ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Manifold Sampling for L1 Nonconvex Optimization[1]

Jeffrey Larson, Matt Menickelly, and Stefan M. Wild

Mathematics and Computer Science Division

Preprint ANL/MCS-P5392-0915

September 2015 (*Revised September 2016*)

[1]This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under contract number DE-AC02-06CH11357.

MANIFOLD SAMPLING FOR L1 NONCONVEX OPTIMIZATION

JEFFREY LARSON*, MATT MENICKELLY†*, AND STEFAN M. WILD*

Abstract. We present a new algorithm, called manifold sampling, for the unconstrained minimization of a nonsmooth composite function $h \circ F$ when h has known structure. In particular, by classifying points in the domain of the nonsmooth function h into manifolds, we adapt search directions within a trust-region framework based on knowledge of manifolds intersecting the current trust region. We motivate this idea through a study of ℓ_1 functions, where it is trivial to classify objective function manifolds using zeroth-order information from the constituent functions F_i, and give an explicit statement of a manifold sampling algorithm in this case. We prove that all cluster points of iterates generated by this algorithm are stationary in the Clarke sense. We prove a similar result for a stochastic variant of the algorithm. Additionally, our algorithm can accept iterates that are points where h is nondifferentiable and requires only an approximation of gradients of F at the trust-region center. Numerical results for several variants of the algorithm show that using manifold information from additional points near the current iterate can improve practical performance. The best variants are also shown to be competitive, particularly in terms of robustness, with other nonsmooth, derivative-free solvers.

Key words. Composite Nonsmooth Optimization, Gradient Sampling, Derivative-Free Optimization

AMS subject classifications. 90C56, 49J52

1. Introduction.

This paper addresses the unconstrained optimization problem $\min\{f(x) : x \in \mathbb{R}^n\}$ when f is of the form

$$f(x) = \sum_{i=1}^{r} |F_i(x)| = \|F(x)\|_1 \tag{1.1}$$

and the function $F : \mathbb{R}^n \to \mathbb{R}^r$ is sufficiently smooth, as formalized in the following assumption.

ASSUMPTION 1. *The function f is of the form (1.1), each F_i is continuously differentiable, and each ∇F_i is Lipschitz continuous with Lipschitz constant L_i; define $L = \sum_{i=1}^{r} L_i$.*

Minimizing the function (1.1) is a special case of more general composite nonsmooth optimization

$$\text{minimize}\,\{f(x) = f_s(x) + h(F(x)) : x \in \mathbb{R}^n\}, \tag{1.2}$$

where f_s and F are smooth but h is nonsmooth with known structure. We find that including f_s and ∇f_s obscures the development of the framework so we do not include them; straightforward modifications can accommodate minimizing $f_s(x) + \|F(x)\|_1$. We focus on the objective function (1.1) in order to succinctly introduce, analyze, and empirically study a general algorithmic framework. Although the form of h studied here is convex, our framework does not require this.

Furthermore, this framework—which we refer to as *manifold sampling*—does not require the availability of the Jacobian ∇F. As a result, manifold sampling is applicable both when inexact values for $\nabla F(x)$ are available and in the derivative-free case,

†Lehigh University, Department of Industrial and Systems Engineering. H.S. Mohler Laboratory, 200 West Packer Avenue, Bethlehem, PA 18015. mjm412@lehigh.edu

*Argonne National Laboratory, Mathematics and Computer Science Division. 9700 South Cass Avenue, Lemont, IL 60439. {jmlarson,wild}@anl.gov

when only the values $F(x)$ are available. In Section 2 we motivate the use of the term manifold in the context of functions of the form (1.1) and show that these manifolds can be determined without using Jacobian information. We also review the literature in composite nonsmooth optimization.

Section 3 introduces our manifold sampling algorithm. The algorithm uses a smooth model M of the mapping F and proceeds like a traditional trust-region algorithm until it encounters an area of possible nondifferentiability. In the case of the function (1.1), a signal for potential nondifferentiability is directly obtained from the signs of the r component functions at the current iterate. We also propose different mechanisms for incorporating such sign information from the current iterate and nearby points.

