

# Chapter 6

## Gradient Sampling Methods for Nonsmooth Optimization



James V. Burke, Frank E. Curtis, Adrian S. Lewis, Michael L. Overton,  
and Lucas E. A. Simões

*Dedicated to Krzysztof Kiwiel, in recognition of his fundamental work on algorithms for nonsmooth optimization*

**Abstract** This article reviews the gradient sampling methodology for solving non-smooth, nonconvex optimization problems. We state an intuitively straightforward gradient sampling algorithm and summarize its convergence properties. Throughout this discussion, we emphasize the simplicity of gradient sampling as an extension of the steepest descent method for minimizing smooth objectives. We provide an overview of various enhancements that have been proposed to improve practical performance, as well as an overview of several extensions that have been proposed in the literature, such as to solve constrained problems. We also clarify certain technical aspects of the analysis of gradient sampling algorithms, most notably related to the assumptions one needs to make about the set of points at which the objective is continuously differentiable. Finally, we discuss possible future research directions.

---

J. V. Burke

Department of Mathematics, University of Washington, Seattle, WA, USA

F. E. Curtis

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

e-mail: [frank.e.curtis@lehigh.edu](mailto:frank.e.curtis@lehigh.edu)

A. S. Lewis

School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

e-mail: [adrian.lewis@cornell.edu](mailto:adrian.lewis@cornell.edu)

M. L. Overton (✉)

Courant Institute of Mathematical Sciences, New York University, New York, NY, USA

e-mail: [mo1@nyu.edu](mailto:mo1@nyu.edu)

L. E. A. Simões

Department of Applied Mathematics, University of Campinas, Campinas, Brazil

## 6.1 Introduction

The gradient sampling (GS) algorithm is a conceptually simple descent method for solving nonsmooth, nonconvex optimization problems, yet it is one that possesses a solid theoretical foundation and has been employed to substantial success in a wide variety of applications. Since the appearance of the fundamental algorithm and its analysis little over a dozen years ago, GS has matured into a comprehensive methodology. Various enhancements have been proposed that make it a competitive approach in many NSO contexts, and it has been extended in various interesting ways, such as for NSO on manifolds and for solving constrained problems. The purpose of this work is to provide background and motivation for the development of the GS method, discuss its theoretical guarantees, and provide an overview of the enhancements and extensions that have been the subject of research over recent years.

The underlying philosophy of GS is that virtually any nonsmooth objective function of interest is differentiable almost everywhere; in particular, this is true if the objective  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is either LLC (see Theorem 1.5) or semialgebraic. In such cases, when  $f$  is evaluated at a randomly generated point  $\mathbf{x} \in \mathbb{R}^n$ , it is differentiable there with probability one. This means that an algorithm can rely on an ability to obtain the objective function value  $f(\mathbf{x})$  and gradient  $\nabla f(\mathbf{x})$ , as when  $f$  is smooth, rather than require an oracle to compute a subgradient. In most interesting settings,  $f$  is *not* differentiable at its local minimizers, but, under reasonable assumptions, the carefully crafted mechanisms of the GS algorithm generate a sequence of iterates—at which  $f$  is differentiable—converging to stationarity.

At the heart of GS is a stabilized steepest descent approach. When  $f$  is differentiable at  $\mathbf{x}$ , the negative gradient  $-\nabla f(\mathbf{x})$  is, of course, the traditional steepest descent direction for  $f$  at  $\mathbf{x}$  in the 2-norm in that

$$-\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} = \operatorname{argmin}_{\|\mathbf{d}\| \leq 1} \nabla f(\mathbf{x})^T \mathbf{d}. \quad (6.1)$$

However, when  $\mathbf{x}$  is near a point where  $f$  is not differentiable, it may be necessary to take a very short step along  $-\nabla f(\mathbf{x})$  to obtain decrease in  $f$ . It is for this reason that the traditional steepest descent method may converge to nonstationary points when  $f$  is nonsmooth.<sup>1</sup> The GS algorithm stabilizes the choice of the search direction to avoid this issue. In each iteration, a descent direction from the current iterate  $\mathbf{x}^k$  is obtained by supplementing the information provided by  $\nabla f(\mathbf{x}^k)$  with gradients evaluated at randomly generated points  $\{\mathbf{x}^{k,1}, \dots, \mathbf{x}^{k,m}\} \subset \bar{B}(\mathbf{x}^k; \epsilon_k)$ , which are *near*  $\mathbf{x}^k$ , and then computing the minimum-norm vector  $\mathbf{g}^k$  in the convex hull of

---

<sup>1</sup>Although this fact has been known for decades, most of the examples that appear in the literature are rather artificial since they were designed with exact line searches in mind. Analyses showing that the steepest descent method with inexact line searches converges to nonstationary points of some simple convex nonsmooth functions have appeared recently in [1, 22].

these gradients. This choice can be motivated by the goal, using the  $\epsilon$ -subdifferential  $\partial_\epsilon^G f(\mathbf{x})$  given in Definition 1.10, that

$$-\frac{\mathbf{g}^k}{\|\mathbf{g}^k\|} \approx \operatorname{argmin}_{\|\mathbf{d}\| \leq 1} \max_{\mathbf{g} \in \partial_\epsilon^G f(\mathbf{x})} \mathbf{g}^T \mathbf{d}; \quad (6.2)$$

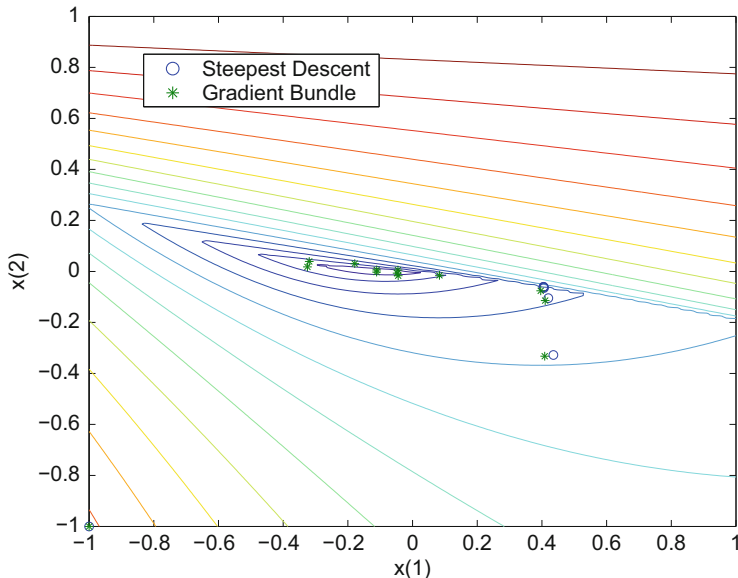
i.e.,  $-\mathbf{g}^k$  can essentially be viewed as a steepest descent direction for  $f$  from  $\mathbf{x}^k$  in a more “robust” sense. A line search is then used to find a positive stepsize  $t_k$  yielding decrease in  $f$ , i.e.,  $f(\mathbf{x}^k - t_k \mathbf{g}^k) < f(\mathbf{x}^k)$ . The sampling radius  $\epsilon_k$  that determines the meaning of “near  $\mathbf{x}^k$ ” may either be fixed or adjusted dynamically.

A specific instance of the GS algorithm is presented in Sect. 6.2. Its convergence guarantees are summarized in Sect. 6.3. We then present various enhancements and extensions of the approach in Sects. 6.4 and 6.5, respectively, followed by a discussion of some successful applications of the GS methodology in Sect. 6.6. Throughout this work, our goal is to emphasize the *simplicity* of the fundamental GS strategy. We believe that this, in addition to its strong convergence properties for locally Lipschitz optimization, makes it an attractive choice when attempting to solve difficult types of NSO problems.

Although the first convergence analysis of a GS algorithm was given by Burke, Lewis, and Overton in [8], an earlier version of the method was presented by these authors in [7]. That algorithm, originally called a “gradient bundle” method, was applied to a function that was not only nonconvex and nonsmooth, but also non-locally-Lipschitz, namely, the spectral abscissa—i.e., the largest of the real parts of the eigenvalues—of a linear matrix function  $A$  mapping a parameter vector  $\mathbf{x}$  to the space of nonsymmetric square matrices. The spectral abscissa is not locally Lipschitz at a matrix  $\bar{X}$  when an eigenvalue of  $\bar{X}$  with largest real part has multiplicity two or more [5], but it is semialgebraic and, hence, differentiable almost everywhere, so a GS algorithm was applicable. The method was surprisingly effective. As anticipated, in most cases the apparent local minimizers that were approximated had the property that the eigenvalues of  $A$  with largest real part had multiplicity two or more. An illustration that appeared in [7] is reproduced in Fig. 6.1; the extremely “steep” contours of the objective function indicate its non-Lipschitzness. Obtaining theoretical results for a GS algorithm applied to solve non-locally-Lipschitz problems seems very challenging; we discuss this issue further in Sect. 6.3.3, after describing the substantial body of theory that has been developed for the locally Lipschitz case in Sects. 6.3.1 and 6.3.2.

## 6.2 Algorithm GS

We now state a specific variant of the GS algorithm. We start by assuming only that the objective function  $f$  is locally Lipschitz over  $\mathbb{R}^n$ , which implies, by Rademacher’s theorem (see Theorem 1.5), that  $f$  is differentiable almost every-



**Fig. 6.1** Contours of the spectral abscissa of an affine matrix family given in [7]. Iterates of the ordinary gradient (“steepest descent”) method with a line search are shown (small circles) along with those of the GS (“gradient bundle”) algorithm (asterisks). Both start at  $(-1, -1)$

where. As previously mentioned, at the heart of the algorithm is the computation of a descent direction by finding the minimum norm element of the convex hull of gradients obtained about each iterate. The remaining procedures relate to the line search to obtain decrease in  $f$  and the selection of a subsequent iterate so that  $f$  is differentiable at all elements of  $\{\mathbf{x}^k\}$ .

