth objectives, we are able to develop a simple deterministic black-box version of the algorithm of Zhang et al. [24] with increased but still dimension-independent complexity. Specifically, our contribution is an algorithm that achieves, up to a *nonconvexity modulus* for the objective, a dimension-independent complexity of $O(\epsilon^{-4}\delta^{-1})$, thus derandomizing the method of Zhang et al. [24] at the expense of an extra order of ϵ. When $\delta = \epsilon$, we arrive at the estimates noted in the abstract: $O(\epsilon^{-4})$ for the original method of Zhang et al. [24] and $O(\epsilon^{-5})$ for our deterministic modification (which strengthens to $O(\epsilon^{-4})$ in the weakly convex case). A higher level of nonconvexity corresponds to a larger nonconvexity modulus and hence, to a larger complexity bound; in this sense, the modulus measures a complexity "cost" for finding nearly critical points of nonconvex functions. Our analysis covers interesting objectives, such as piecewise linear functions, which are not even weakly convex. We relate the nonconvexity modulus of the objective with its distributional second derivative, hinting at an intriguing relationship between such derivatives and algorithmic complexity in general optimization.

2. The Optimization Problem and Oracle

Primarily to emphasize the elementary nature of our development, we adopt a rudimentary setting for our optimization problem. On a real inner product space $\mathbf{X}$ with corresponding norm $|\cdot|$, we consider the problem of minimizing a function $f : \mathbf{X} \to \mathbf{R}$. The objective f may be neither smooth nor convex, and the space $\mathbf{X}$ may be neither finite dimensional nor even complete. The method we develop, following Zhang et al. [24], relies on an the following underlying idea.

Definition 1. We say that an objective function $f : \mathbf{X} \to \mathbf{R}$ has a *directional subgradient map* $G : \mathbf{X}^2 \to \mathbf{X}$ when for all points $x \in \mathbf{X}$ and directions $e \in \mathbf{X}$, the *Gâteaux directional derivative*

$$f'(x;e) = \lim_{t \downarrow 0} \frac{1}{t}(f(x+te) - f(x))$$

exists and satisfies

$$\langle G(x,e), e\rangle = f'(x;e).$$

We say that G is *L-bounded* for some constant $L > 0$ if its norm $|G(x,e)|$ is never larger than L.

In applications, the objective function f is L-Lipschitz, and the vector $G(x, e)$ is a subgradient of some kind for f at the point x associated with the direction e; therefore, we loosely refer to $G(x, e)$ as a "subgradient." Nonetheless, we choose this rudimentary setting to emphasize again the elementary nature of our development, which makes no recourse to variational or Lipschitz analysis.

Example 1 (Differentiable Functions). For any function f that is L-Lipschitz and has a Gâteaux derivative $\nabla f(x)$ at every point $x \in \mathbf{X}$, the equation

$$G(x,e) = \nabla f(x)$$

defines an L-bounded directional subgradient map.

Example 2 (Convex Functions). For an L-Lipschitz convex function f with convex subdifferential ∂f, any map G satisfying

$$G(x,e) \in \operatorname{argmax}\{\langle g,e\rangle : g \in \partial f(x)\} \qquad \text{for all } \; x,e \in \mathbf{X}$$

is an L-bounded directional subgradient map.

More generally, we have the following example, which covers many interesting objectives, including the weakly convex case. For a Lipschitz function $f : \mathbf{R}^n \to \mathbf{R}$, the *Clarke subdifferential* $\partial^c f(x)$ is the convex hull of the set of all limits of the form $\lim \nabla f(x^r)$ for points $x^r \to x$ in $\mathbf{R}^n$. The function f is *subdifferentially regular* when its Gâteaux directional derivative satisfies

$$f'(x;e) = \max\{\langle g,e\rangle : g \in \partial^c f(x)\} \qquad \text{for all } \; x,e \in \mathbf{R}^n.$$

Example 3 (Subdifferentially Regular Functions). Consider any L-Lipschitz subdifferentially regular function $f : \mathbf{R}^n \to \mathbf{R}$. Then, any map G satisfying

$$G(x,e) \in \operatorname{argmax}\{\langle g,e\rangle : g \in \partial^c f(x)\} \qquad \text{for all } \; x,e \in \mathbf{R}^n$$

is an L-bounded directional subgradient map.

Notwithstanding the generality of this example, we emphasize that our framework is not restricted to objectives that are subdifferentially regular, as the following result shows.

Proposition 1 (Directional Clarke Subgradient Maps). *Any locally Lipschitz function $f : \mathbf{R}^n \to \mathbf{R}$ that is directionally differentiable has a directional subgradient map $G : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}^n$ satisfying $G(x,e) \in \partial^c f(x)$ for all $x,e \in \mathbf{R}^n$.*

Proof. We just need to prove that if f has a Gâteaux directional derivative at the point $x \in \mathbf{R}^n$ in the direction $e \in \mathbf{R}^n$, then there exists a Clarke subgradient $g \in \partial^c f(x)$ satisfying $\langle g,e\rangle = f'(x;e)$. For $r = 1,2,3,\ldots$, by the nonsmooth mean value theorem, there exists a point $x_r \in \left[x, x+\frac{1}{r}e\right]$ and a subgradient $g_r \in \partial^c f(x_r)$ satisfying

$$f\left(x+\frac{1}{r}e\right) - f(x) = \left\langle g_r, \frac{1}{r}e\right\rangle.$$

Because the subdifferential $\partial^c f$ mapping is closed and locally bounded, any limit point g of the sequence $\{g_r\}$ has the desired property. □

As discussed in Section 1, rather than trying to minimize the objective f, we instead seek a point $x \in \mathbf{X}$ that is, in some sense, approximately critical. To this end, we make the following definition. We denote the closed ball in $\mathbf{X}$ of radius δ and center x by $B_\delta(x)$.

Definition 2. Consider the setting of Definition 1. Corresponding to any constant $\delta > 0$, a *Goldstein subgradient* at x is a vector of the form

$$\sum_{i=1}^{k} \lambda_i G(x_i, e_i)$$

for a positive integer k, positive weights λ_i summing to one, points $x_i \in B_\delta(x)$, and directions $e_i \in \mathbf{X}$ for $i = 1,2,\ldots,k$. The set of all Goldstein subgradients is denoted by $\partial_\delta f(x)$.

Loosely speaking, the Goldstein subdifferential $\partial_\delta f(x)$ consists of all convex combinations of subgradients at nearby points. Strictly speaking, our notion is potentially smaller than the standard definition for Lipschitz $f : \mathbf{R}^n \to \mathbf{R}$, namely the closed convex hull of the set $\partial^c f(B_\delta(x))$.

We can now state our goal, which relies on a second constant $\epsilon > 0$.

Aim. Find a point $x \in \mathbf{X}$ and a Goldstein subgradient $g \in \partial_\delta f(x)$ such that $|g| \le \epsilon$.

The development of Zhang et al. [24] accomplishes this goal, explicitly in the setting of Proposition 1 and assuming that f is directionally differentiable in the (stronger) Hadamard sense. It relies on the following oracle.

Oracle 1 (Directional Subgradient)

Input:
- a point $x \in \mathbf{X}$,
- a direction $e \in \mathbf{X}$.

Output:
- the objective value $f(x)$,
- the directional derivative $f'(x;e)$,

- a subgradient-like vector $G(x, e)$.

In this work, we rely on the same directional subgradient oracle. We emphasize that this oracle is stronger than the standard subgradient oracle. We cannot directly compare our approach with algorithms relying only on the standard oracle because the extra computational cost of the directional oracle is hidden in our analysis, just as it is in Zhang et al. [24]. How generally available a directional oracle might be is unclear; a cautionary point of comparison is the NP hardness of deciding the existence of descent directions (see Nesterov [18]).

