The Gradient Sampling Methodology

James V. Burke* Frank E. Curtis† Adrian S. Lewis‡ Michael L. Overton§

January 10, 2019

1 Introduction

The principal methodology for minimizing a smooth function is the steepest descent (gradient) method. One way to extend this methodology to the minimization of a nonsmooth function involves approximating subdifferentials through the random sampling of gradients. This approach, known as *gradient sampling* (GS), gained a solid theoretical foundation about a decade ago [BLO05, Kiw07], and has developed into a comprehensive methodology for handling nonsmooth, potentially nonconvex functions in the context of optimization algorithms. In this article, we summarize the foundations of the GS methodology, provide an overview of the enhancements and extensions to it that have developed over the past decade, and highlight some interesting open questions related to GS.

2 Fundamental Idea

The central idea of GS can be explained as follows. When a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at a point x, the traditional steepest descent direction for f at x in the 2-norm is found by observing that

$$\arg \min_{\|d\|_2 \leq 1} \nabla f(x)^T d = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2}; \tag{1}$$

in particular, this leads to calling the negative gradient, namely, $-\nabla f(x)$, the direction of steepest descent for f at x. However, when f is not differentiable near x, one finds that following the negative gradient direction might offer only a small amount of decrease in f; indeed, obtaining decrease from x along $-\nabla f(x)$ may be possible only with a very small stepsize. The GS methodology is based on the idea of stabilizing this definition of steepest descent by instead finding a direction to *approximately* solve

$$\min_{\|d\|_2 \leq 1} \max_{g \in \bar{\partial}_\epsilon f(x)} g^T d, \tag{2}$$

where $\bar{\partial}_\epsilon f(x)$ is the ϵ-subdifferential of f at x [Gol77]. To understand the context of this idea, recall that the (Clarke) subdifferential of a locally Lipschitz f at x, denoted $\bar{\partial} f(x)$, is the convex hull of the limits of all sequences of gradients evaluated at sequences of points, at which f is differentiable, that converge to x [Cla75]. The ϵ-subdifferential, in turn, is the convex hull of all subdifferentials at points within an ϵ-neighborhood of x. Although the ϵ-subdifferential of f at x is not readily computed, the central idea of gradient sampling is to approximate the solution of (2) by finding the smallest norm vector in the convex hull of gradients computed at randomly generated points in an ϵ-neighborhood of x, then normalizing the result to have unit norm. See [BLO02] for analysis on approximating an ϵ-subdifferential by sampling gradients at randomly generated points.

A complete algorithm based on the GS methodology is stated as Algorithm 1, taken from the recent survey paper [BCL+19]. To illustrate the efficacy of this algorithm compared to more standard gradient and subgradient methodologies, let us show its performance on a nonsmooth variant of the nonconvex Rosenbrock function [Ros60], namely,

$$f(x) = 8|x_1^2 - x_2| + (1 - x_1)^2. \tag{3}$$

The contours of this function are shown in Figure 1; the black asterisk indicates the initial iterate $x^0 = (0.1, 0.1)$ and the red asterisk indicates the unique minimizer $x^* = (1, 1)$. The blue dots show the iter-

*Department of Mathematics, University of Washington, Seattle, WA. jvburke01@gmail.com. Supported in part by the U.S. National Science Foundation grant DMS-1514559.

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA. frank.e.curtis@gmail.com. Supported in part by the U.S. Department of Energy grant DE-SC0010615 and National Science Foundation grant CCF-1740796.

‡School of Operations Research and Information Engineering, Cornell University, Ithaca, NY. adrian.lewis@cornell.edu. Supported in part by the U.S. National Science Foundation grant DMS-1613996.

§Courant Institute of Mathematical Sciences, New York University. mo1@nyu.edu. Supported in part by the U.S. National Science Foundation grant DMS-1620083.

ates generated by the gradient sampling method (Algorithm 1) converging to x^*, roughly tracing out the parabola on which f is nonsmooth, but never actually landing on it, even to finite precision. In contrast, the magenta dots show the iterates of the gradient method with the same line search enforcing (4) from Algorithm 1, indicating that these iterates move directly toward the parabola on which f is nonsmooth and stall without moving along it toward the minimizer x^*. The essential difficulty is that the direction of descent tangential to the parabola is overwhelmed by the steepness of the graph of the function near the parabola. The gradient sampling method, on the other hand, by choosing the direction of least norm in the convex hull of sampled gradients, is able to approximate this tangential direction of descent.