Let $M : \mathbb{R}^n \to \mathbb{R}^r$ denote an approximation to F and let $\nabla M : \mathbb{R}^n \to \mathbb{R}^{n\times r}$ denote the Jacobian of M. Under minimal assumptions, in Section 4 we show that our algorithm's smooth, local model of the function f,

$$m^f(x^k + s) \approx f(x^k) + \left\langle s, \mathbf{proj}\Big(0, \nabla M\left(x^k\right) \partial h\left(F\left(x^k\right)\right)\Big)\right\rangle, \tag{1.3}$$

generates descent directions at the current point x^k. Furthermore, we prove that all cluster points of the algorithm are Clarke stationary points. We show in Section 5 that convergence again holds when using sign information from stochastically generated points.

This stochastic sampling proves to be beneficial in practice, as our numerical tests in Section 6 demonstrate. Our experiments also underscore the relative robustness of four variants of the proposed manifold sampling algorithm. The tested deterministic variants include one that performs efficiently when function evaluations occur sequentially as well as variants that can exploit concurrent function evaluations.

2. Background. We now introduce notation and provide context for our manifold sampling algorithm. We follow the convention throughout this paper that finite sets are denoted by capital Roman letters while (possibly) infinite sets are denoted by capital calligraphic letters.

2.1. Nonsmooth Optimization Preliminaries. We define the set of points at which a function f is differentiable by $\mathcal{D} \subseteq \mathbb{R}^n$ and its complement by $\mathcal{D}^c$. In smooth optimization, a first-order necessary condition for x to be a local minimum of f is that $\nabla f(x) = 0$. In nonsmooth optimization, if $x \in \mathcal{D}^c$, then one needs a more generalized first-order necessary condition; we achieve this with the generalized gradient ∂f, which is a set-valued function referred to as the *Clarke subdifferential*.

The *Clarke directional derivative at x in the direction d* is given by

$$f^\circ(x; d) = \limsup_{y\to x, t\downarrow 0} \frac{f(y + td) - f(y)}{t}. \tag{2.1}$$

The Clarke subdifferential at x is the set of linear support functions of $f^\circ(x; d)$ when viewed as a function of $d \in \mathbb{R}^n$:

$$\partial f(x) = \{v \in \mathbb{R}^n : f^\circ(x; d) \geq \langle v, d\rangle \text{ for all } d \in \mathbb{R}^n\}. \tag{2.2}$$

By using the definitions (2.1) and (2.2), one can show (see, e.g., [4, Proposition 2.3.2]) that if f is locally Lipschitz near x and attains a local minimum or maximum at x, then $0 \in \partial f(x)$. Thus, $0 \in \partial f(x)$ can be seen as a nonsmooth analogue of

the first-order necessary condition $\nabla f(x) = 0$. This condition, $0 \in \partial f(x)$, is referred to as *Clarke stationarity*.

Furthermore, as a consequence of Rademacher's theorem, if f is locally Lipschitz, then $\mathcal{D}^c$ is a set of Lebesgue measure zero in $\mathbb{R}^n$. In such cases, an equivalent definition (see, e.g., [4, Theorem 2.5.1]) of the Clarke subdifferential is

$$\partial f(x) = \mathbf{co}\left\{ \lim_{y^j \to x} \nabla f(y^j) : \ \{y^j : j \geq 1\} \subset \mathcal{D} \right\}, \tag{2.3}$$

where **co** denotes the convex hull of a set. Equation (2.3) says that $\partial f(x)$ is the convex hull of all limits of gradients of f at differentiable points in an arbitrarily small neighborhood about x.

2.2. Manifolds of (1.1). The signs of the component functions F_i play a critical role in our focus on functions satisfying Assumption 1. We let *sgn* be the scalar function that returns the values 1, -1, and 0 for positive, negative, and null arguments, respectively, and we define $\mathbf{sign} : \mathbb{R}^n \to \{-1, 0, 1\}^r$ by

$$\mathbf{sign}\,(x) = \left[\ sgn(F_1(x)) \quad sgn(F_2(x)) \quad \cdots \quad sgn(F_r(x)) \ \right]^T.$$

We say that $\mathbf{sign}(x)$ returns the *sign pattern* of a point x. There exist 3^r possible sign patterns for any $x \in \mathbb{R}^n$; by indexing these possible sign patterns, we define $\mathbf{pat}^q \in \{-1, 0, 1\}^r$ for each $q \in \{1, \ldots, 3^r\}$.