While the essence of the methods from [7] and [8] remains intact, Algorithm **GS** differs in subtle yet important ways from the methods presented in these papers, as we now explain.

1. Algorithm **GS** incorporates a key modification proposed by Kiwiel in [28, Algorithm 2.1], namely, the second inequality in (6.4); the version in [8] used  $\epsilon_k$  instead of  $\min\{t_k, \epsilon_k\}$ . As Kiwiel explained, this minor change allowed him to drop the assumption in [8] that the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$  is compact, strengthening the convergence results for the algorithm.
2. A second change suggested in [28, Algorithm 2.1] is the introduction of the termination tolerances  $\nu_{\text{opt}}$  and  $\epsilon_{\text{opt}}$ . These were used in the computational experiments in [8], but not in the algorithm statement or analysis. Note that if  $\epsilon_{\text{opt}}$  is set to zero, then Algorithm **GS** never terminates since  $\epsilon_k$  can never be zero, though it may happen that one obtains  $\|\mathbf{g}^k\| = 0$ .
3. A third change, also suggested by Kiwiel, is the usage of the nonnormalized search direction  $-\mathbf{g}^k$  (originally used in [7]) instead of the normalized search direction  $-\mathbf{g}^k/\|\mathbf{g}^k\|$  (used in [8]). The resulting inequalities in (6.3) and (6.4) are

taken from [28, Section 4.1]. This choice does not affect the main conclusions of the convergence theory as in both cases it is established [8, 28] that the stepsize  $t_k$  can be determined by a finite process. However, since Theorem 6.1 below shows that a subsequence of  $\{\mathbf{g}^k\}$  converges to zero under reasonable conditions, one expects that fewer function evaluations should be required by the line search asymptotically when using the nonnormalized search direction, whereas using the normalized direction may result in the number of function evaluations growing arbitrarily large [28, Section 4.1]. Our practical experience is consistent with this viewpoint.

4. Another aspect of Algorithm GS that is different in both [8] and [28] concerns the randomization procedure in Step 2. In the variants given in those papers, it was stated that the algorithm terminates if  $f$  is not *continuously* differentiable at the randomly sampled points  $\{\mathbf{x}^{k,1}, \dots, \mathbf{x}^{k,m}\}$ . In the theorem stated in the next section, we require only that  $f$  is differentiable at the sampled points. Since by Rademacher's theorem and countable additivity of probabilities this holds for every sampled point with probability one, we do not include a termination condition here.
5. Finally, Steps 11–14 in Algorithm GS *do* require explicit checks that ensure that  $f$  is differentiable at  $\mathbf{x}^{k+1}$ , but unlike in the variants in [8] and [28], it is not required that  $f$  be *continuously* differentiable at  $\mathbf{x}^{k+1}$ . This differentiability requirement is included since it is not the case that  $f$  is differentiable at  $\mathbf{x}^k - t_k \mathbf{g}^k$  with probability one, as is shown via an example in [24], discussed further in Sect. 6.4.2. For a precise procedure for implementing Step 14, see [28].

The computation in Step 3 of Algorithm GS requires solving a strongly convex quadratic optimization problem (QP) to compute the minimum-norm element of the convex hull of the current and sampled gradients, or, equivalently, to compute the 2-norm projection of the origin onto this convex hull. It is essentially the same operation required in every iteration of a bundle method. To see this, observe that solving the QP in Step 3 can be expressed, with

$$G_k := [\nabla f(\mathbf{x}^k) \ \nabla f(\mathbf{x}^{k,1}) \ \dots \ \nabla f(\mathbf{x}^{k,m})],$$

as computing  $(z_k, \mathbf{d}^k, \boldsymbol{\lambda}^k) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^{m+1}$  as the primal-dual solution of the QPs

$$\left\{ \begin{array}{ll} \text{minimize} & z + \frac{1}{2} \|\mathbf{d}\|^2 \\ \text{subject to} & G_k^T \mathbf{d} \leq z \mathbf{1}, \\ & (z, \mathbf{d}) \in \mathbb{R} \times \mathbb{R}^n, \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{ll} \text{minimize} & \frac{1}{2} \|G_k \boldsymbol{\lambda}\|^2 \\ \text{subject to} & \mathbf{1}^T \boldsymbol{\lambda} = 1, \ \boldsymbol{\lambda} \geq \mathbf{0}, \\ & \boldsymbol{\lambda} \in \mathbb{R}^{m+1}. \end{array} \right. \quad (6.5)$$

The latter problem, yielding  $G_k \boldsymbol{\lambda}^k = \mathbf{g}^k$ , can easily be seen to be equivalent to solving the subproblem  $\min_{\mathbf{g} \in \mathcal{G}^k} \frac{1}{2} \|\mathbf{g}\|^2$  stated in Step 3, whereas the former problem, yielding  $\mathbf{d}^k = -\mathbf{g}^k$ , can be seen to have the same form as the subproblems arising in bundle methods.

**Algorithm GS:** Gradient sampling

---

**Data:** initial point  $\mathbf{x}^0$  at which  $f$  is differentiable, initial sampling radius  $\epsilon_0 \in (0, \infty)$ , initial stationarity target  $\nu_0 \in [0, \infty)$ , sample size  $m \geq n + 1$ , line search parameters  $(\beta, \gamma) \in (0, 1) \times (0, 1)$ , termination tolerances  $(\epsilon_{\text{opt}}, \nu_{\text{opt}}) \in [0, \infty) \times [0, \infty)$ , and reduction factors  $(\theta_\epsilon, \theta_\nu) \in (0, 1] \times (0, 1]$

**1 for**  $k \in \mathbb{N}$  **do**

**2** | independently sample  $\{\mathbf{x}^{k,1}, \dots, \mathbf{x}^{k,m}\}$  uniformly from  $\bar{B}(\mathbf{x}^k; \epsilon_k)$

**3** | compute  $\mathbf{g}^k$  as the solution of  $\min_{\mathbf{g} \in \mathcal{G}^k} \frac{1}{2} \|\mathbf{g}\|^2$ , where

$$\mathcal{G}^k := \text{conv}\{\nabla f(\mathbf{x}^k), \nabla f(\mathbf{x}^{k,1}), \dots, \nabla f(\mathbf{x}^{k,m})\};$$

**4** | **if**  $\|\mathbf{g}^k\| \leq \nu_{\text{opt}}$  and  $\epsilon_k \leq \epsilon_{\text{opt}}$ , **then**

**5** | | terminate;

**6** | **else**

**7** | | **if**  $\|\mathbf{g}^k\| \leq \nu_k$ , **then**

**8** | | | set  $\nu_{k+1} \leftarrow \theta_\nu \nu_k$ ,  $\epsilon_{k+1} \leftarrow \theta_\epsilon \epsilon_k$ , and  $t_k \leftarrow 0$ ;

**9** | | **else**

**10** | | | set  $\nu_{k+1} \leftarrow \nu_k$ ,  $\epsilon_{k+1} \leftarrow \epsilon_k$ , and

$$t_k \leftarrow \max \left\{ t \in \{1, \gamma, \gamma^2, \dots\} : \right.$$

$$\left. f(\mathbf{x}^k - t\mathbf{g}^k) < f(\mathbf{x}^k) - \beta t \|\mathbf{g}^k\|^2 \right\}; \quad (6.3)$$

**11** | | **if**  $f$  is differentiable at  $\mathbf{x}^k - t_k \mathbf{g}^k$ , **then**

**12** | | | set  $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - t_k \mathbf{g}^k$ ;

**13** | | **else**

**14** | | | set  $\mathbf{x}^{k+1}$  randomly as any point where  $f$  is differentiable such that

$$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k) - \beta t_k \|\mathbf{g}^k\|^2$$

$$\text{and } \|\mathbf{x}^k - t_k \mathbf{g}^k - \mathbf{x}^{k+1}\| \leq \min\{t_k, \epsilon_k\} \|\mathbf{g}^k\|; \quad (6.4)$$


---

Normally, the initial stationarity target  $\nu_0$  is chosen to be positive and the reduction factors  $\theta_\nu$  and  $\theta_\epsilon$  are chosen to be less than one so that the stationarity target and sampling radius are reduced every time the condition  $\|\mathbf{g}^k\| \leq \nu_k$  is satisfied. However, it is also interesting to consider the variant with  $\nu_0 = 0$  and  $\theta_\epsilon = 1$ , forcing the algorithm to run forever with  $\epsilon$  fixed unless it terminates with  $\mathbf{g}^k = \mathbf{0}$  for some  $k \in \mathbb{N}$ . We consider both of these variants in the global convergence theory given in the next section.