In generic practice, however, for a Lipschitz objective f, we may expect that the algorithm we describe never encounters points x where f is nondifferentiable, in which case any subgradient oracle simply returns $g = \nabla f(x)$ (see Bianchi et al. [1]). More formally, nonetheless, we must consider nonsmooth points. Undeterred, Zhang et al. [24] argue that availability of the directional oracle, although a nontrivial restriction, may be a reasonable assumption, directional subgradients being potentially computable via nonsmooth calculus rules. In contrast with Zhang et al. [24], our aim here is a fully deterministic algorithm. Accordingly, although we use this same stronger oracle, we instead use a deterministic line search, much as in Kornowski and Shamir [15]. To ensure termination, we make a mild assumption about the directional behavior of the objective function f, similar in spirit to the idea of *semismoothness* (see Mifflin [17]) common in nonsmooth computation but simpler and weaker.

Rather than the directional subgradient oracle on which we here rely, combined with a line search, one might instead consider other potential oracles. In particular, given an oracle that returns a descent direction, one might try to mimic smooth techniques, simply searching along that direction. However, even laying aside the NP hardness of a general descent direction oracle noted, such a basic approach is flawed. A classical example of Wolfe [23] shows that gradient descent with exact line search on a nonsmooth continuous convex function can encounter only smooth points and yet, converge to a point that is not a minimizer. Nonsmoothness necessitates a more robust approach.

Definition 3. Consider a directionally differentiable function $f : \mathbf{X} \to \mathbf{R}$. We call f *directionally semismooth* if all points $x \in \mathbf{X}$ and directions $e \in \mathbf{X}$ satisfy

$$\lim_{t \downarrow 0} f'(x + te; e) = f'(x; e).$$

On the other hand, following the approach of Facchinei and Pang [9], f is *semismooth* if it is Lipschitz, and the following stronger property holds:

$$f'(x + e; e) - f'(x; e) = o(e) \quad \text{as} \quad e \to 0.$$

Most Lipschitz functions in practice are semismooth, including in particular, difference-of-convex functions and semialgebraic functions; see Bolte et al. [3]. As we shall see, directional semismoothness suffices to guarantee termination of our algorithm, but before describing it, we focus first on the line search.

Following Zhang et al. [24], as is usual in black-box-style analysis, we suppose that the optimizer has no access to the implementation of Oracle 1 (directional subgradient). In one interesting class of practical examples, the output "subgradients" may be generated cheaply but opaquely by standard autodifferentiation algorithms, like that in PyTorch (see Paszke et al. [19]). Indeed, in the image classification experiments in Zhang et al. [24], where precise implementation of the directional subgradient oracle seems challenging, the authors simply use as a heuristic an autodifferentiation routine, assuming that it should never encounter nonsmooth ingredients in practice. In fact, even in the presence of nonsmooth ingredients, the occasional failure of autodifferentiation to output correct subgradients seems to have no impact on practical optimization, for reasons discussed by Bolte and Pauwels [2].

As a more precise illustration of the computational cost of the directional subgradient oracle, we present a simple formal example of an easily implementable version—a special case of Example 3. Consider an objective of the form $f(x) = \max_{i \in I} f_i(x)$ for smooth functions f_i indexed by a finite set I. Given an input consisting of a point x and direction e, the oracle works by first finding the active subset $I(x)$ of indices i maximizing $f_i(x)$ and then, chooses as the output subgradient any gradient $\nabla f_i(x)$ maximizing the inner product $\langle \nabla f_i(x), e \rangle$ over $i \in I(x)$. Typically, the gradient computations dominate, so the cost of the directional oracle is proportional to the size of the set I. We emphasize, however, that to be interesting, this computation should be invisible to the optimizer; given the same access to the functions f_i as the oracle, the optimizer could solve the problem easily using classical nonlinear programming tools.

3. A Simple Line Search

We pose the line search problem as a self-contained question. Consider points $p < q$ in $\mathbf{R}$ and a function $h : [p, q] \to \mathbf{R}$ satisfying $h(p) > h(q)$. Suppose that h is right differentiable on the interval $[p, q)$, and an oracle returns,

for any input $t \in [p,q)$, the value $h(t)$ and the right derivative

$$h'_+(t) = \lim_{s \downarrow t} \frac{h(s) - h(t)}{s - t}$$

(possibly extended valued). How difficult is it to find a point t satisfying $h'_+(t) < 0$?

When h is Lipschitz, the most basic randomized strategy—uniformly sampling random points t in the interval—solves this problem with high probability. Denoting the Lipschitz constant by L, the right derivative h'_+ always lies in the interval $[-L, L]$. Denote the measure of the subset S of the interval $[p,q]$ where $h'_+ \geq 0$ by λ. Then, providing that the average slope satisfies

$$\frac{h(q) - h(p)}{q - p} = -\sigma < 0,$$

the fundamental theorem of calculus implies

$$-\sigma(q - p) = \int_p^q h'_+(t)\, dt.$$

Because the integrand is bounded below by zero on S and by $-L$ on the complement S^c, a set of measure $(q - p) - \lambda$, we deduce

$$-\sigma(q - p) \geq L\Big(\lambda - (q - p)\Big).$$

Hence, the probability $\frac{\lambda}{q-p}$ that a uniformly distributed random point $t \in [p,q]$ fails to satisfy $h'(t) < 0$ is no larger than $1 - \frac{\sigma}{L}$. Thus, for small σ, using at least $\frac{L}{\sigma}$ independent samples, the probability of success is at least $\frac{1}{2}$.

However, we seek a deterministic algorithm, so we instead consider the following simple method, similar in spirit to one described by Davis et al. [6]. We repeatedly bisect the interval $[p,q]$, each time discarding the subinterval over which the function h decreases the least. The algorithm checks whether the right derivative at the midpoint of the current interval is negative, in which case it terminates.

Algorithm 1 (Search h by Bisection for Negative Derivative)
input: initial interval $[p,q]$
if $h'_+(p) < 0$ **then**
  return p
end if
$l = p$
$r = q$
while not done **do**
  $m = \frac{1}{2}(l + r)$
  if $h'_+(m) < 0$ **then**
    return m
  else if $2h(m) < h(l) + h(r)$ **then**
    $r = m$
  else
    $l = m$
  end if
end while

Notice that the algorithm initially calls the oracle at the left end point p of the given interval, calculates the function value at the right end point q, and then, calls the oracle once during each bisection.

In general, this algorithm may fail to terminate. It is easy to construct a Lipschitz function h satisfying $h(p) > h(q)$, and yet, the derivative of h at the initial end point p and at every midpoint m is positive. To rule out such oscillatory examples, we can rely on directional semismoothness of h, which in this univariate setting,

simply means that the right derivative exists and is right continuous,

$$\lim_{t\downarrow\bar{t}} h'_+(t) = h'_+(\bar{t}),$$

and that the left derivative also exists and is left continuous:

$$h'_-(t) = \lim_{s\uparrow t} \frac{h(s)-h(t)}{s-t} \qquad \text{satisfies} \qquad \lim_{t\uparrow\bar{t}} h'_-(t) = h'_-(\bar{t}).$$

For Lipschitz functions h, these two properties amount exactly to the property of semismoothness, as discussed at the end of Section 2. Most Lipschitz functions in practice are semismooth, including in particular, convex and concave functions and piecewise smooth functions. Furthermore, any linear combination of semismooth functions is semismooth. When the function h is semismooth, the Clarke subdifferential is given by

$$\partial^c h(t) = \operatorname{conv}\{h'_-(t),\ h'_+(t)\},$$

and the following property also holds (see Henrion and Outrata [12, lemma 2.2] and Mifflin [17, lemma 2]):

$$\lim_{t\downarrow\bar{t}} h'_-(t) = h'_+(\bar{t}) \qquad \text{and} \qquad \lim_{t\uparrow\bar{t}} h'_+(t) = h'_-(\bar{t}).$$

Semismoothness is more than enough to prove termination of the line search. The simple argument also applies to non-Lipschitz functions.