For any $x \in \mathbb{R}^n$, its *manifold* $\mathcal{M}(x)$ is the maximal topologically connected set satisfying

$$\mathcal{M}(x) = \{y \in \mathbb{R}^n : \mathbf{sign}(y) = \mathbf{sign}(x)\}.$$

We define the union of all manifolds with the same sign pattern $\mathbf{pat}^q$ as a *manifold set* and denote it by $\mathcal{M}^q = \bigcup_{\{x : \mathbf{sign}(x) = \mathbf{pat}^q\}} \mathcal{M}(x)$. Letting $\mathcal{B}(x, \epsilon) = \{x : \|x\| \leq \epsilon\}$ denote the ball of radius ϵ around a point x, we say that a manifold set $\mathcal{M}^q$ is *active* at x provided that $\mathcal{M}^q \cap \mathcal{B}(x, \epsilon) \neq \emptyset$ for all $\epsilon > 0$.

We note that if Assumption 1 is satisfied, then the function f is locally Lipschitz. By using chain rule results (dependent on regularity conditions; see, e.g., [4, Definition 2.3.4]) that are ensured by Assumption 1, the subdifferential of f at a point x is

$$\partial f(x) = \sum_{i : F_i(x) \neq 0} sgn(F_i(x)) \nabla F_i(x) + \sum_{i : F_i(x) = 0} \mathbf{co}\left\{-\nabla F_i(x), \nabla F_i(x)\right\}, \tag{2.4}$$

where addition is understood to be setwise. Consequently, f is differentiable at x if and only if $F_i(x) = 0$ implies $\nabla F_i(x) = 0$ for $i \in \{1, \ldots, r\}$. This observation motivates our definition of the *nondifferentiable set of* F_i, given by

$$\mathcal{D}_i^c = \{x \in \mathbb{R}^n : F_i(x) = 0 \text{ and } \nabla F_i(x) \neq 0\}. \tag{2.5}$$

Note that for a function satisfying Assumption 1, $\mathcal{D}^c = \cup_{i=1}^r \mathcal{D}_i^c$. Our algorithmic framework in Section 3 is predicated on a relaxation of $\mathcal{D}_i^c$ that does not require the (exact) derivative $\nabla F_i(x)$.

2.3. Related Work. The analogue to steepest descent for nonsmooth optimization involves steps along the negative of the minimum-norm element of the subdifferential, $-\mathbf{proj}\left(0, \partial f(x^k)\right)$. The algorithm that we propose is related to cutting-plane and bundle methods [14], in that the subdifferential in this step is approximated by a finite set of generators.

A class of methods for nonsmooth optimization related to our proposed algorithm is that of gradient sampling. Such methods exploit situations when the underlying nonsmooth function is differentiable almost everywhere by using local gradient information around a current iterate to build a stabilized descent direction [3]. Kiwiel proposes gradient sampling variants in [16], including an approach that performs a line search within a sampling trust region. Such methods use the fact that, under weak regularity conditions (such as those given in Assumption 1), the closure of the set $\mathbf{co}\left\{\nabla f(y) : y \in \mathcal{B}(x^k, \Delta) \cap \mathcal{D}\right\}$ is a superset of $\partial f(x^k)$. Gradient sampling methods employ a finite sample set $\{y^1, \ldots, y^p\} \subset \mathcal{B}(x^k, \Delta) \cap \mathcal{D}$ and approximate $\partial f(x^k)$ by $\mathbf{co}\{\nabla f(y^1), \ldots, \nabla f(y^p)\}$. The negative of the minimum-norm element of this latter set is used as the search direction.