### 6.3 Convergence Theory for Algorithm GS

Any NSO method seeks to find a point  $\mathbf{x}$  that is Clarke stationary (such that  $\mathbf{0} \in \partial f(\mathbf{x})$ ; see (1.7), Definition 1.8 and Theorem 1.2) or at least  $\epsilon$ -stationary (such that  $\mathbf{0} \in \partial_\epsilon^G f(\mathbf{x})$ ; see Definition 1.10). For all practical purposes, one cannot generally evaluate  $\partial f$  (or  $\partial_\epsilon^G f$ ) at (or near) any point where  $f$  is not differentiable. That said, Algorithm GS is based on the idea that one can approximate the minimum norm element in  $\partial_\epsilon^G f(\mathbf{x}^k)$  through random sampling of gradients in the ball  $\bar{B}(\mathbf{x}^k; \epsilon)$ . To a large extent this idea is motivated by [6] which investigates how well the entire Clarke subdifferential  $\partial f(\mathbf{x})$  can be approximated through random sampling. However, the results in [6] cannot be directly exploited in the analysis of the GS algorithm because the gradients are sampled only at a finite number of points near any given iterate.

One might wonder why Algorithm GS involves the use of  $m \geq n + 1$  gradients at randomly sampled points in addition to the gradient at the current iterate. Loosely speaking, this lower bound on the number of gradients is required in the analysis to ensure that one can use Carathéodory's theorem to argue that there is a sufficiently good chance that  $-\mathbf{g}^k$  either (1) is a sufficiently good search direction from  $\mathbf{x}^k$ , or (2) is small enough to indicate that the current iterate is *approximately*  $\epsilon^k$ -stationary. In the latter case, the analysis actually involves checking for  $\epsilon^k$ -stationarity of a point *near*  $\mathbf{x}^k$ , not of  $\mathbf{x}^k$  itself, which explains why  $n + 1$  gradients are needed *in addition to*  $\nabla f(\mathbf{x}^k)$ . It remains an open question whether it would suffice to sample gradients at just  $n$  randomly sampled points in addition to the gradient at  $\mathbf{x}^k$ ; however, such a possibility is not of much consequence, especially if one employs the *adaptive sampling* enhancement discussed in Sect. 6.4.3.

#### 6.3.1 Global Convergence Guarantees

A critical aspect of theoretical convergence guarantees for Algorithm GS concerns the set of points where  $f$  is *continuously* differentiable, which we denote by  $D$ . Consideration of  $D$  played a crucial role in the analysis in both [8] and [28], but there were some oversights concerning both the requirements of the algorithm with respect to  $D$  and the assumptions on  $D$ . Regarding the requirements of the algorithm with respect to  $D$ , there is actually no need, from a theoretical point of view, for either the iterates  $\{\mathbf{x}^k\}$  or the randomly generated sampled points  $\{\mathbf{x}^{k,j}\}$  to lie in  $D$ ; all that is needed is that  $f$  is differentiable at these points. Most implementations of GS algorithms do not attempt to check any form of differentiability in any case, but if one were to attempt to implement such a check, it is certainly more tractable to check for differentiability than continuous differentiability. Regarding the assumptions on  $D$ , in the theorems that we state below, we assume that  $D$  is an open set with full measure in  $\mathbb{R}^n$ . In contrast, the relevant assumption stated in [8, 28] is weaker, namely, that  $D$  is an open dense subset of  $\mathbb{R}^n$ . However, the proofs

of convergence actually require the full measure assumption on  $D$  that we include below.<sup>2</sup>

There are three types of global convergence guarantees of interest for Algorithm **GS**: one when the input parameters ensure that  $\{\epsilon_k\} \downarrow 0$ , one when  $\epsilon_k$  is repeatedly reduced but a positive stopping tolerance prevents it from converging to zero, and one when  $\epsilon_k = \epsilon > 0$  for all  $k$ . These lead to different properties for the iterate sequence. The first theorem below relates to cases when the stationarity tolerance and sampling radius tolerance are both set to zero so that the algorithm can never terminate.

**Theorem 6.1** *Suppose that  $f$  is locally Lipschitz on  $\mathbb{R}^n$  and continuously differentiable on an open set  $D$  with full measure in  $\mathbb{R}^n$ . Suppose further that Algorithm **GS** is run with  $\nu_0 > 0$ ,  $\nu_{\text{opt}} = \epsilon_{\text{opt}} = 0$ , and strict reduction factors  $\theta_\nu < 1$  and  $\theta_\epsilon < 1$ . Then, with probability one, Algorithm **GS** is well defined in the sense that the gradients in Step 3 exist in every iteration, the algorithm does not terminate, and either*

- (i)  $f(\mathbf{x}^k) \downarrow -\infty$ ; or
- (ii)  $\nu_k \downarrow 0$ ,  $\epsilon_k \downarrow 0$ , and every cluster point of  $\{\mathbf{x}^k\}$  is Clarke stationary for  $f$ .

Theorem 6.1 is essentially the same as [28, Theorem 3.3] (with the modifications given in [28, Section 4.1] for nonnormalized directions), except for two aspects:

1. The proof given in [28] implicitly assumes that  $D$  is an open set with full measure, as does the proof of [8, Theorem 3.4] on which Kiwiel's proof is based, although the relevant assumption on  $D$  in both papers is the weaker condition that  $D$  is an open dense set. Details are given in Appendix 1.
2. In the algorithms analyzed in [28] and [8], the iterates  $\{\mathbf{x}^k\}$  and the randomly sampled points  $\{\mathbf{x}^{k,j}\}$  were enforced to lie in the set  $D$  where  $f$  is continuously differentiable. We show in Appendix 2 that the theorem still holds when this requirement is relaxed to ensure only that  $f$  is differentiable at these points.

As Kiwiel argues, Theorem 6.1 is essentially the best result that could be expected. Furthermore, as pointed out in [28, Remark 3.7(ii)], it leads immediately to the following corollary.

**Corollary 6.1** *Suppose that  $f$  is locally Lipschitz on  $\mathbb{R}^n$  and continuously differentiable on an open set  $D$  with full measure in  $\mathbb{R}^n$ . Suppose further that Algorithm **GS** is run with  $\nu_0 > \nu_{\text{opt}} > 0$ ,  $\epsilon_0 > \epsilon_{\text{opt}} > 0$ , and strict reduction factors  $\theta_\nu < 1$  and  $\theta_\epsilon < 1$ . Then, with probability one, Algorithm **GS** is well defined in the sense that the gradients in Step 3 exist at every iteration, and either*

- (i)  $f(\mathbf{x}^k) \downarrow -\infty$ ; or
- (ii) Algorithm **GS** terminates by the stopping criteria in Step 4.

---

<sup>2</sup>This oversight went unnoticed for 12 years until J. Portegies and T. Mitchell brought it to our attention recently.



The final result that we state concerns the case when the sampling radius is fixed. A proof of this result is essentially given by that of [28, Theorem 3.5], again taking into account the comments in the Appendices.

**Theorem 6.2** *Suppose that  $f$  is locally Lipschitz on  $\mathbb{R}^n$  and continuously differentiable on an open set  $D$  with full measure in  $\mathbb{R}^n$ . Suppose further that Algorithm GS is run with  $v_0 = v_{\text{opt}} = 0$ ,  $\epsilon_0 = \epsilon_{\text{opt}} = \epsilon > 0$ , and  $\theta_\epsilon = 1$ . Then, with probability one, Algorithm GS is well defined in the sense that the gradients in Step 3 exist at every iteration, and one of the following occurs:*

- (a)  $f(\mathbf{x}^k) \downarrow -\infty$ ; or
- (b) Algorithm GS terminates for some  $k \in \mathbb{N}$  with  $\mathbf{g}^k = \mathbf{0}$ ; or
- (c) there exists  $\mathcal{K} \subseteq \mathbb{N}$  with  $\{\mathbf{g}^k\}_{k \in \mathcal{K}} \rightarrow \mathbf{0}$  and every cluster point of  $\{\mathbf{x}^k\}_{k \in \mathcal{K}}$  is  $\epsilon$ -stationary for  $f$ .

Of the five open questions regarding the convergence analysis for GS raised in [8], three were answered explicitly by Kiwiel in [28]. Another open question was: ‘‘Under what conditions can one guarantee that the GS algorithm terminates finitely?’’ This was posed in the context of a fixed sampling radius and therefore asks how one might know whether outcome (b) or (c) occurs in Theorem 6.2, assuming  $f$  is bounded below. This remains open, but Kiwiel’s introduction of the termination tolerances in the GS algorithm statement led to Corollary 6.1 which guarantees that when the sampling radius is reduced dynamically and the tolerances are nonzero, Algorithm GS must terminate if  $f$  is bounded below. The only other open question concerns extending the convergence analysis to the non-Lipschitz case.

Overall, Algorithm GS has a very satisfactory convergence theory in the locally Lipschitz case. Its main weakness is its per-iteration cost, most notably due to the need to compute  $m \geq n + 1$  gradients in every iteration and solve a corresponding QP. However, enhancements to the algorithm have been proposed that can vastly reduce this per-iteration cost while maintaining these guarantees. We discuss these and other enhancements in Sect. 6.4.

### 6.3.2 A Local Linear Convergence Result

Given the relationship between GS and a traditional steepest descent approach, one might ask if there are scenarios in which Algorithm GS can attain a linear rate of local convergence. The study in [25] answers this in the affirmative, at least in a certain probabilistic sense. If (1) the set of sampled points is *good* in a certain sense described in [25], (2) the objective function  $f$  belongs to a class of functions defined by the maximum of a finite number of smooth functions (‘‘finite-max’’ functions), and (3) the input parameters are set appropriately, then Algorithm GS will produce a step yielding a reduction in  $f$  that is significant. This analysis involves  $\mathcal{WU}$ -decomposition ideas [31, 32, 36], where in particular it is shown that the reduction in  $f$  is comparable to that achieved by a steepest descent method restricted to the

smooth  $\mathcal{U}$ -space of  $f$ . This means that a linear rate of local convergence can be attained over any infinite subsequence of iterations in which the sets of sampled points are *good*.