Proposition 2 (Termination of the Line Search). *Suppose that the function $h : [p,q] \to \mathbf{R}$ satisfies $h(p) > h(q)$ and that its left and right derivatives satisfy the semismoothness conditions*

$$\lim_{t\downarrow\bar{t}} h'_+(t) = h'_+(\bar{t}) \quad \text{for } \bar{t} \in [p,q)$$
$$\lim_{t\uparrow\bar{t}} h'_+(t) = h'_-(\bar{t}) \quad \text{for } \bar{t} \in (p,q].$$

Then, Algorithm 1 terminates.

Proof. If the algorithm does not terminate, then it generates monotonic sequences $l_k \uparrow$ and $r_k \downarrow$, satisfying $r_k - l_k \to 0$,

$$h'_+(l_k) \geq 0, \qquad \text{and} \qquad \frac{h(r_k)-h(l_k)}{r_k-l_k} \leq -\sigma < 0$$

for each iteration $k = 0, 1, 2, \ldots$. (The line search ensures that the ratio in the second inequality never increases.) Denote the two sequences' mutual limit by $\overline{m}$. Semismoothness ensures $h'_+(l_k) \to h'_-(\overline{m})$, so $h'_-(\overline{m}) \geq 0$.

If $r_k = \overline{m}$ for all large k, then

$$\frac{h(r_k)-h(l_k)}{r_k-l_k} = \frac{h(\overline{m})-h(l_k)}{\overline{m}-l_k} \to h'_-(\overline{m}),$$

which is a contradiction. Hence, for all large k, we have $q > r_k > \overline{m}$. The right end points r_k decrease to $\overline{m}$ and so, are always updated eventually; hence, the line search guarantees $h'_+(r_k) \geq 0$. Consequently, semismoothness ensures $h'_+(r_k) \to h'_+(\overline{m})$, so $h'_+(\overline{m}) \geq 0$. We deduce

$$h(\overline{m}) - h(l_k) \geq -\frac{\sigma}{2}(\overline{m}-l_k) \qquad \text{and} \qquad h(r_k) - h(\overline{m}) > -\frac{\sigma}{2}(r_k-\overline{m}).$$

Adding now gives a contradiction. □

4. The Optimization Algorithm

To minimize a locally Lipschitz function $f : \mathbf{X} \to \mathbf{R}$ using the directional subgradient oracle described, we study the following algorithm. The method we describe is essentially that of Zhang et al. [24] but with the deterministic line search described in the preceding section.

Algorithm 2 (Minimize Nonsmooth f with Subgradient Oracle G)

input: tolerance $\epsilon > 0$, radius $\delta > 0$, initial point $x \in \mathbf{X}$
$g = G(x, 0)$ % *Initialize subgradient.*
while not done **do**
  if $|g| \le \epsilon$ **then**
    return x % *Small subgradient so stop.*
  end if
  $\hat{g} = \frac{g}{|g|}$ % *Normalize subgradient.*
  $x' = x - \delta\hat{g}$ % *Trial step of fixed length.*
  if $f(x) - f(x') \ge \frac{\delta\epsilon}{3}$
    $x = x'$ % *Sufficient decrease so update point.*
    $g = G(x, 0)$ % *Reinitialize subgradient.*
  else % *Insufficient decrease so update subgradient.*
    Define h on $[0, \delta]$ by
      $x(t) = x + (t - \delta)\hat{g}$
      $h(t) = f(x(t)) - \frac{\epsilon t}{2}$.
    Apply Algorithm 1 (bisection) using the formula
      $h'_+(t) = \langle G(x(t), \hat{g}), \hat{g}\rangle - \frac{\epsilon}{2}$
    to find $t \in [0, \delta]$ satisfying $h'_+(t) < 0$.
    g = shortest vector in $[g, G(x(t), \hat{g})]$
  end if
end while

Notice that, in contrast with the classical subgradient method but like many classical approaches involving line searches, trust regions, or subgradient bundling techniques, this algorithm only updates the current iterate when it satisfies a sufficient decrease condition. Intermediate "null" steps serve to shorten the current Goldstein subgradient g.

When the objective f is directionally semismooth, Algorithm 1 terminates by Proposition 2, which in turn, guarantees termination of Algorithm 2, as we shall now prove. We use the following simple tool, following Zhang et al. [24].

Lemma 1. *If two vectors $g, g' \in B_L(0)$ satisfy $\langle g', g\rangle \le \frac{1}{2}|g|^2$, then the shortest vector g'' in the line segment $[g, g']$ satisfies*

$$|g''|^2 \le |g|^2\left(1 - \frac{|g|^2}{16L^2}\right).$$

Proof. For all $t \in [0, 1]$, we have

$$\begin{aligned}|g''|^2 &\le |g + t(g' - g)|^2 = |g|^2 + t^2|g' - g|^2 + 2t\langle g, g' - g\rangle \\ &\le |g|^2(1 + t - 2t) + t^2(|g| + |g'|)^2 \le (1 - t)|g|^2 + 4L^2t^2.\end{aligned}$$

Setting $t = \frac{|g|^2}{8L^2}$ proves the result. □

We can now prove the validity of the algorithm, again imitating parts of the argument in Zhang et al. [24], which we reproduce for ease of reading.

Theorem 1 (Finite Termination). *Suppose that we apply Algorithm 2 to a directionally semismooth function $f : \mathbf{X} \to \mathbf{R}$ that is bounded below, with tolerance $\epsilon > 0$, radius $\delta > 0$, and initial point $x_0 \in \mathbf{X}$. Suppose that the directional subgradient map in Oracle 1 is L-bounded. Then, the algorithm terminates with a point $x \in \mathbf{X}$ and a subgradient $g \in \partial_\delta f(x)$ satisfying $|g| \le \epsilon$. The number of line searches required does not exceed*

$$\left\lceil \frac{3(f(x_0) - \inf f)}{\delta\epsilon} \right\rceil \cdot \frac{16L^2}{\epsilon^2}. \tag{1}$$

Proof. Suppose that the current subgradient g is *inadequate* in the sense that, in the terminology of the algorithm description, it neither is small nor generates sufficient decrease. We then apply the bisection method, Algorithm 1,

to the given function h. The initial interval is $[p,q] = [0,\delta]$, and the average slope satisfies

$$\frac{h(q)-h(p)}{q-p} = \frac{1}{\delta}\left(f(x) - \frac{\delta\epsilon}{2} - f(x')\right) < -\frac{\epsilon}{6}.$$

We thus arrive at a subgradient $g' \in \partial_\delta f(x)$ satisfying $\langle g', \hat{g}\rangle < \frac{\epsilon}{2}$. We deduce $\langle g', g\rangle < \frac{|g|^2}{2}$. The algorithm replaces the current subgradient g by the shortest vector g'' in the line segment $[g,g']$. Because $g'' \in \partial_\delta f(x)$, we can repeat this shortening process, providing that g'' is also inadequate. Suppose that the subgradient g remains inadequate after completing k such shortening steps. Let ρ_i denote the quantity $\frac{|g|^2}{16L^2}$ after $i = 0,1,2,\ldots,k$ steps. Then, $\rho_0 \le \frac{1}{16}$, and for each I, we have

$$0 < \rho_{i+1} \le \rho_i(1-\rho_i),$$

so

$$\frac{1}{\rho_{i+1}} \ge \frac{1}{\rho_i} + \frac{1}{1-\rho_i} > \frac{1}{\rho_i} + 1.$$

Consequently,

$$\frac{1}{\rho_k} \ge 16 + k,$$

so we deduce

$$\frac{\epsilon^2}{16L^2} < \frac{|g|^2}{16L^2} \le \frac{1}{16+k}.$$

Hence, after no more than

$$16\left(\frac{L^2}{\epsilon^2} - 1\right)$$

shortening steps, each requiring one line search, we arrive at an adequate subgradient $g \in \partial_\delta f(x)$. To summarize, starting at any point with an inadequate subgradient, we require no more than $\frac{16L^2}{\epsilon^2}$ line searches before finding an adequate subgradient g.