The poor behavior of the gradient method in this context is well known, even in the convex case [HUL93]; see [AO18] for a discussion of the behavior of the gradient method on a simple nonsmooth convex function. In these experiments, for both algorithms, $x_1^2 - x_2$ was nonzero at all iterates, even in finite precision, so gradients were always defined. Figure 2 shows the function values $\{f(x^k)\}$ generated by the two methods. Both algorithms were terminated as soon as the objective and/or gradient was evaluated at 2000 points—including iterates, trial points in the line searches, and randomly generated points at which the gradient is evaluated for Algorithm 1. Algorithm 1 is able to reach iterates with much better objective values within the same budget.[1]

It is also instructive to consider a subgradient method [Sho85, Rus06] that sets iterates by

$$x^{k+1} \leftarrow x^k - t_k d^k,$$

where d^k is any subgradient of f at x^k (i.e., any element of $\bar{\partial} f(x^k)$) and $\{t_k\}$ is set as a fixed stepsize or according to a diminishing stepsize schedule. This is a popular approach in the optimization literature, which has convergence guarantees in various contexts without requiring that the value of f decreases at each iteration. By not requiring monotonic decrease, the method does not get stuck near the parabola on which f is nonsmooth. However, progress is slow since the method has no mechanism for identifying the tangential direction of descent along the parabola. Instead, it is destined to oscillate back-and-forth across the parabola as it creeps tangentially toward the minimizer x^*. In this experiment, $x_1^2 - x_2$ was nonzero (even in finite precision) at all but a handful of the iterates, and since the only subgradient of f at such a point is the gradient, the method is, for all practical purposes, identical to a gradient method with the same stepsizes. The iterates of this method with $\{t_k\} = \{0.1/k\}$ are shown in Figure 3, and the performance with different choices for $\{t_k\}$ is shown in Figure 4. With the same function and gradient evaluation budget as the methods above, this approach—for all stepsize choices—is slow. One might be able to obtain better results by tuning the stepsize choice further. Note, however, that Algorithm 1 does not require such parameter tuning.

Of course, there are other effective methods for nonsmooth optimization that we do not consider here, in part because they are significantly more complicated to describe; these include bundle methods [Kiw85, SZ92], which have been used extensively for decades, and quasi-Newton methods [LO13].

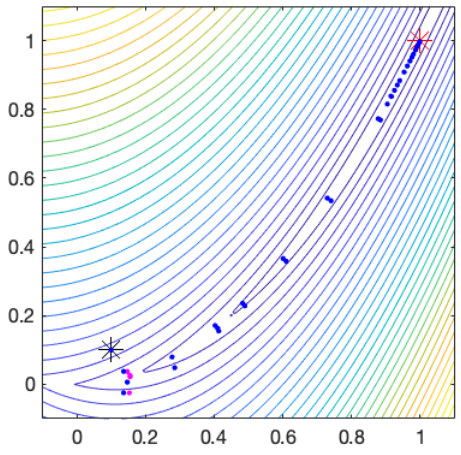

Figure 1: Contours of the nonsmooth Rosenbrock function (3) showing iterates generated by the gradient sampling method (blue dots) and the ordinary gradient method with the same line search (magenta dots). The black asterisk is the initial point and the red asterisk shows the unique minimizer.

3 Convergence Theory

Algorithm 1 is conceptually straightforward. At each iterate, one need only compute gradients at randomly sampled points, project the origin onto the convex

[1] We used the following parameters for Algorithm 1: $\epsilon_0 = \nu_0 = 0.1$, $m = 3$, $\beta = 10^{-8}$, $\gamma = 0.5$, $\epsilon_{\text{opt}} = \nu_{\text{opt}} = 0$, and $\theta_\epsilon = \theta_\nu = 0.1$. The gradient method used the same line search with $\beta = 10^{-8}$ and $\gamma = 0.5$. Both algorithms used most of their function and gradient evaluations in later iterations. The final sampling radius for Algorithm 1 was 10^{-5}.