### 6.3.3 The Non-Lipschitz Case

In the non-locally-Lipschitz case, the Clarke subdifferential  $\partial f$  is defined in [6, p. 573]; unlike in the Lipschitz case, this set may be unbounded, presenting obvious difficulties for approximating it through random sampling of gradients. In fact, more than half of [6] is devoted to investigating this issue, relying heavily on modern variational analysis as expounded in [41]. Some positive results were obtained, specifically in the case that  $f$  is “directionally Lipschitz” at  $\bar{x}$ , which means that the “horizon cone” [6, p. 572] of  $f$  at  $\bar{x}$  is pointed, that is, it does not contain a line. For example, this excludes the function on  $\mathbb{R}$  defined by  $f(x) = |x|^{1/2}$  at  $\bar{x} = 0$ , but it applies to the case  $f(x) = (\max\{0, x\})^{1/2}$  even at  $\bar{x} = 0$ . The discussion of the directionally Lipschitz case culminates with Corollary 6.1, which establishes that the Clarke subdifferential can indeed be approximated by convex hulls of gradients. On the more negative side, Example 7.2 shows that this approximation can fail badly in the general Lipschitz case. Motivated by these results, Burke and Lin have recently extended the GS convergence theory to the directionally Lipschitz case [4, 34]. However, it would seem difficult to extend these results to the more general non-Lipschitz case.

Suppose  $f : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  is defined by

$$f(X) = \max\{\operatorname{Re}\lambda : \det(\lambda I - X) = 0\},$$

the spectral abscissa (maximum of the real parts of the eigenvalues) of  $X$ . Assume that  $\bar{X}$  has the property that its only eigenvalues whose real parts coincide with  $f$  make up a zero eigenvalue with multiplicity  $q$  associated with a single Jordan block (the generic case). In this case the results in [5] tell us that the horizon cone of  $f$  is pointed at  $\bar{X}$  if and only if the multiplicity  $q \leq 2$ ; on the other hand  $f$  is locally Lipschitz at  $X$  if and only if  $q = 1$ . In the light of the previous paragraph, one might expect much greater practical success in applying GS to minimize the spectral abscissa of a parameterized matrix if the optimal multiplicities are limited to 1 or 2. However, this seems not to be the case. The results reported in [7] for unconstrained spectral abscissa minimization, as well as results for applying the algorithm of [12] (see Sect. 6.5.2 below) for constrained nonsmooth, nonconvex optimization to problems with spectral radius objective and constraints, as reported in [15, Section 4.2 and Appendix A.1], do not show any marked deterioration as the optimal multiplicities increase from 2 or 3, although certainly the problems are much more challenging for larger multiplicities. We view understanding the rather remarkably good behavior of the GS algorithm on such examples as a potentially rewarding, though certainly challenging, line of investigation.

## 6.4 Enhancements

As explained above, the statement of Algorithm **GS** differs in several ways from the algorithms stated in [7, 8], and [28]. Other variants of the strategy have also been proposed in recent years, in some cases to pose new solutions to theoretical obstacles (such as the need, in theory, to check for differentiability of  $f$  at each new iterate), and in others to enhance the practical performance of the approach. In this section, we discuss a few of these enhancements.

### 6.4.1 Restricting the Line Search to Within a Trust Region

Since the gradient information about  $f$  is obtained only within the ball  $\bar{B}(\mathbf{x}^k; \epsilon_k)$  for all  $k \in \mathbb{N}$ , and since one might expect that smaller steps should be made when the sampling radius is small, an argument can be made that the algorithm might benefit by restricting the line search to within the ball  $\bar{B}(\mathbf{x}^k; \epsilon_k)$  for all  $k \in \mathbb{N}$ . In [28, Section 4.2], such a variant is proposed where in place of  $-\mathbf{g}^k$  the search direction is set as  $-\epsilon_k \mathbf{g}^k / \|\mathbf{g}^k\|$ . With minor corresponding changes to conditions (6.3) and (6.4), all of the theoretical convergence guarantees of the algorithm are maintained. Such a variant with the trust region radius defined as a positive multiple of the sampling radius  $\epsilon_k$  for all  $k \in \mathbb{N}$  would have similar properties. This variant might perform well in practice, especially in situations when otherwise setting the search direction as  $-\mathbf{g}^k$  would lead to significant effort being spent in the line search.

### 6.4.2 Avoiding the Differentiability Check

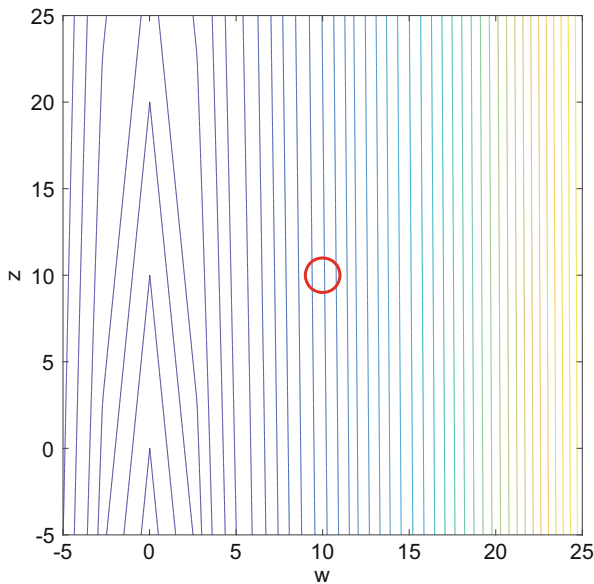
The largest distraction from the fundamentally simple nature of Algorithm **GS** is the procedure for choosing a perturbed subsequent iterate if  $f$  is not differentiable at  $\mathbf{x}^k - t_k \mathbf{g}^k$ ; see Steps 11–14. This procedure is necessary for the algorithm to be well defined since, to ensure that  $-\mathbf{g}^k$  is a descent direction for all  $k \in \mathbb{N}$ , the algorithm relies on the existence of and ability to compute  $-\nabla f(\mathbf{x}^k)$  for all  $k \in \mathbb{N}$ . One might hope that this procedure, while necessary for theoretical convergence guarantees, could be ignored in practice. However, due to the deterministic nature of the line search, situations exist in which landing on a point of nondifferentiability of  $f$  occurs with positive probability.

*Example 6.1* Consider the function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  given by

$$f(w, z) = \max\{0.5w^2 + 0.1z, w + 0.1z + 1, -w + 0.1z + 1, -0.05z - 50\};$$

see Fig. 6.2. As shown by Helou, Santos, and Simões in [24], if Algorithm GS is initialized at  $\mathbf{x}^0 = (w_0, z_0)$  chosen anywhere in the unit ball centered at  $(10, 10)$ , then there is a positive probability that the function  $f$  will not be differentiable at  $\mathbf{x}^0 - t_0 \mathbf{g}^0$ . This can be explained as follows. At any point in the unit ball centered at  $(10, 10)$ , the function  $f$  is continuously differentiable and  $\nabla f(\mathbf{x}^0) = (w_0, 0.1)$ . Moreover, there is a positive probability that the sampled points obtained at this first iteration will yield  $\mathbf{g}^0 = \nabla f(\mathbf{x}^0)$ . Therefore, given a reasonable value for the parameter  $\beta$  that appears in (6.3) (e.g.,  $\beta = 10^{-4}$ ), the sufficient decrease of the function value is attained with  $t_0 = 1$ . This guarantees that the function  $f$  will not be differentiable at the next iterate, since the first coordinate of  $\mathbf{x}^1 = (w_1, z_1)$  will be zero.

The authors of [24] propose two strategies to avoid the issues highlighted by this example. The first is that, rather than perturb the iterate after the line search, one could perturb the search direction before the line search. It is shown that if the random perturbation of the search direction is sufficiently small such that, for one thing, the resulting direction is still one of descent for  $f$ , then  $f$  will be differentiable at all iterates with probability one. Their second proposed strategy involves the use of a nonmonotone line search. In particular, it is shown that if a strictly positive value  $\Delta_k$  is added on the right-hand side of the sufficient decrease condition in (6.3) such



**Fig. 6.2** Contours of a function illustrating the necessity of the differentiability check in Algorithm GS. Initialized uniformly within the illustrated ball, there is a positive probability that  $\mathbf{x}^0 - t_0 \mathbf{g}^0 \notin D$

that  $\{\Delta_k\}$  is summable, then one can remove  $\nabla f(\mathbf{x}^k)$  from the set  $\mathcal{G}_k$  for all  $k \in \mathbb{N}$  and maintain convergence guarantees even when  $f$  is *not* differentiable at  $\{\mathbf{x}^k\}$ . This can be shown by noticing that, due to the positive term  $\Delta_k$ , the line search procedure continues to be well defined even if  $-\mathbf{g}^k$  is not a descent direction for  $f$  at  $\mathbf{x}^k$  (which may happen since  $\nabla f(\mathbf{x}^k)$  is no longer involved in the computation of  $\mathbf{g}^k$ ). However, the summability of  $\{\Delta_k\}$  implies that the possible increases in  $f$  will be finite and, when the sampled points are good enough to describe the local behavior of  $f$  around the current iterate, the function value will necessarily decrease if the method has not reached (approximate) stationarity. Overall, the sufficient reductions in  $f$  achieved in certain iterations will ultimately exceed any increases.