There are now two possibilities. Either the subgradient g satisfies $|g| \le \epsilon$, in which case we stop, or we perform a reduction step, replacing the current point x by $x - \delta\frac{g}{|g|}$, thereby decreasing the objective value by at least the quantity $\frac{\delta\epsilon}{3}$. Because the objective is bounded below, beginning from the initial point x_0, this procedure terminates after no more than $\lceil \frac{3}{\delta\epsilon}(f(x_0) - \inf f)\rceil$ reduction steps, from which the line search bound (1) follows. Proposition 2 ensures that each line search requires only finitely many oracle calls, completing the proof. □

5. Complexity of the Line Search

To complete our complexity analysis for the minimization algorithm, we simply need to bound the number of oracle calls needed by each line search and multiply by our bound (1) on the number of line searches. Consider, therefore, the bisection method. When the function $h : [p,q] \to \mathbf{R}$ is convex, the problem is trivial; because

$$h(p) > h(q) \quad \Rightarrow \quad h'_+(p) < 0,$$

the algorithm terminates at the first oracle call. More generally, we proceed by correcting any lack of convexity in h by adding a convex perturbation $s : [p,q] \to \mathbf{R}$.

Recall that, for any interval J, a function $h : J \to \mathbf{R}$ is *difference of convex* when there exists a convex function $s : J \to \mathbf{R}$ such that $h + s$ is also convex. For such functions, we have the following tool.

Lemma 2. *Consider a function $h : [p,q] \to \mathbf{R}$ and a convex function $s : [p,q] \to \mathbf{R}$ such that $h + s$ is also convex. For any points $x < y$ in the interval $[p,q]$, if $h'_+(x) \ge 0$, then*

$$\frac{h(y)-h(x)}{y-x} \ge s'_+(x) - s'_-(y).$$

Proof. Denote the convex function $h + s$ by r. The convex functions r and s satisfy

$$r'_+(x) \in \partial r(x) \qquad \text{and} \qquad s'_-(y) \in \partial s(y).$$

Hence,

$$\begin{aligned} s'_-(y)(x-y) &\le s(x) - s(y) = r(x) - h(x) - r(y) + h(y) \\ &\le h(y) - h(x) + r'_+(x)(x-y) \le h(y) - h(x) + s'_+(x)(x-y), \end{aligned}$$

and the result follows. □

We can then use the change in derivative of the necessary perturbation s to bound the number of iterations in the line search.

Theorem 2. *Consider a function $h : [p,q] \to \mathbf{R}$ and a convex function $s : [p,q] \to \mathbf{R}$ such that $h + s$ is also convex. If the bisection method, Algorithm 1, evaluates h'_+, the right derivative, $k \ge 1$ times without terminating, then*

$$\frac{h(q) - h(p)}{q - p} \ge \frac{s'_+(p) - s'_-(q)}{k}.$$

Proof. We proceed by induction on the number of evaluations $k = 1, 2, 3, \ldots$. The case $k = 1$ follows immediately from Lemma 2 by setting $x = p$ and $y = q$.

Suppose that the result holds for any points $p < q$ and for any number of evaluations no larger than k. Now, consider an instance of the algorithm that completes $k + 1$ evaluations. After the first evaluation, consider the midpoint $m = \frac{1}{2}(p + q)$. There are two possible cases depending on whether

$$2h(m) < h(p) + h(q). \tag{2}$$

We consider them in turn.

Suppose first that Inequality (2) holds. After the first evaluation, the algorithm makes k further evaluations, beginning with the initial interval $[p, m]$. Hence, by the induction hypothesis,

$$\frac{h(m) - h(p)}{m - p} \ge \frac{s'_+(p) - s'_-(m)}{k}.$$

Because the algorithm did not terminate during the first two evaluations, we know $h'_+(m) \ge 0$. Applying Lemma 2 with $x = m$ and $y = q$ shows

$$\frac{h(q) - h(m)}{q - m} \ge s'_+(m) - s'_-(q).$$

Hence,

$$\begin{aligned} s'_+(p) - s'_-(q) &\le (s'_+(p) - s'_-(m)) + (s'_+(m) - s'_-(q)) \\ &\le k\frac{h(m) - h(p)}{m - p} + \frac{h(q) - h(m)}{q - m} \\ &= (k-1)\frac{h(m) - h(p)}{m - p} + \left(\frac{h(m) - h(p)}{m - p} + \frac{h(q) - h(m)}{q - m}\right) \\ &= (k-1)\frac{h(m) - h(p)}{m - p} + 2\frac{h(q) - h(p)}{q - p} \\ &< (k-1)\frac{h(q) - h(p)}{q - p} + 2\frac{h(q) - h(p)}{q - p} = (k+1)\frac{h(q) - h(p)}{q - p}, \end{aligned}$$

as required.

The case where Inequality (2) fails is similar. After the first bisection, the algorithm makes k further bisections, beginning with the initial interval $[m, q]$. Hence, by the induction hypothesis,

$$\frac{h(q) - h(m)}{q - m} \ge \frac{s'_+(m) - s'_-(q)}{k}.$$

Because the algorithm did not terminate during the first bisection, we know $h'_+(p) \geq 0$. Applying Lemma 2 with $x = p$ and $y = m$ shows

$$\frac{h(m) - h(p)}{m - p} \geq s'_+(p) - s'_-(m).$$

Hence,

$$\begin{aligned} s'_+(p) - s'_-(q) &\leq (s'_+(p) - s'_-(m)) + (s'_+(m) - s'_-(q)) \\ &\leq \frac{h(m) - h(p)}{m - p} + k\frac{h(q) - h(m)}{q - m} \\ &= \left(\frac{h(m) - h(p)}{m - p} + \frac{h(q) - h(m)}{q - m}\right) + (k-1)\frac{h(q) - h(m)}{q - m} \\ &= 2\frac{h(q) - h(p)}{q - p} + (k-1)\frac{h(q) - h(m)}{q - m} \\ &\leq 2\frac{h(q) - h(p)}{q - p} + (k-1)\frac{h(q) - h(p)}{q - p} = (k+1)\frac{h(q) - h(p)}{q - p}, \end{aligned}$$

as required. □

Definition 4. Given any interval $J \subset \mathbf{R}$, the *concave deviation* of a function $h : J \to \mathbf{R}$ is the infimum of the Lipschitz constants of convex functions $s : J \to \mathbf{R}$ such that the sum $h + s$ is also convex.

Consider, for example, a *ρ-weakly convex* function h, for some constant $\rho \geq 0$, meaning that the function $t \longmapsto h(t) + \frac{\rho}{2}t^2$ is convex.

Proposition 3. *Any ρ-weakly convex function $h : [p,q] \to \mathbf{R}$, for $\rho \geq 0$, has concave deviation at most $\frac{\rho}{2}(q-p)$.*

Proof. The function $s(t) = \frac{\rho}{2}\left(t - \frac{p+q}{2}\right)^2$ is convex, with Lipschitz constant $\frac{\rho}{2}(q-p)$, and $h + s$ is also convex. □

The concave deviation for functions that are not weakly convex may shrink more slowly than the length of the interval. For example, on the interval $[-\delta, \delta]$, the function $h(t) = -|t|^{\frac{3}{2}}$ has concave deviation $\frac{3}{2}\sqrt{\delta}$, and the piecewise linear function $-|\cdot|$ has concave deviation 1.