Algorithm 1 : Gradient Sampling with a Line Search

Require: initial point 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 $\{x^{k,1}, \ldots, x^{k,m}\}$ uniformly from $\mathbb{B}(x^k, \epsilon_k)$

3: compute g^k as the solution of $\min_{g \in \mathcal{G}^k} \frac{1}{2}\|g\|_2^2$, where $\mathcal{G}^k := \text{conv}\{\nabla f(x^k), \nabla f(x^{k,1}), \ldots, \nabla f(x^{k,m})\}$

4: **if** $\|g^k\|_2 \leq \nu_{\text{opt}}$ and $\epsilon_k \leq \epsilon_{\text{opt}}$ **then** terminate

5: **if** $\|g^k\|_2 \leq \nu_k$

6: **then** set $\nu_{k+1} \leftarrow \theta_\nu \nu_k$, $\epsilon_{k+1} \leftarrow \theta_\epsilon \epsilon_k$, and $t_k \leftarrow 0$

7: **else** set $\nu_{k+1} \leftarrow \nu_k$, $\epsilon_{k+1} \leftarrow \epsilon_k$, and

$$t_k \leftarrow \max\left\{t \in \{1, \gamma, \gamma^2, \ldots\} : f(x^k - t g^k) < f(x^k) - \beta t \|g^k\|_2^2\right\} \tag{4}$$

8: **if** f is differentiable at $x^k - t_k g^k$

9: **then** set $x^{k+1} \leftarrow x^k - t_k g^k$

10: **else** set x^{k+1} randomly as any point where f is differentiable such that

$$f(x^{k+1}) < f(x^k) - \beta t_k \|g^k\|_2^2 \quad \text{and} \quad \|x^k - t_k g^k - x^{k+1}\|_2 \leq \min\{t_k, \epsilon_k\}\|g^k\|_2 \tag{5}$$

11: **end for**

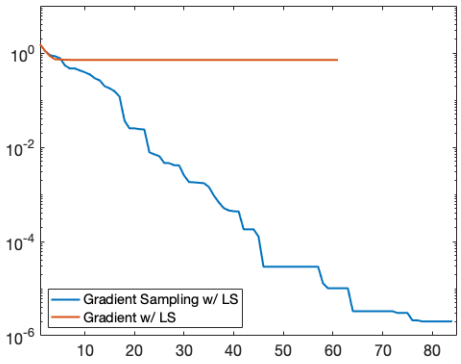

Figure 2: Function values by iteration number for the gradient sampling method and the gradient method, both run with a backtracking line search (LS).

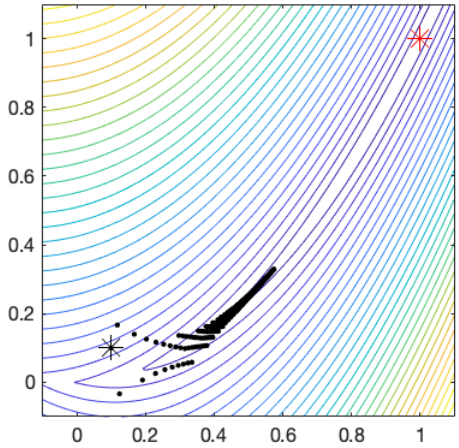

Figure 3: Contours of the nonsmooth Rosenbrock function (3) showing iterates generated by the subgradient method with $\{t_k\} = \{0.1/k\}$. The black asterisk is the initial point and the red asterisk shows the unique minimizer.

hull of these gradients (by solving a strongly convex quadratic program (QP) for which specialized algorithms have been designed [Kiw86]), and perform a line search. The other details relate to dynamically setting the sampling radii $\{\epsilon_k\}$ and ensuring that the objective f is differentiable at each iterate.

On the other hand, the convergence theory for the algorithm when minimizing a locally Lipschitz function involves important, subtle details. Rademacher's theorem states that locally Lipschitz functions are differentiable almost everywhere [Cla83], ensuring that the gradients sampled at the randomly generated points are well defined with probability one. However, this is not sufficient to ensure convergence. To obtain a satisfactory convergence result it is required that the set of points at which f is *continuously differentiable* has full measure in $\mathbb{R}^n$. For further discussion of this issue, see [BCL+19].