Another proposal for avoiding the need to have  $f$  differentiable at  $\mathbf{x}^k$  for all  $k \in \mathbb{N}$  is given in [28, Section 4.3], wherein a technique is proposed for using a limited line search, potentially causing the algorithm to take null steps in some iterations. In fact, this idea of using a limited line search can also be used to avoid the need to sample a new set of  $m \geq n + 1$  gradients in each iteration, as we discuss next.

### 6.4.3 Adaptive Sampling

As previously mentioned, the main weakness of Algorithm **GS** is the cost of computing  $m \geq n + 1$  gradients in every iteration and solving a corresponding QP. Indeed, in many situations in practice, the sampling of such a large number of gradients in each iteration can lead to a significant amount of wasted computational effort; e.g., this is the case if a productive step would have been produced if fewer gradients were computed and used in the step computation. To take advantage of such situations, one can instead sample *adaptively*, attempting to search along directions computed using fewer gradients and proceeding as long as a sufficient reduction is attained.

In [13], Curtis and Que show how such an adaptive sampling strategy can be employed so that the convergence guarantees of Algorithm **GS** are maintained while only a constant number (independent of  $n$ ) of gradients need to be sampled in each iteration. A key aspect that allows one to maintain these guarantees is the employment of a limited line search, as first proposed in [28], potentially leading to a null step when fewer than  $n + 1$  gradients are currently in hand and when the line search is not successful after a prescribed finite number of function evaluations. See also [14] for further development of these ideas, where it is shown that one might not need to sample *any* gradients as long as a sufficient reduction is attained.

The work in [13] also introduces the idea that, when adaptive sampling is employed, the algorithm can exploit a practical feature commonly used in bundle methods. This idea relates to warm-starting the algorithm for solving the QP subproblems. In particular, suppose that one has solved the primal-dual pair of QPs in (6.5) for some  $m \in \mathbb{N}$  to obtain  $(z_k, \mathbf{d}^k, \boldsymbol{\lambda}^k)$ , yielding  $\mathbf{g}^k = -\mathbf{d}^k$ . If one were to

subsequently aim to solve the pair of QPs corresponding to the augmented matrix of gradients

$$\bar{G}_k = [G_k \nabla f(\mathbf{x}^{k,m+1}) \dots \nabla f(\mathbf{x}^{k,m+p})],$$

then one obtains a viable feasible starting point for the latter QP in (6.5) by augmenting the vector  $\lambda^k$  with  $p$  zeros. This can be exploited, e.g., in an active-set method for solving this QP; see [27].

As a further practical enhancement, the work in [13, 14] also proposes the natural idea that, after moving to a new iterate  $\mathbf{x}^k$ , gradients computed in previous iterations can be “reused” if they correspond to points that lie within  $\bar{B}(\mathbf{x}^k; \epsilon_k)$ . This may further reduce the number of sampled points needed in practice.

#### 6.4.4 Second Order-Type Variants

The solution vector  $\mathbf{d}^k$  of the QP in (6.5) can be viewed as the minimizer of the model of  $f$  at  $\mathbf{x}^k$  given by

$$q_k(\mathbf{d}) = f(\mathbf{x}^k) + \max_{\mathbf{g} \in \mathcal{G}_k} \mathbf{g}^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T H_k \mathbf{d}$$

with  $H_k = I$ , the identity matrix. As in other second order-type methods for nonlinear optimization, one might also consider algorithm variants where  $H_k$  is set to some other symmetric positive definite matrix. Ideas of this type have been explored in the literature. For example, in [13], two techniques are proposed: one in which  $H_k$  is set using a quasi-Newton updating strategy and one in which the matrix is set in an attempt to ensure that the model  $q_k$  represents an upper bounding model for  $f$ . The idea of employing a quasi-Newton approach, inspired by the success of quasi-Newton methods in practice for NSO (see [33]), has also been explored further in [14, 16].

Another approach, motivated by the encouraging results obtained when employing spectral gradient methods to solve smooth [3, 21] and nonsmooth [11] optimization problems, has been to employ a Barzilai-Borwein (BB) strategy for computing initial stepsizes in a GS approach; see [35] and the background in [2, 38, 39]. Using a BB strategy can be viewed as choosing  $H_k = \alpha_k I$  for all  $k \in \mathbb{N}$  where the scalar  $\alpha_k$  is set according to iterate and gradient displacements in the latest iteration.

In all of these second order-type approaches, one is able to maintain convergence guarantees of the algorithm as long as the procedure for setting the matrix  $H_k$  is safeguarded during the optimization process. For example, one way to maintain guarantees is to restrict each  $H_k$  to the set of symmetric matrices whose eigenvalues are contained within a fixed positive interval. One might also attempt to exploit the self-correcting properties of BFGS updating; see [16].

## 6.5 Extensions

In this section, we discuss extensions to the GS methodology to solve classes of problems beyond unconstrained optimization on  $\mathbb{R}^n$ .

### 6.5.1 Riemannian GS for Optimization on Manifolds

Hosseini and Uschmajew in [26] have extended the GS methodology for minimizing a locally Lipschitz  $f$  over a set  $\mathcal{M}$ , where  $\mathcal{M}$  is a complete Riemannian manifold of dimension  $n$ . The main idea of this extension is to employ the convex hull of gradients from tangent spaces at randomly sampled points *transported* to the tangent space of the current iterate. In this manner, the algorithm can be characterized as a generalization of the Riemannian steepest descent method just as GS is a generalization of traditional steepest descent. Assuming that the vector transport satisfies certain assumptions, including a *locking condition*, the algorithm attains convergence guarantees on par with those for Algorithm GS.

### 6.5.2 SQP-GS for Constrained Optimization

Curtis and Overton in [12] proposed a combination sequential quadratic programming (SQP) and GS method for solving constrained optimization problems involving potentially nonsmooth constraint functions, i.e.,

$$\begin{cases} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) \leq \mathbf{0}, \\ & \mathbf{x} \in \mathbb{R}^n, \end{cases} \quad (6.6)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $\mathbf{c} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_c}$  are locally Lipschitz. A key aspect of the proposed SQP-GS approach is that sampled points for the objective and each individual constraint function are generated independently. With this important feature, it is shown that the algorithm, which follows a penalty-SQP strategy (e.g., see [20]), attains convergence guarantees for minimizing an exact penalty function that are similar to those in Sect. 6.3.1. Moreover, with the algorithm's penalty parameter updating strategy, it is shown that either the penalty function is driven to  $-\infty$ , the penalty parameter settles at a finite value and any limit point will be feasible for the constraints and stationary for the penalty function, or the penalty parameter will be driven to zero and any limit point of the algorithm will be stationary for a constraint violation measure. As for other exact penalty function methods for nonlinear optimization, one can translate between guarantees

for minimizing the exact penalty function and solving the constrained problem (6.6); in particular, if the problem is *calm* [10, 40], then at any local minimizer  $\mathbf{x}_*$  of (6.6) there exists a threshold for the penalty parameter beyond which  $\mathbf{x}_*$  will be a local minimizer of the penalty function.

Tang, Liu, Jian, and Li have also proposed in [42] a *feasible* variant of the SQP-GS method in which the iterates are forced to remain feasible for the constraints and the objective function is monotonically decreasing throughout the optimization process. This opens the door to employing a two-phase approach common for solving some optimization problems, where phase 1 is responsible for attaining a feasible point and phase 2 seeks optimality while maintaining feasibility.

### 6.5.3 Derivative-Free Optimization

Given its simple nature, GS has proved to be an attractive basis for the design of new algorithms even when gradient information cannot be computed explicitly. Indeed, there have been a few variants of *derivative-free* algorithms that have been inspired by GS.

The first algorithm for derivative-free optimization inspired by GS was proposed by Kiwiel in [29]. In short, in place of the gradients appearing in Algorithm GS, this approach employs Gupal's estimates of gradients of the Steklov averages of  $f$ . In this manner, function values only—specifically,  $\mathcal{O}(mn)$  per iteration—are required for convergence guarantees. A less expensive *incremental* version is also proposed.

Another derivative-free variant of GS, proposed by Hare and Nutini in [23], is specifically designed for minimizing finite-max functions. This approach exploits knowledge about which of these functions are *almost active*—in terms of having value close to the objective function—at a particular point. In so doing, rather than attempt to approximate gradients at nearby points, as in done in [29], this approach only attempts to approximate gradients of almost active functions. The convergence guarantees proved for the algorithm are similar to those for GS methods, though the practical performance is improved by the algorithm's tailored gradient approximation strategy.

Finally, we mention the *manifold sampling* algorithm, proposed by Larson, Menickelly, and Wild in [30], for solving nonconvex problems where the objective function is the  $L_1$ -norm of a smooth vector function  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^r$ . While this approach does not employ a straightforward GS methodology in that it does not randomly sample points, it does employ a GS-type approach in the way that the gradient of a model of the objective function is constructed by solving a QP of the type in (6.5). Random sampling can be avoided in this construction since the algorithm can exploit knowledge of the signs of the elements of  $\mathbf{F}(\mathbf{x})$  at any  $\mathbf{x} \in \mathbb{R}^n$  along with knowledge of  $\partial \|\cdot\|_1$ .



## 6.6 Applications

We mentioned in the introduction that the original GS paper [7] reported results for spectral abscissa optimization problems that had not been solved previously. The second GS paper [8] reported results for many more applications that again had not been solved previously: these included Chebyshev approximation by exponential sums, eigenvalue product minimization for symmetric matrices, spectral and pseudospectral abscissa minimization, maximization of the “distance to instability”, and fixed order controller design by static output feedback.