Constructing this set is more difficult when ∇f is unavailable. A nonderivative version of the gradient sampling algorithm is shown in [17] to converge almost surely to a first-order stationary point. However, the analysis depends on the use of a Gupal estimate of Steklov averaged gradients as a gradient approximation. Such an approach requires $2n$ function evaluations to compute each approximate gradient. In effect, a single gradient approximation is as expensive to compute as a central-difference gradient approximation, and the approximations must be computed for each point near the current iterate. With this motivation, Hare and Nutini [13] propose an approximate gradient-sampling algorithm that uses standard (e.g., central difference, simplex) gradient approximations to solve finite minimax problems of the composite form minimize $\max_i F_i(x)$. Similar to our focus on the form (1.1), Hare and Nutini [13] model the smooth component functions F_i and use the particular, known structure of their subdifferential $\mathbf{co}\left\{\nabla F_j(x) : j \in \arg\max_i F_i(x)\right\}$ within their convergence analysis. Both [13] and [17] employ a line search strategy for globalization. Our method employs a trust-region framework that links the trust-region radius with the norm of a model gradient, which can serve as a stationarity measure.

Other methods also exploit the general structure in composite nonsmooth optimization problems (1.2). When the function h is convex, typical trust-region-based approaches (e.g., [2, 8, 9, 23]) solve the *nonsmooth* subproblem

$$\text{minimize}\left\{h\left(F(x^k) + \langle s, \nabla M(x^k)\rangle\right) : s \in \mathcal{B}(x^k, \Delta)\right\}. \tag{2.6}$$

The model M is a Taylor expansion of F when derivatives are available. In this case, Griewank et al. [12] propose building piecewise linear models using ∇F from recent function evaluations. These nonsmooth models are then minimized to find future iterates and ∇F information for subsequent models. When derivatives are unavailable, recent work has used sufficiently accurate models of the Jacobian [10, 11]. We follow a similar approach in our approximation of ∇F but employ a fundamentally different subproblem, locally minimizing smooth models related to (1.3). In contrast to (2.6), our subproblem does not rely on the convexity of h.

There also exist derivative-free methods for nonsmooth optimization unrelated to gradient sampling. For example, a generalization of mesh adaptive direct search [1] finds descent directions for nonsmooth problems by generating an asymptotically dense set of search directions. Similar density requirements exist for general direct

search methods [20]. One of the derivative-free methods in [10] employs a smoothing function $f_\mu(x)$ parameterized by a smoothing parameter $\mu > 0$ satisfying

$$\lim_{z \to x, \mu \downarrow 0} f_\mu(z) = f(x),$$

for any $x \in \mathbb{R}^n$. By iteratively driving $\mu \to 0$ within a trust-region framework, the authors prove convergence to Clarke stationary points and provide convergence rates. We note that the Steklov averaged gradients in [17] are also essentially smoothing convolution kernels. Our proposed method requires neither a dense set of search directions nor smoothing parameters.

3. Manifold Sampling Algorithm. In order to prove convergence of our algorithm, the models m^{F_i} must sufficiently approximate F_i in a neighborhood of x^k. We require the models to be *fully linear* in the trust region, a notion formalized in the following assumption.

ASSUMPTION 2. *For $i \in \{1, \dots, r\}$, let m^{F_i} denote a twice continuously differentiable model intended to approximate F_i on some $\mathcal{B}(x, \Delta)$. For each $i \in \{1, \dots, r\}$, for all $x \in \mathbb{R}^n$, and for all $\Delta > 0$, there exist constants $\kappa_{i,\text{ef}}$ and $\kappa_{i,\text{eg}}$, independent of x and Δ, so that*

$$\begin{aligned} \left|F_i(x+s) - m^{F_i}(x+s)\right| &\leq \kappa_{i,\text{ef}}\Delta^2 \quad \forall s \in \mathcal{B}(0,\Delta) \\ \left\|\nabla F_i(x+s) - \nabla m^{F_i}(x+s)\right\| &\leq \kappa_{i,\text{eg}}\Delta \quad \forall s \in \mathcal{B}(0,\Delta). \end{aligned}$$

Furthermore, for $i \in \{1, \dots, r\}$ there exists $\kappa_{i,\text{mh}}$ so that $\|\nabla^2 m^{F_i}(x)\| \leq \kappa_{i,\text{mh}}$ for all $x \in \mathbb{R}^n$. For these constants, define $\kappa_\text{f} = \sum_{i=1}^r \kappa_{i,\text{ef}}$, $\kappa_\text{g} = \sum_{i=1}^r \kappa_{i,\text{eg}}$, and $\kappa_\text{mh} = \sum_{i=1}^r \kappa_{i,\text{mh}}$.