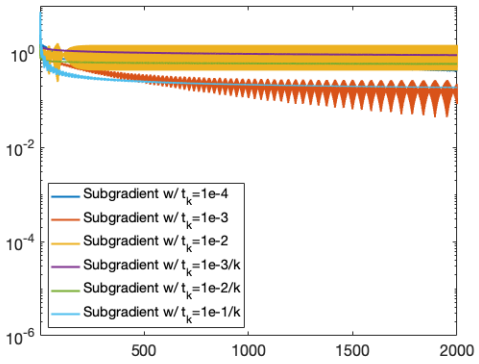

Figure 4: Function values by iteration number for the subgradient method with different stepsize sequences, indicated by the formula for t_k in the legend.

The following theorem, whose precise statement is taken from [BCL+19], but whose proof depends on the convergence theorems in [BLO05, Kiw07], is the main convergence result for Algorithm 1. Other results of interest that can be proved relate to the behavior of the algorithm when the tolerances ν_{opt} and ϵ_{opt} are positive, so the algorithm terminates finitely, or when one sets $\theta_\epsilon = 1$, so that the sampling radius is fixed, in which case one can prove convergence to ϵ-stationarity; see [BCL+19] for more details.

Theorem 1. *Suppose that f is locally Lipschitz on $\mathbb{R}^n$ and continuously differentiable on an open set with full measure in $\mathbb{R}^n$. Suppose also that Algorithm 1 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 1 is well defined in the sense that the sampled gradients exist in every iteration, the algorithm does not terminate, and either*

- $\{f(x^k)\} \searrow -\infty$ *or*
- $\{\nu_k\} \searrow 0$, $\{\epsilon_k\} \searrow 0$, *and each limit point $\bar{x}$ of the sequence $\{x^k\}$ is Clarke stationary for f, that is, $0 \in \bar{\partial} f(\bar{x})$.*

It has been shown that the result of Theorem 1 can be extended for some cases of non-locally Lipschitz f, in particular, when it is *directionally Lipschitz* [Lin09]. Extending it to the general non-locally Lipschitz setting, on the other hand, seems quite difficult. One can also prove that, in the case of minimizing finite-max functions, Algorithm 1 can achieve a linear rate of local convergence, at least in a certain probabilistic sense [HSS17]. This should not be too surprising given the connection between the GS methodology and standard steepest descent.

4 Enhancements

Since the inception and analysis of the initial GS algorithm in [BLO05], various enhancements and extensions have appeared in the literature. First, a few fundamental advances were published in [Kiw07]; in particular, in this work, Kiwiel showed how to simplify the analysis of a basic GS algorithm and extend it for some interesting algorithm variants, such as when invoking a trust region methodology. Other proposed enhancements include techniques for avoiding the differentiability check in Steps 8–10 of Algorithm 1) [Kiw07, HSS16], performing the gradient sampling adaptively so that only $\mathcal{O}(1)$ gradients need to be sampled in each iteration [CQ13, CQ15], and for incorporating second-order derivatives or approximations, say by borrowing quasi-Newton ideas from the smooth optimization literature [CQ13, CQ15]. Added benefits of adaptive sampling are that one can re-use gradients computed in previous iterations and *warm-start* the solve of each QP so that the computation of each search direction becomes relatively inexpensive.

The GS methodology has also been extended for solving constrained optimization problems. Specifically, a Riemannian GS method has been proposed for optimization on manifolds [HU17], and a so-called SQP-GS method, which merges the GS methodology with that of a penalty sequential quadratic programming (SQP) technique from the smooth optimization literature, has been proposed for solving inequality constrained optimization problems in which the objective and constraint functions may be nonsmooth and/or nonconvex [CO12]. A feasible variant of the SQP-GS method has also been proposed, which establishes a path for the design of two-phase approaches: a first phase seeking feasibility and a second phase seeking optimality [TLJL14].