Subsequently, the GS algorithm played a key role in the HANSO (hybrid algorithm for nonsmooth optimization)<sup>3</sup> and HIFOO (H-infinity fixed order optimization)<sup>4</sup> toolboxes. The former is a stand-alone code for unconstrained NSO while the latter is a more specialized code used for the design of low order controllers for linear dynamical systems with input and output, computing fixed order controllers by optimizing stability measures that are generally nonsmooth at local minimizers [9]. HIFOO calls HANSO to carry out the optimization. The use of “hybrid” in the expansion of the HANSO acronym indicated that, from its inception, HANSO combined the use of both a quasi-Newton algorithm (BFGS) and GS. The quasi-Newton method was used in an initial phase which, rather surprisingly, typically worked very effectively even in the presence of nonsmoothness, very often providing a fast way to approximate a local minimizer. This was followed by a GS phase to refine the approximation, typically verifying a loose measure of local optimality. The HIFOO toolbox has been used successfully in a wide variety of applications, including synchronization of heterogeneous multi-agent systems and networks, design of motorized gimbals that stabilize an angular motion of an optical payload around an axis, flight control via static output feedback, robust observer-based fault detection and isolation, influence of tire damping on control of quarter-car suspensions, flexible aircraft lateral flight dynamic control, optimal control of aircraft with a blended wing body, vibration control of a fluid/plate system, controller design of a nose landing gear steering system, bilateral teleoperation for minimally invasive surgery, design of an aircraft controller for improved gust alleviation and passenger comfort, robust controller design for a proton exchange membrane fuel cell system, design of power systems controllers, and design of winding systems for elastic web materials—for a full list of references, see [15].

The successful use of BFGS in HANSO and HIFOO led to papers on the use of quasi-Newton methods in the nonsmooth context, both for unconstrained [33] and constrained [15] optimization. The latter paper introduced a new BFGS-SQP method for nonsmooth constrained optimization and compared it with the SQP-GS method discussed in Sect. 6.5.2 on a suite of challenging static output feedback

---

<sup>3</sup>[www.cs.nyu.edu/overton/software/hanso/](http://www.cs.nyu.edu/overton/software/hanso/).

<sup>4</sup>[www.cs.nyu.edu/overton/software/hifoo/](http://www.cs.nyu.edu/overton/software/hifoo/).

controller design problems, half of them non-Lipschitz (spectral radius minimization) and half of them locally Lipschitz (pseudospectral radius minimization). It was found that although the BFGS-SQP method was much faster than SQP-GS, nonetheless, if the latter method based on GS was allowed sufficient running time, it frequently found better approximate solutions than the former method based on BFGS in a well defined sense, evaluated using “relative minimization profiles”. Interestingly, this was particularly pronounced on the non-Lipschitz problems, despite the fact that the GS convergence theory does not extend to this domain. See Sect. 6.3.3 for further discussion of this issue.

Finally, we mention an interesting application of GS to robot path planning [43]. This work is based on the observation that shortest paths generated through gradient descent on a value function have a tendency to chatter and/or require an unreasonable number of steps to synthesize. The authors demonstrate that the GS algorithm can largely alleviate this problem. For systems subject to state uncertainty whose state estimate is tracked using a particle filter, they proposed the GS with particle filter (GSPF) algorithm, which uses the particles as the locations in which to sample the gradient. At each step, the GSPF efficiently finds a consensus direction suitable for all particles or identifies the type of stationary point on which it is stuck. If the stationary point is a minimum, the system has reached its goal (to within the limits of the state uncertainty) and the algorithm terminates; otherwise, the authors propose two approaches to find a suitable descent direction. They illustrated the effectiveness of the GSPF on several examples using well known robot simulation environments. This work was recently extended and modified in [19], where the practical effectiveness of both the GSPF algorithm and the new modification was demonstrated on a Segway robotic mobility platform.

## 6.7 Conclusion and Future Directions

Gradient sampling is a conceptually straightforward approximate steepest descent method. With a solid convergence theory, the method has blossomed into a powerful methodology for solving nonsmooth minimization problems. The theme of our treatment of GS in this work has been to emphasize the fact that, even though the basic algorithm has been enhanced and extended in various ways, the foundation of the approach is fundamentally simple in nature.

We have also corrected an oversight in the original GS theory (i.e., that the convergence results depend on assuming that the set of points over which the Lipschitz function  $f$  is continuously differentiable has full measure, although we do not have a counterexample to convergence of GS in the absence of this assumption). At the same time we have loosened the requirements of the algorithm (showing that  $f$  need only be differentiable at the iterates and sampled points). An open question that still remains is whether one can extend the GS theory to broader function classes, such as the case where  $f$  is assumed to be semi-algebraic but not necessarily locally Lipschitz or directionally Lipschitz.

Opportunities for extending GS theory for broader function classes may include connecting the algorithm to other randomized/stochastic optimization methods. For example, one might view the algorithm as a stochastic-gradient-like method applied to a smoothed objective. (A similar philosophy underlies the analysis by Nesterov and Spokoiny in [37]). More precisely, given a locally Lipschitz objective  $f$ , consider a smoothing  $f_\epsilon$  whose value at any point  $\mathbf{x}$  is given by the mean value of  $f$  over the ball  $B(\mathbf{x}; \epsilon)$ . The GS algorithm uses gradients of  $f$  at uniformly distributed random points in this ball. Notice that each such gradient can also be viewed as a stochastic gradient for the smoothing  $f_\epsilon$  in the sense that its expectation is the gradient of  $f_\epsilon$  at  $\mathbf{x}$ . Thus, one might hope to prove convergence results for a GS algorithm (with predetermined stepsizes rather than line searches) that parallel convergence theory for stochastic gradient methods. Recent work by Davis, Drusvyatskiy, Kakade and Lee [18] gives convergence results for stochastic *subgradient* methods on a broad class of problems.

Another potentially interesting connection is with the work of Davis and Drusvyatskiy [17] on stochastic model-based optimization. Consider a GS variant that successively minimizes stochastic models of the objective function  $f$ , where we assume for simplicity that  $f$  is a globally Lipschitz convex function. In this variant, rather than moving along the direction  $-\mathbf{g}^k$ , consider instead the construction of a cutting plane approximation of  $f$  from its affine minorants at the current iterate  $\mathbf{x}^k$  and the sampled points  $\{\mathbf{x}^{k,i}\}$ , augmented by the proximal term  $\beta_k \|\mathbf{x} - \mathbf{x}^k\|^2$ , where  $\{\beta_k\}$  is a predetermined sequence. Suppose that the next iterate is chosen as the minimizer of this model; for a given  $k$  and with  $\beta_k = 1$ , by Eq. (6.5), this scheme and GS produce similar descent directions as the sampling radius tends to zero. It follows from the results of [17] that the expected norm of the gradient of the Moreau envelope [17, p. 6] is reduced below  $\epsilon$  in  $\mathcal{O}(\epsilon^{-4})$  iterations. In fact, the assumptions on  $f$  in [17] are substantially weaker than convexity, and do not require any property of the set on which  $f$  is continuously differentiable.

Connecting the convergence theory for GS to stochastic methods as suggested in the previous two paragraphs could be enlightening. However, while stochastic methods are often designed for settings in which it is intractable to compute function values exactly—a feature reflected in the fact that the analyses for such methods are based on using predetermined stepsize sequences—the GS methodology has so far been motivated by problems for which functions and gradients are tractable to compute. In such settings, the line search in Algorithm GS is an ingredient that is crucial to its practical success.

**Acknowledgements** The authors would like to acknowledge the following financial support. J.V. Burke was supported in part by the U.S. National Science Foundation grant DMS-1514559. F.E. Curtis was supported in part by the U.S. Department of Energy grant DE-SC0010615. A.S. Lewis was supported in part by the U.S. National Science Foundation grant DMS-1613996. M.L. Overton was supported in part by the U.S. National Science Foundation grant DMS-1620083. L.E.A. Simões was supported in part by the São Paulo Research Foundation (FAPESP), Brazil, under grants 2016/22989-2 and 2017/07265-0.

## Appendix 1

This appendix is devoted to justifying the requirement that  $D$ , the set of points on which the locally Lipschitz function  $f$  is continuously differentiable, must be an open full-measure subset of  $\mathbb{R}^n$ , instead of the original assumption in [8] that  $D$  should be an open and dense set in  $\mathbb{R}^n$ .

There are two ways in which the analyses in [8, 28] actually depend on  $D$  having full measure:

1. The most obvious is that both papers require that the points sampled in each iteration should lie in  $D$ , and a statement is made in both papers that this occurs with probability one, but this is not the case if  $D$  is assumed only to be an open dense subset of  $\mathbb{R}^n$ . However, as already noted earlier and justified in [Appendix 2](#), this requirement can be relaxed, as in [Algorithm GS](#) given in [Sect. 6.2](#), to require only that  $f$  be differentiable at the sampled points.
2. The set  $D$  must have full measure for [Property 6.1](#), stated below, to hold. The proofs in [8, 28] depend critically on this property, which follows from [6, Eq. (1.2)] (where it was stated without proof). For completeness we give a proof here, followed by an example that demonstrates the necessity of the full measure assumption.