Assumption 2 is nonrestrictive, and one can derive classes of models satisfying the assumption both when ∇F_i is available inexactly and when ∇F_i is unavailable. For instance, in the latter case, the assumption holds when m^{F_i} is a linear model interpolating F_i at a set of sufficiently affinely independent points (see, e.g., [5, Chapter 10]); in this case, $\kappa_{i,\text{ef}}$ and $\kappa_{i,\text{eg}}$ scale with n and the respective Lipschitz constants of m^{F_i} and ∇m^{F_i}. If ∇F_i is available inexactly, then a model m^{F_i} that uses the inexact gradient while still satisfying Assumption 2 is a suitable model as well.

3.1. Algorithmic Framework. We now outline our algorithm, which samples manifolds (as opposed to gradients or gradient approximations) in order to approximate the subdifferential $\partial f(x^k)$. When x^k is changed, the r component function values $F_i(x^k)$ are computed, immediately yielding $\mathbf{sign}(x^k)$. Then, r component models, m^{F_i}, approximating F_i near x^k are built. Using the value of $\mathbf{sign}(x^k)$, we infer a set of generators, $\mathfrak{G}^k$, using the manifolds that are potentially active at x^k. The set of generators contains information about the manifolds active at (or around) the current iterate; our procedures (Algorithm 2 and Algorithm 3) for constructing $\mathfrak{G}^k$ are detailed in Section 3.2. The set $\mathbf{co}\left(\mathfrak{G}^k\right)$ is then used as an approximation to $\partial f(x^k)$.

We let g^k denote the minimum-norm element of $\mathbf{co}\left(\mathfrak{G}^k\right)$,

$$g^k = \mathbf{proj}\left(0, \mathbf{co}\left(\mathfrak{G}^k\right)\right), \tag{3.1}$$

which can be calculated by solving the quadratic optimization problem

$$\text{minimize}\left\{\frac{1}{2}\lambda^T (G^k)^T G^k \lambda : \; e^T\lambda = 1, \, \lambda \geq 0\right\}, \tag{3.2}$$

where the columns of G^k are the generators in $\mathfrak{G}^k$. A solution λ^* to (3.2) is a set of weights on the subgradient approximations that minimize $\|G\lambda\|^2$. That is, $g^k = G^k\lambda^*$.

Suppose there are $t \leq 3^r$ generators in $\mathfrak{G}^k$ and define the matrix

$$P^k = \begin{bmatrix} | & & | \\ \mathbf{pat}^1 & \cdots & \mathbf{pat}^t \\ | & & | \end{bmatrix}.$$

Then, since the qth generator in $\mathfrak{G}^k$ (alternatively, the qth column of G^k) is given by $\nabla M(x^k)\mathbf{pat}^q$, we have $G^k = \nabla M(x^k)P^k$. Thus, to maintain the property that the master model gradient is the optimal solution to (3.1), we consider a set of weights $w^k = P^k\lambda^*$ and define a smooth *master model* $m_k^f : \mathbb{R}^n \to \mathbb{R}$,

$$m_k^f(x) = \sum_{i=1}^{r} w_i^k m^{F_i}(x). \tag{3.3}$$

We make a few observations about this choice of w^k. Firstly, as intended by construction,

$$\nabla m_k^f(x^k) = \sum_{i=1}^{r} w_i^k \nabla m^{F_i}(x^k) = \sum_{i=1}^{r} \nabla m^{F_i}(x^k)(P^k\lambda^*)_i = \nabla M(x^k)P^k\lambda^* = G^k\lambda^* = g^k.$$