Another interesting line of work has been on adaptions of the GS methodology for designing derivative-free algorithms for minimizing nonsmooth functions. In a couple of these cases, authors have proposed to use the GS methodology in a relatively straightforward manner with gradients replaced by gradient approximations constructed using function evaluations [Kiw10, HN13]. There has also been work on methods that do not borrow the gradient sampling strategy *per se*, but are still motivated by the GS methodology in

terms of the types of subproblems that are employed to compute search directions [LMW16].

For more information on the enhancements and extensions that have been made to the GS methodology over the past decade, as well as information about available software and success stories in practice, see [BCL+19].

5 Closing Remarks

Gradient sampling is a conceptually straightforward, yet powerful approach for extending the steepest descent methodology to the minimization of nonsmooth, nonconvex functions. The fundamental idea of GS is to stabilize the notion of a steepest descent direction by finding the minimum norm element of the convex hull of gradients evaluated at points randomly sampled near the current iterate. The methodology enjoys a solid theoretical foundation and has been enhanced and extended in various ways, such as for solving constrained optimization problems.

There remain various interesting avenues of research related to the GS methodology. For example, it remains an open question how far one may be able to extend the convergence theory for a GS method in terms of minimizing non-locally Lipschitz functions; e.g., can one extend the GS theory for the class of semi-algebraic, but not locally Lipschitz or directionally Lipschitz functions? On the other hand, one can imagine various opportunities for exploring tailored GS approaches when one aims to minimize a function for which one has knowledge about the structure of the nonsmoothness of a function f. How should sampling be performed when, at any given iterate, one has knowledge about the directions in which f is smooth and directions in which it is nonsmooth (at least in a neighborhood of the current iterate)?

One also has the sense that there remain numerous avenues to pursue in the context of constrained optimization. Given the range of methodologies for solving smooth constrained optimization problems, one could explore techniques that combine these approaches with gradient sampling so that convergence guarantees could potentially be obtained when handling nonsmooth functions as well. One might also re-evaluate the use of certain methods, such as some exact penalty methods, which have previously fallen out of favor due to the presence of nonsmoothness. After all, the issues that inhibited the effectiveness of such approaches might no longer be of concern since GS might naturally overcome them.

Finally, there remain various open questions about the possible connections between the GS methodology and other randomized and/or stochastic optimization methods. The basic GS method involves computing the minimum norm element in the convex hull of gradients evaluated at randomly generated points. Can the GS theory be extended when the subproblems for computing the search directions are only solved approximately? If so, this might represent a step toward tying the convergence theory of GS with those of other randomized/stochastic gradient/subgradient approaches, which have attracted a lot of recent attention; see, e.g., [DD18, DDKL19].

Acknowledgment

The authors would like to sincerely thank the ICS Prize Committee, which was chaired by Andreas Wächter and included Fatma Kilinc-Karzan, Douglas Shier, and Cole Smith. The prize was awarded for the articles [BLO05, CMO17, CO12, CQ13, CQ15, BCL+19], which involved co-authorship by Tim Mitchell, Xiaocun Que and Lucas E. A. Simões. We are honored to receive this prize and are grateful for this opportunity to promote work related to the GS methodology.

References

[AO18] A. Asl and M. L. Overton. Analysis of the Gradient Method with an Armijo-Wolfe Line Search on a Class of Nonsmooth Convex Functions, 2018. arXiv:1711.08517v2.

[BCL+19] J. V. Burke, F. E. Curtis, A. S. Lewis, M. L. Overton, and L. E. A. Simões. Gradient Sampling Methods for Nonsmooth Optimization. In A. Bagirov, M. Gaudioso, N. Karmitsa, and M. Mäkelä, editors, *Special Methods for Nonsmooth Optimization*. Springer, 2019. https://arxiv.org/abs/1804.11003.

[BLO02] J. V. Burke, A. S. Lewis, and M. L. Overton. Approximating Subdifferentials by Random Sampling of Gradients. *Math. Oper. Res.*, 27(3):567–584, 2002.

[BLO05] J. V. Burke, A. S. Lewis, and M. L. Overton. A Robust Gradient Sampling Al-

gorithm for Nonsmooth, Nonconvex Optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.

[Cla75] F. H. Clarke. Generalized Gradients and Applications. *Trans. Amer. Math. Soc.*, 205:247–262, 1975.

[Cla83] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley, New York, 1983. Reprinted by SIAM, Philadelphia, 1990.