*Property 6.1* Assume that  $D$  has full measure and let

$$G_\epsilon(\mathbf{x}) := \text{cl conv } \nabla f (\bar{B}(\mathbf{x}; \epsilon) \cap D).$$

For all  $\epsilon > 0$  and all  $\mathbf{x} \in \mathbb{R}^n$ , one has  $\partial f(\mathbf{x}) \subseteq G_\epsilon(\mathbf{x})$ , where  $\partial f$  is the Clarke subdifferential set presented in [Definition 1.8](#).

**Proof of Property 6.1** Let  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{v} \in \partial f(\mathbf{x})$ . We have from [10, Theorem 2.5.1] that [Theorem 1.2](#) can be stated in a more general manner. Indeed, for any set  $S$  with zero measure, and considering  $\Omega_f$  to be the set of points at which  $f$  fails to be differentiable, the following holds:

$$\partial f(\mathbf{x}) = \text{conv} \left\{ \lim_j \nabla f(\mathbf{y}^j) : \mathbf{y}^j \rightarrow \mathbf{x} \text{ where } \mathbf{y}^j \notin S \cup \Omega_f \text{ for all } j \in \mathbb{N} \right\}.$$

In particular, since  $D$  has full measure and  $f$  is differentiable on  $D$ , it follows that

$$\partial f(\mathbf{x}) = \text{conv} \left\{ \lim_j \nabla f(\mathbf{y}^j) : \mathbf{y}^j \rightarrow \mathbf{x} \text{ with } \mathbf{y}^j \in D \text{ for all } j \in \mathbb{N} \right\}.$$

Considering this last relation and Carathéodory's theorem, it follows that  $\mathbf{v} \in \text{conv} \left\{ \boldsymbol{\xi}^1, \dots, \boldsymbol{\xi}^{n+1} \right\}$ , where, for all  $i \in \{1, \dots, n+1\}$ , one has  $\boldsymbol{\xi}^i = \lim_j \nabla f(\mathbf{y}^{j,i})$

for some sequence  $\{y^{j,i}\}_{j \in \mathbb{N}} \subset D$  converging to  $x$ . Hence, there must exist a sufficiently large  $j_i \in \mathbb{N}$  such that, for all  $j \geq j_i$ , one obtains

$$y^{j,i} \in \bar{B}(x; \epsilon) \cap D \implies \nabla f(y^{j,i}) \in \nabla f(\bar{B}(x; \epsilon) \cap D) \subseteq \text{conv } \nabla f(\bar{B}(x; \epsilon) \cap D).$$

Recalling that  $G_\epsilon(x)$  is the closure of  $\text{conv } \nabla f(\bar{B}(x; \epsilon) \cap D)$ , it follows that  $\xi^i \in G_\epsilon(x)$  for all  $i \in \{1, \dots, n+1\}$ . Moreover, since  $G_\epsilon(x)$  is convex, we have  $v \in G_\epsilon(x)$ . The result follows since  $x \in \mathbb{R}^n$  and  $v \in \partial f(x)$  were arbitrarily chosen.  $\square$

With the assumption that  $D$  has full measure, Property 6.1 holds and hence the proofs of the results in [8, 28] are all valid. In particular, the proof of (ii) in [28, Lemma 3.2], which borrows from [8, Lemma 3.2], depends on Property 6.1. See also [8, the top of p. 762].

The following example shows that Property 6.1 might not hold if  $D$  is assumed only to be an open dense set, not necessarily of full measure.

*Example 6.2* Let  $\delta \in (0, 1)$  and  $\{q_k\}_{k \in \mathbb{N}}$  be the enumeration of the rational numbers in  $(0, 1)$ . Define

$$D := \bigcup_{k=1}^{\infty} Q_k, \text{ where } Q_k := \left( q_k - \frac{\delta}{2^{k+1}}, q_k + \frac{\delta}{2^{k+1}} \right).$$

Clearly, its Lebesgue measure satisfies  $0 < \lambda(D) \leq \delta$ . Moreover, the set  $D$  is an open dense subset of  $[0, 1]$ . Now, let  $i_D : [0, 1] \rightarrow \mathbb{R}$  be the indicator function of the set  $D$ ,

$$i_D(x) = \begin{cases} 1, & \text{if } x \in D, \\ 0, & \text{if } x \notin D. \end{cases}$$

Then, considering the Lebesgue integral, we define the function  $f : [0, 1] \rightarrow \mathbb{R}$ ,

$$f(x) = \int_{[0,x]} i_D d\lambda.$$

Let us prove that  $f$  is a Lipschitz continuous function on  $(0, 1)$ . To see this, note that given any  $a, b \in (0, 1)$  with  $b > a$ , it follows that

$$|f(b) - f(a)| = \left| \int_{[0,b]} i_D d\lambda - \int_{[0,a]} i_D d\lambda \right| = \left| \int_{(a,b)} i_D d\lambda \right| \leq \int_{(a,b)} 1 d\lambda = b - a,$$

which ensures that  $f$  is a Lipschitz continuous function on  $(0, 1)$ . Consequently, the Clarke subdifferential set of  $f$  at any point in  $(0, 1)$  is well defined.

Moreover, we claim that, for all  $k \in \mathbb{N}$ ,  $f$  is continuously differentiable at any point  $q \in \mathcal{Q}_k$  and the following holds

$$f'(q) = i_D(q) = 1. \tag{6.7}$$

Indeed, given any  $q \in \mathcal{Q}_k$ , we have

$$f(q + t) - f(q) = \int_{[0, q+t]} i_D d\lambda - \int_{[0, q]} i_D d\lambda = \int_{(q, q+t]} i_D d\lambda, \text{ for } t > 0.$$

Since  $\mathcal{Q}_k$  is an open set, we can find  $\bar{t} > 0$  such that  $[q, q + t] \subset \mathcal{Q}_k \subset D$ , for all  $t \leq \bar{t}$ . Hence, given any  $t \in (0, \bar{t}]$ , it follows that

$$f(q + t) - f(q) = \int_{(q, q+t]} 1 d\lambda = t \implies \lim_{t \downarrow 0} \frac{f(q + t) - f(q)}{t} = 1 = i_D(q).$$

The same reasoning can be used to see that the left derivative of  $f$  at  $q$  exists and it is equal to  $i_D(q)$ . Consequently, we have  $f'(q) = i_D(q) = 1$  for all  $q \in \mathcal{Q}_k$ , which yields that  $f$  is continuously differentiable on  $D$ .

By the Lebesgue differentiation theorem, we know that  $f'(x) = i_D(x)$  almost everywhere. Since the set  $[0, 1] \setminus D$  does not have measure zero, this means that there must exist  $z \in [0, 1] \setminus D$  such that  $f'(z) = i_D(z) = 0$ . Defining  $\epsilon := \min\{z, 1 - z\}/2$ , we see, by (6.7), that the set

$$G_\epsilon(z) := \text{cl conv } \nabla f([z - \epsilon, z + \epsilon] \cap D)$$

is a singleton  $G_\epsilon(z) = \{1\}$ . However, since  $f'(z) = 0$ , it follows that  $0 \in \partial f(z)$ , which implies  $\partial f(z) \not\subset G_\epsilon(z)$ .

Note that it is stated on [8, p. 754] and [28, p. 381] that the following holds: for all  $0 \leq \epsilon_1 < \epsilon_2$  and all  $\mathbf{x} \in \mathbb{R}^n$ , one has  $\hat{\partial}_{\epsilon_1} f(\mathbf{x}) \subseteq G_{\epsilon_2}(\mathbf{x})$ . Property 6.1 is a special case of this statement with  $\epsilon_1 = 0$ , and hence this statement too holds only under the full measure assumption.

Finally, it is worth mentioning that in practice, the full measure assumption on  $D$  usually holds. In particular, whenever a real-valued function is semi-algebraic (or, more generally, “tame”)—in other words, for all practical purposes virtually always—it is continuously differentiable on an open set of full measure. Hence, the original proofs hold in such contexts.

## Appendix 2

In this appendix, we summarize why it is not necessary that the iterates and sampled points of the algorithm lie in the set  $D$  in which  $f$  is continuously differentiable, and that rather it is sufficient to ensure that  $f$  is differentiable at these points, as in Algorithm GS. We do this by outlining how to modify the proofs in [28] to extend to this case.

1. That the gradients at the sampled points  $\{\mathbf{x}^{k,j}\}$  exist follows with probability one from Rademacher’s theorem, while the existence of the gradients at the iterates  $\{\mathbf{x}^k\}$  is ensured by the statement of Algorithm GS. Notice that the proof of part (ii) of [28, Theorem 3.3] still holds in our setting with the statement that the components of the sampled points are “sampled independently and uniformly from  $\bar{B}(\mathbf{x}^k; \epsilon) \cap D$ ” replaced with “sampled independently and uniformly from  $\bar{B}(\mathbf{x}^k; \epsilon)$ ”.
2. One needs to verify that  $f$  being differentiable at  $\mathbf{x}^k$  is enough to ensure that the line search procedure presented in (6.3) terminates finitely. This is straightforward. Since  $\nabla f(\mathbf{x}^k)$  exists, it follows that the directional derivative along any vector  $\mathbf{d} \in \mathbb{R}^n \setminus \{0\}$  is given by  $f'(\mathbf{x}^k; \mathbf{d}) = \nabla f(\mathbf{x}^k)^T \mathbf{d}$ . Furthermore, since  $-\nabla f(\mathbf{x}^k)^T \mathbf{g}^k \leq -\|\mathbf{g}^k\|^2$  (see [8, p. 756]), it follows, for any  $\beta \in (0, 1)$ , that there exists  $\bar{t} > 0$  such that

$$f(\mathbf{x}^k - t\mathbf{g}^k) < f(\mathbf{x}^k) - t\beta\|\mathbf{g}^k\|^2 \text{ for any } t \in (0, \bar{t}).$$

This shows that the line search is well defined.