The concave deviation of a function $h : [p,q] \to \mathbf{R}$ that is difference of convex may not be finite. An example is the concave function $\sqrt{\cdot}$ on the interval $[0,1]$. However, if h extends to a difference-of-convex function on an open interval containing the interval $[p,q]$, then its nonconvexity bound on $[p,q]$ must be finite because we can write h as a difference-of-convex functions, each of which must be Lipschitz on $[p,q]$.

Consider any continuous semialgebraic function $h : \mathbf{R} \to \mathbf{R}$. We can partition $\mathbf{R}$ into finitely many closed intervals J such that each restriction $h|_J$ is either convex or concave. In general, such a partition may not guarantee that h is difference of convex; an example is the function $x^{\frac{1}{3}}$. However, if h is also Lipschitz, then each convex or concave ingredient $h|_J$ extends to a corresponding convex or concave Lipschitz function on $\mathbf{R}$, and from these, we can easily decompose h into a difference-of-convex Lipschitz functions. Thus, all semialgebraic Lipschitz functions on $\mathbf{R}$ are difference of convex, with finite concave deviation on any bounded interval.

Corollary 1 (Line Search Complexity). *If a function $h : [p,q] \to \mathbf{R}$ has finite concave deviation M and average rate of decrease*

$$\sigma = -\frac{h(q) - h(p)}{q - p} > 0,$$

then the number of evaluations of h'_+, the right derivative, required before the bisection method, Algorithm 1, terminates is no more than $1 + \lfloor \frac{2M}{\sigma} \rfloor$.

Proof. Suppose that the bisection method evaluates the right derivative $k \geq 1$ times without terminating. Fix any value $M' > M$. By assumption, there exists a convex function s with Lipschitz constant less than M' such that the sum $h + s$ is also convex. From Theorem 2, we deduce the inequalities

$$-\sigma \geq \frac{s'_+(p) - s'_-(q)}{k} > \frac{-2M'}{k},$$

so $k < \frac{2M'}{\sigma}$. Because M' was arbitrary, we deduce $k \leq \frac{2M}{\sigma}$, and hence, $k \leq \lfloor \frac{2M}{\sigma} \rfloor$. The result follows. □

Given an open interval I, consider a difference-of-convex function $h : I \to \mathbf{R}$. As observed by Hartman [11], such functions are characterized by having left and right derivatives everywhere, which furthermore, are of bounded variation on every compact interval in I. Any such function h also has a second derivative D^2h in the distributional sense; in general, it is a signed Radon measure on the interval I (see Dudley [7]).

We review briefly the underlying construction. Consider any convex function $s : I \to \mathbf{R}$. Its right derivative s'_+ is nondecreasing and right continuous, and hence, it defines a nonnegative Radon measure D^2s on the interval I via the property

$$(D^2s)(p,q] = s'_+(q) - s'_+(p) \qquad \text{for all } p < q \text{ in } I.$$

(We could equivalently work from the property $(D^2s)(p,q) = s'_-(q) - s'_+(p)$.) More generally, for any difference-of-convex function $h : I \to \mathbf{R}$, consider any convex function $s : I \to \mathbf{R}$ such that $h + s$ is also convex. The second derivative D^2h is just the signed measure $D^2(h+s) - D^2s$, which is independent of the choice of s. Convexity of h is characterized by the property $D^2h \geq 0$. More generally, the Jordan decomposition decomposes D^2h uniquely into a difference of nonnegative Radon measures,

$$D^2h = (D^2h)^+ - (D^2h)^-,$$

with the minimality property (see Rudin [20, p. 127]) that any other decomposition into a difference of nonnegative Radon measures $D^2h = \lambda - \mu$ satisfies $\lambda \geq (D^2h)^+$ and $\mu \geq (D^2h)^-$.

Theorem 3. *If the function $h : [p,q] \to \mathbf{R}$ is difference of convex, then its concave deviation is*

$$\frac{1}{2}(D^2h)^-(p,q).$$

Proof. Consider any convex function $s : [p,q] \to \mathbf{R}$ such that $h + s$ is also convex. The second derivatives satisfy

$$D^2h = D^2(h+s) - D^2s, \quad \text{with} \quad D^2(h+s) \geq 0 \quad \text{and} \quad D^2s \geq 0,$$

so the minimality of the Jordan decomposition implies $D^2s \geq (D^2h)^-$. If s is M-Lipschitz on $[p,q]$, then

$$M \geq \max\{-s'_+(p), s'_-(q)\} \geq \frac{s'_-(q) - s'_+(p)}{2} = \frac{1}{2}(D^2s)(p,q) \geq \frac{1}{2}(D^2h)^-(p,q).$$

If the right-hand side is infinite, this completes the proof; so, suppose that it is finite. Define a function $g : (p,q] \to \mathbf{R}$ by

$$g(t) = (D^2h)^-(p,t).$$

Then, g is a nonnegative nondecreasing left-continuous function that is bounded above and $g(t) \downarrow 0$ as $t \downarrow p$. Now, define a convex function $s : [p,q] \to \mathbf{R}$ by

$$s(t) = \int_p^t g(\tau)\, d\tau.$$

Then, $s'_-(t) = g(t)$ for all $t \in (p,q]$ and $s'_+(p) = 0$. Furthermore, we have

$$D^2(h+s) = D^2h + D^2s = D^2h + (D^2h)^- = (D^2h)^+ \geq 0,$$

so the sum $h + s$ is also convex. The function

$$t \longmapsto \tilde{s}(t) = s(t) - \frac{1}{2}s'_-(q)t$$

is also convex, as is $h + \tilde{s}$, and the function $\tilde{s}$ has Lipschitz constant

$$-\tilde{s}'_+(p) = \tilde{s}'_-(q) = \frac{1}{2}s'_-(q) = \frac{1}{2}(D^2h)^-(p,q).$$

This completes the proof. □

As an illustration, we have the following result.

Corollary 2 (Piecewise Linear Functions). *Consider a continuous piecewise linear function $h : [p,q] \to \mathbf{R}$, with m derivative discontinuities $t_1 < t_2 < \cdots < t_m$ in the interval (p, q). Define $t_0 = p$ and $t_{m+1} = q$, and let g_i be the value of the derivative on the interval (t_i, t_{i+1}) for $0 \le i \le m$. Then, h has concave deviation*

$$\frac{1}{2}\sum_{i=1}^{m}(g_{i-1} - g_i)^+.$$

If h is L-Lipschitz, then this bound is no larger than $\lceil \frac{m}{2} \rceil L$.

Proof. Denoting a unit point mass at the point t by δ_t, we have

$$D^2 h = \sum_i (g_i - g_{i-1})\delta_{t_i},$$

and hence,

$$(D^2 h)^- = \sum_i (g_{i-1} - g_i)^+ \delta_{t_i},$$

from which the claimed equation follows. The inequality is an easy consequence, using $|g_i| \le L$ for each i. □

As an illustration, we present an example that underlines why the line search complexity estimate in Corollary 1 is the best that we can expect in general. In outline, although the functions $h : [p,q] \to \mathbf{R}$ that we consider satisfy $h(p) > h(q)$, their derivatives may often be positive.

Example 4 (Optimality of the Line Search). Consider any constant $M > 0$ and a set $T \subset [0,1)$ of cardinality strictly less than $2M$. Then, there exists a function $h : [0,1] \to \mathbf{R}$ with concave deviation less than M that satisfies $h(0) = 0$ and $h(1) = -1$ and that has strictly positive derivative throughout T.

To see this, suppose first $T \subset (0,1)$. (The case when T contains zero is an easy modification.) Enumerate the points in increasing order:

$$t_1 < t_2 < \cdots < t_k,$$

where $k < 2M$. Define $h(0) = 0$ and $h(1) = -1$. Fix any small $\gamma > 0$, and define

$$h(t_i - \gamma) = -t_i - \gamma^2 \quad \text{and} \quad h(t_i + \gamma) = -t_i + \gamma^2 \quad \text{for } i = 1, 2, \ldots, k.$$

At intermediate points in $[0,1]$, define h by linear interpolation. A quick calculation, using Corollary 2, shows that h has concave deviation

$$\frac{1}{2}\sum_{i=1}^{k-1}\left(\frac{t_{i+1} - t_i + 2\gamma^2}{t_{i+1} - t_i - 2\gamma} + \gamma\right) + \frac{1}{2}\left(\frac{1 - t_k + \gamma^2}{1 - t_k - \gamma} + \gamma\right) = \frac{k}{2} + O(\gamma) < M$$

providing that γ is sufficiently small.