[CMO17] F. E. Curtis, T. Mitchell, and M. L. Overton. A BFGS-SQP method for Nonsmooth, Nonconvex, Constrained Optimization and its Evaluation using Relative Minimization Profiles. *Optimization Methods and Software*, 32(1):148–181, 2017.

[CO12] F. E. Curtis and M. L. Overton. A Sequential Quadratic Programming Algorithm for Nonconvex, Nonsmooth Constrained Optimization. *SIAM Journal on Optimization*, 22(2):474–500, 2012.

[CQ13] F. E. Curtis and X. Que. An Adaptive Gradient Sampling Algorithm for Nonsmooth Optimization. *Optimization Methods and Software*, 28(6):1302–1324, 2013.

[CQ15] F. E. Curtis and X. Que. A Quasi-Newton Algorithm for Nonconvex, Nonsmooth Optimization with Global Convergence Guarantees. *Mathematical Programming Computation*, 7(4):399–428, 2015.

[DD18] D. Davis and D. Drusvyatskiy. Stochastic Model-Based Minimization of Weakly Convex Functions, March 2018. arXiv:1803.06523.

[DDKL19] D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee. Stochastic Subgradient Method Converges on Tame Functions. *Found. Comput. Math.*, 2019. To appear.

[Gol77] A. A. Goldstein. Optimization of Lipschitz Continuous Functions. *Math. Programming*, 13(1):14–22, 1977.

[HN13] W. Hare and J. Nutini. A Derivative-Free Approximate Gradient Sampling Algorithm for Finite Minimax Problems. *Computational Optimization and Applications*, 56(1):1–38, Sep 2013.

[HSS16] E. S. Helou, S. A. Santos, and L. E. A. Simões. On the Differentiability Check in Gradient Sampling Methods. *Optimization Methods and Software*, 31(5):983–1007, 2016.

[HSS17] E. S. Helou, S. A. Santos, and L. E. A. Simões. On the Local Convergence Analysis of the Gradient Sampling Method for Finite Max-Functions. *Journal of Optimization Theory and Applications*, 175(1):137–157, 2017.

[HU17] S. Hosseini and A. Uschmajew. A Riemannian Gradient Sampling Algorithm for Nonsmooth Optimization on Manifolds. *SIAM Journal on Optimization*, 27(1):173–189, 2017.

[HUL93] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms. I*, volume 305 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1993.

[Kiw85] K. C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics. Springer-Verlag, New York, NY, USA, 1985.

[Kiw86] K. C. Kiwiel. A Method for Solving Certain Quadratic Programming Problems Arising in Nonsmooth Optimization. *IMA Journal of Numerical Analysis*, 6(2):137–152, 1986.

[Kiw07] K. C. Kiwiel. Convergence of the Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization. *SIAM Journal on Optimization*, 18(2):379–388, 2007.

[Kiw10] K. C. Kiwiel. A Nonderivative Version of the Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization. *SIAM Journal on Optimization*, 20(4):1983–1994, 2010.

[Lin09] Q. Lin. *Sparsity and Nonconvex Nonsmooth Optimization*. PhD thesis, Department of Mathematics, University of Washington, 2009.

[LMW16] J. Larson, M. Menickelly, and S. M. Wild. Manifold Sampling for ℓ_1 Nonconvex Optimization. *SIAM Journal on Optimization*, 26(4):2540–2563, 2016.

[LO13] A. S. Lewis and M. L. Overton. Nonsmooth Optimization via Quasi-Newton Methods. *Math. Program.*, 141(1-2, Ser. A):135–163, 2013.

[Ros60] H. H. Rosenbrock. An Automatic Method for Finding the Greatest or Least Value of a Function. *The Computer Journal*, 3(3):175–184, 1960.

[Rus06] A. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, Princeton, NJ, 2006.

[Sho85] N.Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, Heidelberg, Germany, 1985.

[SZ92] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.

[TLJL14] C.-M. Tang, S. Liu, J.-B. Jian, and J.-L. Li. A Feasible SQP-GS Algorithm for Nonconvex, Nonsmooth Constrained Optimization. *Numerical Algorithms*, 65(1):1–22, Jan 2014.