3. The only place where we actually need to modify the proof in [28] concerns item (ii) in Lemma 3.2, where it is stated that  $\nabla f(\mathbf{x}^k) \in G_\epsilon(\bar{\mathbf{x}})$  (for a particular point  $\bar{\mathbf{x}}$ ) because  $\mathbf{x}^k \in \bar{B}(\bar{\mathbf{x}}; \epsilon/3) \cap D$ ; the latter is not true if  $\mathbf{x}^k \notin D$ . However, using Property 6.1, we have

$$\nabla f(\mathbf{x}^k) \in \partial f(\mathbf{x}^k) \subset G_{\epsilon/3}(\mathbf{x}^k) \subset G_\epsilon(\bar{\mathbf{x}}) \text{ when } \mathbf{x}^k \in \bar{B}(\bar{\mathbf{x}}; \epsilon/3),$$

and therefore,  $\nabla f(\mathbf{x}^k) \in G_\epsilon(\bar{\mathbf{x}})$  even when  $\mathbf{x}^k \notin D$ .

Finally, although it was convenient in Appendix 1 to state Property 1 in terms of  $D$ , it actually holds if  $D$  is replaced by any full measure set on which  $f$  is differentiable. Nonetheless, it is important to note that the proofs of the results in [8, 28] do require that  $f$  be *continuously* differentiable on  $D$ . This assumption is used in the proof of (i) in [28, Lemma 3.2].

## References

1. Asl, A., Overton, M.L.: Analysis of the gradient method with an Armijo–Wolfe line search on a class of nonsmooth convex functions. Optim. Method Softw. (2017). <https://doi.org/10.1080/10556788.2019.1673388>

2. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. *IMA J. Numer. Anal.* **8**(1), 141–148 (1988)
3. Birgin, E., Martinez, J., Raydan, M.: Spectral projected gradient methods: review and perspectives. *J. Stat. Softw.* **60**(3), 1–21 (2014)
4. Burke, J.V., Lin, Q.: The gradient sampling algorithm for directionally Lipschitzian functions (in preparation)
5. Burke, J.V., Overton, M.L.: Variational analysis of non-Lipschitz spectral functions. *Math. Program.* **90**(2, Ser. A), 317–351 (2001)
6. Burke, J.V., Lewis, A.S., Overton, M.L.: Approximating subdifferentials by random sampling of gradients. *Math. Oper. Res.* **27**(3), 567–584 (2002)
7. Burke, J.V., Lewis, A.S., Overton, M.L.: Two numerical methods for optimizing matrix stability. *Linear Algebra Appl.* **351/352**, 117–145 (2002)
8. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM J. Optim.* **15**(3), 751–779 (2005)
9. Burke, J.V., Henrion, D., Lewis, A.S., Overton, M.L.: HIFOO—a MATLAB package for fixed-order controller design and  $H_\infty$  optimization. In: *Fifth IFAC Symposium on Robust Control Design*, Toulouse (2006)
10. Clarke, F.H.: *Optimization and Nonsmooth Analysis*. Wiley, New York (1983). Reprinted by SIAM, Philadelphia, 1990
11. Crema, A., Loreto, M., Raydan, M.: Spectral projected subgradient with a momentum term for the Lagrangean dual approach. *Comput. Oper. Res.* **34**(10), 3174–3186 (2007)
12. Curtis, F.E., Overton, M.L.: A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. *SIAM J. Optim.* **22**(2), 474–500 (2012)
13. Curtis, F.E., Que, X.: An adaptive gradient sampling algorithm for nonsmooth optimization. *Optim. Methods Softw.* **28**(6), 1302–1324 (2013)
14. Curtis, F.E., Que, X.: A quasi-Newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees. *Math. Program. Comput.* **7**(4), 399–428 (2015)
15. Curtis, F.E., Mitchell, T., Overton, M.L.: A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optim. Methods Softw.* **32**(1), 148–181 (2017)
16. Curtis, F.E., Robinson, D.P., Zhou, B.: A self-correcting variable-metric algorithm framework for nonsmooth optimization. *IMA J. Numer. Anal.* (2019). <https://doi.org/10.1093/imanum/drz008>; <https://academic.oup.com/imajna/advance-article/doi/10.1093/imanum/drz008/5369122?guestAccessKey=a7e5eee5-9ed6-4a95-9f6c-f305237d0849>
17. Davis, D., Drusvyatskiy, D.: Stochastic model-based minimization of weakly convex functions. *SIAM J. Optim.* **29**(1), 207–239 (2019). <https://doi.org/10.1137/18M1178244>
18. Davis, D., Drusvyatskiy, D., Kakade, S., Lee, J.D.: Stochastic subgradient method converges on tame functions. *Found. Comput. Math.* (2019). <https://doi.org/10.1007/s10208-018-09409-5>
19. Estrada, A., Mitchell, I.M.: Control synthesis and classification for unicycle dynamics using the gradient and value sampling particle filters. In: *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, pp. 108–114 (2018).
20. Fletcher, R.: *Practical Methods of Optimization*, 2nd edn. Wiley, New York (1987)
21. Fletcher, R.: On the Barzilai-Borwein method. In: Qi, L., Teo, K., Yang, X. (eds.) *Optimization and Control with Applications*, pp. 235–256. Springer, Boston (2005)
22. Guo, J., Lewis, A.S.: Nonsmooth variants of Powell’s BFGS convergence theorem. *SIAM J. Optim.* **28**(2), 1301–1311 (2018). <https://doi.org/10.1137/17M1121883>
23. Hare, W., Nutini, J.: A derivative-free approximate gradient sampling algorithm for finite minimax problems. *Comput. Optim. Appl.* **56**(1), 1–38 (2013). <https://doi.org/10.1007/s10589-013-9547-6>
24. Helou, E.S., Santos, S.A., Simões, L.E.A.: On the differentiability check in gradient sampling methods. *Optim. Methods Softw.* **31**(5), 983–1007 (2016)
25. Helou, E.S., Santos, S.A., Simões, L.E.A.: On the local convergence analysis of the gradient sampling method for finite max-functions. *J. Optim. Theory Appl.* **175**(1), 137–157 (2017)



26. Hosseini, S., Uschmajew, A.: A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM J. Optim.* **27**(1), 173–189 (2017). <https://doi.org/10.1137/16M1069298>
27. Kiwiel, K.C.: A method for solving certain quadratic programming problems arising in nonsmooth optimization. *IMA J. Numer. Anal.* **6**(2), 137–152 (1986)
28. Kiwiel, K.C.: Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM J. Optim.* **18**(2), 379–388 (2007)
29. Kiwiel, K.C.: A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM J. Optim.* **20**(4), 1983–1994 (2010). <https://doi.org/10.1137/090748408>
30. Larson, J., Menickelly, M., Wild, S.M.: Manifold sampling for  $\ell_1$  nonconvex optimization. *SIAM J. Optim.* **26**(4), 2540–2563 (2016). <https://doi.org/10.1137/15M1042097>
31. Lemaréchal, C., Oustry, F., Sagastizábal, C.: The U-Lagrangian of a convex function. *Trans. Am. Math. Soc.* **352**(2), 711–729 (2000)
32. Lewis, A.S.: Active sets, nonsmoothness, and sensitivity. *SIAM J. Optim.* **13**(3), 702–725 (2002)
33. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via quasi-Newton methods. *Math. Program.* **141**(1–2, Ser. A), 135–163 (2013). <https://doi.org/10.1007/s10107-012-0514-2>
34. Lin, Q.: Sparsity and nonconvex nonsmooth optimization. Ph.D. thesis, Department of Mathematics, University of Washington (2009)
35. Loreto, M., Aponte, H., Cores, D., Raydan, M.: Nonsmooth spectral gradient methods for unconstrained optimization. *EURO J. Comput. Optim.* **5**(4), 529–553 (2017)
36. Mifflin, R., Sagastizábal, C.: A VU-algorithm for convex minimization. *Math. Program.* **104**(2–3), 583–608 (2005)
37. Nesterov, Y., Spokoiny, V.: Random gradient-free minimization of convex functions. *Found. Comput. Math.* **17**(2), 527–566 (2017). <https://doi.org/10.1007/s10208-015-9296-2>
38. Raydan, M.: On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.* **13**(3), 321–326 (1993)
39. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **7**(1), 26–33 (1997)
40. Rockafellar, R.T.: Lagrange multipliers and subderivatives of optimal value functions in nonlinear programming. In: Sorensen, D.C., Wets, R.J.B. (eds.) *Mathematical Programming Study*, Mathematical Programming Studies, Chap. 3, pp. 28–66. North-Holland, Amsterdam (1982). <http://www.springerlink.com/index/g03582565267714p.pdf>
41. Rockafellar R.T., Wets, R.J.B.: *Variational Analysis*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 317. Springer, Berlin (1998). <https://doi.org/10.1007/978-3-642-02431-3>
42. Tang, C.M., Liu, S., Jian, J.B., Li, J.L.: A feasible SQP-GS algorithm for nonconvex, nonsmooth constrained optimization. *Numer. Algorithms* **65**(1), 1–22 (2014). <https://doi.org/10.1007/s11075-012-9692-5>
43. Traft, N., Mitchell, I.M.: Improved action and path synthesis using gradient sampling. In: *Proceedings of the IEEE Conference on Decision and Control*, pp. 6016–6023 (2016)