Now, consider any line search method applicable to functions $h : [0,1] \to \mathbf{R}$ satisfying $h(0) = 0$ and $h(1) = -1$, relying on evaluations of the value h and the right derivative h'_+ at points chosen one by one, and terminating once a derivative is negative. Suppose that the method is guaranteed to terminate after at most k queries, providing that the underlying function h has concave deviation strictly less than some given value $M > 0$. The example proves $k \ge 2M$.

6. Multivariate Functions

To understand the complexity of Algorithm 2 (nonsmooth minimization), we apply our analysis in the previous section to restrictions of multivariate objectives f to line segments. For any convex set $C \subset \mathbf{X}$, consider a function $f : C \to \mathbf{R}$. Given any length $\delta > 0$, let $\Lambda(\delta)$ denote the supremum over all points $x, y \in C$ with $|x - y| \le \delta$ of the concave deviation for the function $h : [0,\delta] \to \mathbf{R}$ defined by

$$h(t) = f\left(x + \frac{t}{\delta}(y - x)\right). \tag{3}$$

We call the function $\Lambda : \mathbf{R}_{++} \to [0, +\infty]$ the *nonconvexity modulus* for f. The following illustration follows immediately from Proposition 3.

Proposition 4. *The nonconvexity modulus of any ρ-weakly convex function (for $\rho \geq 0$) satisfies*

$$\Lambda(\delta) \leq \frac{\rho\delta}{2}.$$

More generally, the function f is *difference of convex* when there exists a convex function $q : C \to \mathbf{R}$ such that $f + q$ is also convex.

Proposition 5. *Consider a convex set $C \subset \mathbf{X}$ and functions $f, q : C \to \mathbf{R}$ with both q and $f + q$ convex. If q is M-Lipschitz, then the nonconvexity modulus of f satisfies $\Lambda(\delta) \leq M$ for all $\delta > 0$.*

Proof. Consider any points $x, y \in C$ with $|x - y| \leq \delta$ and the function h defined by Equation (3). The function $s : [0, \delta] \to \mathbf{R}$ defined by

$$s(t) = q\left(x + \frac{t}{\delta}(y - x)\right)$$

is convex and M-Lipschitz, and $h + s$ is convex; therefore, the concave deviation of h is no larger than M. The result follows. □

As a consequence, we deduce the following result.

Corollary 3. *Consider any convex sets $C \subset C' \subset \mathbf{R}^n$, where C is nonempty and compact and C' is open, and any difference-of-convex function $f : C' \to \mathbf{R}$. Then, the nonconvexity modulus of the restriction f_C is uniformly bounded; there exists a finite constant M such that $\Lambda(\delta) \leq M$ for all $\delta > 0$.*

In particular, because polyhedral functions are globally Lipschitz, we have the following fact.

Corollary 4. *If a function $f : \mathbf{X} \to \mathbf{R}$ is the difference $p - q$ between a convex function $p : \mathbf{X} \to \mathbf{R}$ and a polyhedral convex function $q : \mathbf{X} \to \mathbf{R}$, then the nonconvexity modulus of f is no larger than any Lipschitz constant for q.*

More generally, consider a continuous function $f : \mathbf{X} \to \mathbf{R}$ that is *semilinear* in the sense that $\mathbf{X}$ is a finite union of polyhedra, on each of which the function f is affine. Any such function has a Lipschitz constant L and furthermore, a uniform upper bound m on the number of possible gradient discontinuities in any function of the form (3). By Corollary 2, we deduce that the nonconvexity modulus $\Lambda(\delta)$ is no larger than $\lceil \frac{m}{2} \rceil L$.

Returning to our analysis of Algorithm 2, we are ready for our main result.

Theorem 4 (Complexity of Minimization). *Given a tolerance $\epsilon > 0$ and a radius $\delta > 0$, consider a convex set $C \subset \mathbf{X}$, a function $f : C \to \mathbf{R}$ that is bounded below, an associated L-bounded directional subgradient map $G : \mathbf{X}^2 \to \mathbf{X}$, and an initial point $x_0 \in C$ such that*

$$f(x) \leq f(x_0) \quad \text{and} \quad |y| \leq \delta \quad \Rightarrow \quad x + y \in C.$$

Suppose that f has finite nonconvexity modulus $\Lambda(\delta)$. Then, Algorithm 2 (nonsmooth minimization) requires at most

$$\left\lceil \frac{3(f(x_0) - \inf f)}{\delta\epsilon} \right\rceil \cdot \frac{16L^2}{\epsilon^2} \cdot \left(1 + \left\lfloor \frac{12\Lambda(\delta)}{\epsilon} \right\rfloor\right)$$

calls to Oracle 1 (directional subgradient) to find a point $x \in \mathbf{X}$ and a Goldstein subgradient $g \in \partial_\delta f(x)$ satisfying $|g| \leq \epsilon$.

Proof. Corollary 1 with $\sigma = \frac{\epsilon}{6}$ shows that the bisection method requires at most

$$1 + \left\lfloor \frac{12\Lambda(\delta)}{\epsilon} \right\rfloor$$

oracle calls to terminate. Then, one further call returns the desired subgradient $g' \in \partial_\delta f(x)$ satisfying $\langle g', g \rangle < \frac{|g|^2}{2}$. Multiplying by the bound (1) on the number of line searches completes the argument. □

When the objective f is weakly convex, Proposition 4 implies a complexity bound of the form $O\left(\left(\frac{1}{\epsilon^3}\right)\left(\frac{1}{\epsilon} + \frac{1}{\delta}\right)\right)$. In the case $\epsilon = \delta$, we arrive at the bound $O\left(\frac{1}{\epsilon^4}\right)$ noted in the abstract.

7. Conclusion

In summary, our deterministic algorithm, when applied to difference-of-convex objectives with bounded nonconvexity modulus, returns a point with an ϵ-Goldstein subgradient of norm no larger than ϵ after $O(\epsilon^{-5})$ oracle

calls. (In the weakly convex case, Proposition 5 reduces this bound to $O(\epsilon^{-4})$.) By contrast, for arbitrary Lipschitz objectives, the algorithm of Zhang et al. [24] enjoys a superior $O(\epsilon^{-4})$ complexity bound but depends fundamentally on randomization. Our deterministic approach also has the merit of quantifying, through the modulus, how the cost in complexity of optimizing nonsmooth objectives grows with their level of nonconvexity.

Acknowledgments

The authors thank two anonymous referees for many constructive suggestions.

Appendix. Distributional Second Derivatives

We saw previously that the concave deviation of a univariate function h is determined by the negative part of its second distributional derivative D^2h. In our application, we consider functions h that are restrictions of the underlying objective f to line segments of fixed length δ. We would, therefore, expect the nonconvexity modulus of f to be related to its own distributional second derivative. Here, we explore that relationship informally.

Consider a locally Lipschitz function $f : \mathbf{R}^n \to \mathbf{R}$. The distributional derivative of f is an n-vector Df, entries of which are distributions—linear functionals on the space of smooth, compactly supported functions $g : \mathbf{R}^n \to \mathbf{R}$ (*test functions*)—that are continuous with respect to uniform convergence on compact sets. We can define Df through the relationship

$$\langle u^T(Df), g\rangle = -\int f(u^T\nabla g)$$

for all vectors $u \in \mathbf{R}^n$ and test functions $g : \mathbf{R}^n \to \mathbf{R}$. However, by a suitable version of Rademacher's theorem (see Evans and Gariepy [8, section 6.2, theorem 1]), the gradient ∇f exists almost everywhere and is essentially bounded, and it satisfies

$$\langle u^T(Df), g\rangle = \int g(u^T\nabla f).$$

In standard terminology (see Evans and Gariepy [8]), we can identify the classical gradient ∇f with both the distributional derivative Df and the "weak" derivative of f.

The second distributional derivative of f is an n-by-n matrix D^2f, entries of which are distributions. We can define D^2f through the relationship

$$\langle u^T(D^2f)v, g\rangle = -\int (u^T\nabla f)(v^T\nabla g)$$

for all vectors $u, v \in \mathbf{R}^n$ and test functions g. If f is smooth, then D^2f is just the matrix-valued measure with density $\nabla^2 f$. More generally, we must consider D^2f as a distribution, but at least for *convex* functions, we can be more specific; it is a positive semidefinite-valued Radon measure (see Dudley [7] and Evans and Gariepy [8, section 6.3]).

Example A.1 (A Piecewise Linear Function). Consider the convex function $f : \mathbf{R}^2 \to \mathbf{R}$ defined by $f(x) = x_1^+$. For any vectors $u, v \in \mathbf{R}^2$ and smooth, compactly supported function $g : \mathbf{R}^2 \to \mathbf{R}$, we have

$$\begin{aligned}\langle u^T(D^2f)v, g\rangle &= -\int_{x_1>0} u_1 v^T\nabla g(x)\,dx = -u_1\int_{x_1>0} \operatorname{div}(g(x)v)\,dx\\ &= -u_1\int_{\mathbf{R}} (-e_1)^T\left(g\left(\begin{bmatrix}0\\ y\end{bmatrix}\right)v\right)dy = u_1v_1\int g\left(\begin{bmatrix}0\\ y\end{bmatrix}\right)dy,\end{aligned}$$

by the Gauss–Green formula. Thus, D^2f is the matrix $\begin{bmatrix}\mu & 0\\ 0 & 0\end{bmatrix}$, where the measure μ is related to Lebesgue measure λ via

$$\mu(S) = \lambda\left\{s \in \mathbf{R} : \begin{bmatrix}0\\ s\end{bmatrix} \in S\right\}, \tag{A.1}$$

for all measurable subsets of $S \subset \mathbf{R}^2$.

For a more general understanding, we begin with the univariate case.

Example A.2 (Univariate Convex Functions). Consider a convex function $f : \mathbf{R} \to \mathbf{R}$. For $0 < \gamma < 1$, we can construct a smooth approximation $h_\gamma : \mathbf{R} \to [0,1]$ of the standard step function, with the following properties:

$$h_\gamma(t) = \begin{cases} 0 & (t \le 0)\\ \gamma^2 & (t = \gamma^2)\\ 1-\gamma^2 & (t = \gamma - \gamma^2)\\ 1 & (t \ge \gamma),\end{cases}$$

and h is convex on $[0,\gamma^2]$, linear on $[\gamma^2, \gamma-\gamma^2]$, and concave on $[\gamma-\gamma^2, \gamma]$. For any interval $(p,q] \subset \mathbf{R}$, the test function r_γ :

$\mathbf{R} \to [0,1]$ defined by

$$r_\gamma(t) = \begin{cases} h_\gamma(t-p) & (t \le p+\gamma^2) \\ 1 & (p+\gamma^2 \le t \le q) \\ 1-h_\gamma(t-q) & (t \ge q) \end{cases}$$

converges pointwise to the characteristic function $\chi_{(p,q]}$ pointwise as $\gamma \downarrow 0$. By dominated convergence, we deduce

$$\int r_\gamma \, d(D^2 f) \to \int \chi_{(p,q]} \, d(D^2 f) = (D^2 f)(p,q].$$

However, the left-hand side is

$$\begin{aligned} -\int r_\gamma' f' &= -\int_p^{p+\gamma} \left(\frac{1}{\gamma} + O(1)\right) f' - \int_q^{q+\gamma} \left(-\frac{1}{\gamma} + O(1)\right) f' \\ &= \frac{f(q+\gamma)-f(q)}{\gamma} - \frac{f(p+\gamma)-f(p)}{\gamma} + O(\gamma) \\ &= f_+'(q) - f_+'(p) + O(\gamma) \end{aligned}$$

as $\gamma \downarrow 0$. We, thus, reproduce our earlier definition:

$$(D^2 f)(p,q] = f_+'(q) - f_+'(p).$$

Clearly, this fact also holds for any difference-of-convex function $f : \mathbf{R} \to \mathbf{R}$.

The modulus of nonconvexity for a function $f : \mathbf{R}^n \to \mathbf{R}$, which we denoted $\Lambda(\delta)$ (for $\delta > 0$), is the supremum of the concave deviation of the restriction of f to line segments of the form

$$S = \{z + tw : 0 \le t \le \delta\}$$

for some point $z \in \mathbf{R}^n$ and unit direction $w \in \mathbf{R}^n$. That concave deviation is the measure of S under the negative part of the measure $D^2(f|_S)$. We would, therefore, like to compare the distributional second derivative of this restriction with the distributional second derivative $D^2 f$. As we see in the next result, we should focus specifically on the *directional distributional second derivative* $w^T(D^2 f)w$.

A simple approach is furnished by mollification. We fix a *mollifier* $\phi : \mathbf{R}^n \to \mathbf{R}$: a test function satisfying $\int \phi = 1$ and with the property that, as $\gamma \downarrow 0$, the function $\phi_\gamma(x) = \gamma^{-n}\phi\left(\frac{1}{\gamma}x\right)$ converges as a distribution to the Dirac delta function. Given a Radon measure μ on $\mathbf{R}^n$ and any test function $g : \mathbf{R}^n \to \mathbf{R}$, we can define the convolution $g \star \mu : \mathbf{R}^n \to \mathbf{R}$ by

$$(g \star \mu)(y) = \int g(y-x) \, d\mu(x).$$

Theorem A.1 (Chain Rule via Mollification). *Consider a locally Lipschitz function $f : \mathbf{R}^n \to \mathbf{R}$, a mollifier $\phi : \mathbf{R}^n \to \mathbf{R}$, and a direction $w \in \mathbf{R}^n$. Then, for almost all points $z \in \mathbf{R}^n$, the function f is differentiable almost everywhere on the line $z + \mathbf{R}w$, and the distributional second derivative $D^2 h$ of the function defined by*

$$h(t) = f(z+tw) \qquad (t \in \mathbf{R})$$

is the distributional limit, as $\gamma \downarrow 0$, of the convolution

$$t \longmapsto \left(\phi_\gamma \star \left(w^T(D^2 f)w\right)\right)(z+tw).$$

Proof. By Rademacher's theorem and standard properties of convolutions (see Evans and Gariepy [8, section 4.2, theorem 1(iv)]), there exists a full measure set $\Omega \subset \mathbf{R}^n$, on which f is differentiable and the convolution $\phi_\gamma \star (w^T \nabla f)$ converges pointwise to the essentially bounded function $w^T \nabla f$. By Fubini's theorem, for almost all points $z \in \mathbf{R}^n$, we have $z + tw \in \Omega$ for almost all $t \in \mathbf{R}$. Restricting attention to such z, consider any test function $g : \mathbf{R} \to \mathbf{R}$. Fubini's theorem implies

$$\begin{aligned} &\int_{t \in \mathbf{R}} g(t) \int_{x \in \mathbf{R}^n} \phi_\gamma(z+tw-x) \, d(w^T(D^2 f)w)(x) \, dt \\ &= \int_x \left(\int_t g(t) \phi_\gamma(z+tw-x) \, dt \right) d(w^T(D^2 f)w)(x) \\ &= \int_x w^T \nabla f(x) \left(w^T \int_t g(t) \nabla \phi_\gamma(z+tw-x) \, dt \right) dx. \end{aligned}$$

(We can interchange the order of differentiation and integration because the test functions g and ϕ_γ are well behaved.) Rewriting, integrating by parts, and using Fubini's theorem and dominated convergence again, we obtain

$$\begin{aligned}
&\int_x w^T\nabla f(x)\int_t g(t)w^T\nabla\phi_\gamma(z+tw-x)\,dt\,dx\\
&=\int_x w^T\nabla f(x)\int_t g(t)\frac{d}{dt}\phi_\gamma(z+tw-x)\,dt\,dx\\
&=-\int_x w^T\nabla f(x)\int_t g'(t)\phi_\gamma(z+tw-x)\,dt\,dx\\
&=-\int_t g'(t)\int_x \phi_\gamma(z+tw-x)w^T\nabla f(x)\,dx\,dt\\
&=-\int_t g'(t)\Big(\phi_\gamma\star(w^T\nabla f)\Big)(z+tw)\,dt\\
&\to-\int_t g'(t)w^T\nabla f(z+tw)\,dt=-\int g'h'=\int g\,d(D^2h),
\end{aligned}$$

as desired. □

In this result, the effect of the convolution is to focus attention on the line through the point z in the direction w. In informal language, we deduce that the modulus of nonconvexity $\Lambda(\delta)$ is determined by the concentration of the negative parts of the measures $w^T(D^2f)w$ around line segments of length δ in unit directions w.

For more intuition on directional distributional second derivatives of the form $w^T(D^2f)w$, let us consider a convex function $f:\mathbf{R}^2\to\mathbf{R}$. After a suitable choice of basis, we can suppose that w is the first unit vector e^1 and therefore, consider the Radon measure $(Df)_{11}$. To understand this measure, consider the integral

$$\int_{\mathbf{R}^2} r_\gamma(x_1)g(x_2)\,d(D^2f)_{11}(x)$$

for the function r_γ of Example A.2 and any test function $g:\mathbf{R}\to\mathbf{R}$. As $\gamma\downarrow 0$, we observe

$$\begin{aligned}
\int_{\mathbf{R}^2} r_\gamma(x_1)g(x_2)\,d\Big(e_1^T(D^2f)e_1\Big)(x)&=-\int_{\mathbf{R}^2}\Big(e_1^T\nabla f(x)\Big)\cdot\Big(e_1^T\nabla\Big(r_\gamma(x_1)g(x_2)\Big)\Big)\,dx\\
&=-\int_{\mathbf{R}^2}\frac{\partial f}{\partial x_1}r_\gamma'(x_1)g(x_2)\,dx_1\,dx_2\\
&\to\int\left(f'\left(\begin{bmatrix}q\\t\end{bmatrix};\begin{bmatrix}1\\0\end{bmatrix}\right)-f'\left(\begin{bmatrix}p\\t\end{bmatrix};\begin{bmatrix}1\\0\end{bmatrix}\right)\right)g(t)\,dt.
\end{aligned}$$

More generally, this argument suggests, loosely, that $w^T(D^2f)w$ measures the variation of the directional derivative $f'(\cdot;w)$ along the direction w.

Endnote

[1] We became aware of these concurrent independent works after completing the initial draft of this manuscript.

References

[1] Bianchi P, Hachem W, Schechtman S (2022) Convergence of constant step stochastic gradient descent for non-smooth non-convex functions. *Set-Valued Variational Anal.* 30(3):1117–1147.
[2] Bolte J, Pauwels E (2021) Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Math. Programming* 188(1):19–51.
[3] Bolte J, Daniilidis A, Lewis A (2009) Tame functions are semismooth. *Math. Programming* 117(1):5–19.
[4] Davis D, Drusvyatskiy D (2018) Complexity of finding near-stationary points of convex functions stochastically. Preprint, submitted February 21, https://arxiv.org/abs/1802.08556.
[5] Davis D, Drusvyatskiy D (2019) Stochastic model-based minimization of weakly convex functions. *SIAM J. Optim.* 29(1):207–239.
[6] Davis D, Drusvyatskiy D, Lee YT, Padmanabhan S, Ye G (2022) A gradient sampling method with complexity guarantees for Lipschitz functions in high and low dimensions. Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, eds. *Adv. Neural Inf. Process*, vol. 35 (Curran Associates, Inc., Red Hook, NY), 6692–6703.
[7] Dudley R (1977) On second derivatives of convex functions. *Mathematica Scandinavica* 41(1):159–174.
[8] Evans L, Gariepy R (1992) *Measure Theory and Fine Properties of Functions* (CRC Press, Boca Raton, FL).
[9] Facchinei F, Pang JS (2003) *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vol. II, Springer Series in Operations Research (Springer-Verlag, New York).
[10] Goldstein A (1977) Optimization of Lipschitz continuous functions. *Math. Programming* 13(1):14–22.
[11] Hartman P (1959) On functions representable as a difference of convex functions. *Pacific J. Math.* 9(3):707–713.
[12] Henrion R, Outrata J (2001) A subdifferential condition for calmness of multifunctions. *J. Math. Anal. Appl.* 258(1):110–130.

[13] Jordan M, Lin T, Zampetakis M (2022) On the complexity of deterministic nonsmooth and nonconvex optimization. Preprint, submitted September 26, https://arxiv.org/abs/2209.12463.
[14] Jordan M, Kornowski G, Lin T, Shamir O, Zampetakis M (2023) Deterministic nonsmooth nonconvex optimization. Gergely N, Lorenzo R, eds. *Proc. Thirty Sixth Conf. Learn. Theory, Proc. Machine Learn. Res.* 4570–4597.
[15] Kornowski G, Shamir O (2022) On the complexity of finding small subgradients in nonsmooth optimization. Preprint, submitted September 21, https://arxiv.org/abs/2209.10346.
[16] Mahdavi-Amiri N, Yousefpour R (2012) An effective nonsmooth optimization algorithm for locally Lpschitz functions. *J. Optim. Theory Appl.* 155(1):180–195.
[17] Mifflin R (1977) An algorithm for constrained optimization with semismooth functions. *Math. Oper. Res.* 2(2):191–207.
[18] Nesterov Y (2012) Lecture 1: Intrinsic complexity of black-box optimization. Accessed November 20, 2023, https://people.montefiore.uliege.be/francqui/slides/Lect1_Complexity_Boadilla.pdf.
[19] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. (2019) PyTorch: An imperative style, high-Performance deep learning library. *Adv. Neural Inf. Process*, vol. 32 (Curran Associates, Inc., Red Hook, NY), 8024–8035.
[20] Rudin W (1966) *Real and Complex Analysis* (McGraw-Hill, New York).
[21] Sagastizábal C (2018) A $\mathcal{VU}$-point of view of nonsmooth optimization. Sirakov B, de Souza PN, Viana M, eds. *Proc. Internat. Congress Math.*, vol. 4 (World Scientific, Singapore), 3815–3836.
[22] Tian L, So AMC (2021) Computing Goldstein (ϵ, δ)-stationary points of Lipschitz functions in $\tilde{O}(\epsilon^{-3}\delta^{-1})$ iterations via random conic perturbation. Accessed November 20, 2023, https://arxiv.org/pdf/2112.09002v1.pdf.
[23] Wolfe P (1975) A method of conjugate subgradients for minimizing nondifferentiable functions. Balinski ML, Wolfe P, eds. *Nondifferentiable Optimization*, Mathematical Programming Studies, vol. 3 (Springer, Berlin), 145–173.
[24] Zhang J, Lin H, Jegelka S, Sra S, Jadbabaie A (2020) Complexity of finding stationary points of nonconvex nonsmooth functions. Daumé H III, Singh A, eds. *Proc. Machine Learn. Res.*, vol. 119, 11173–11182.