Secondly, $w_i^k \in [-1, 1]$ for each $i \in \{1, \ldots, t\}$ due to the constraints on λ in (3.2). Thirdly, notice that if G^k contains exactly one generator (i.e., $t = 1$), then $\lambda^* = 1$ is the trivial solution to (3.2) and so the master model in this case is simply

$$m_k^f(x) = \sum_{i=1}^{r} (\mathbf{sign}(x^k))_i m^{F_i}(x) = \langle M(x), \mathbf{sign}(x^k)\rangle.$$

In the kth iteration, the master model will be used in the trust-region subproblem

$$\text{minimize} \left\{ m_k^f(x^k + s) : s \in \mathcal{B}(0, \Delta_k) \right\}. \tag{3.4}$$

Provided the solution $x^k + s^k$ of (3.4) belongs to a manifold that was included in the construction of g^k, we apply a ratio test to determine the successfulness of the proposed step, as in a standard trust-region method. If the manifold containing $x^k + s^k$ is not contributing a generator in $\mathfrak{G}^k$, we augment $\mathfrak{G}^k$ and construct a new g^k. Since there are finitely many manifolds (at most 3^r in the case of (1.1)), this process of adding to $\mathfrak{G}^k$ will terminate.

Our ratio test quantity

$$\rho_k = \frac{\langle F(x^k), \mathbf{sign}(x^k + s^k)\rangle - \langle F(x^k + s^k), \mathbf{sign}(x^k + s^k)\rangle}{\langle M(x^k), \mathbf{sign}(x^k + s^k)\rangle - \langle M(x^k + s^k), \mathbf{sign}(x^k + s^k)\rangle} \tag{3.5}$$

is different from the usual ratio test of actual reduction to predicted reduction in that it considers the function decrease from the perspective of the manifold of the trial step $x^k + s^k$. In particular, our numerator is more conservative than the actual reduction since

$$\begin{aligned} &\langle F(x^k) - F(x^k + s^k), \mathbf{sign}(x^k + s^k)\rangle - \big(f(x^k) - f(x^k + s^k)\big) \\ &= \langle F(x^k) - F(x^k + s^k), \mathbf{sign}(x^k + s^k)\rangle \\ &\quad - \big(\langle F(x^k), \mathbf{sign}(x^k)\rangle - \langle F(x^k + s^k), \mathbf{sign}(x^k + s^k)\rangle\big) \\ &= \langle F(x^k), \mathbf{sign}(x^k + s^k) - \mathbf{sign}(x^k)\rangle \leq 0, \end{aligned}$$

where the inequality holds because $F_i(x^k)[\mathbf{sign}(x^k)]_i = |F_i(x^k)| \geq u_i F_i(x^k)$ for any i and for all $u_i \in [-1, 1]$.

We now state our framework in Algorithm 1, which includes assumptions on algorithmic parameters. We note that an input to the algorithm is a parameter $\kappa_{\mathrm{mh}} \geq 0$ bounding the curvature of the component models. It is always possible to construct fully linear models that are linear (i.e., $\nabla^2 m^{F_i} = 0$); see [22, Algorithm 2 and Theorem 4.5] for an example of how a linear fully linear model can be augmented into a nonlinear fully linear model with bounded curvature. Consequently, the choice of the parameter κ_{mh} does not affect our ability to construct fully linear models.

Algorithm 1: Manifold sampling.

1 Set parameters $\eta_1 \in (0,1)$, $\kappa_{\mathrm{mh}} \geq 0$, $\frac{1}{\eta_2} \in (\kappa_{\mathrm{mh}}, \infty)$, $\kappa_{\mathrm{d}} \in (0,1)$, $\gamma_{\mathrm{dec}} \in (0,1)$, and $\gamma_{\mathrm{inc}} \geq 1$
2 Choose initial iterate x^0 and trust-region radius $\Delta_0 > 0$; set $k = 0$
3 **while true do**
4 For each F_i, build a model m^{F_i} that is fully linear in $\mathcal{B}(x^k, \Delta_k)$ and satisfies $\sum_{i=1}^r \|\nabla^2 m^{F_i}(x)\| \leq \kappa_{\mathrm{mh}}$ for all $x \in \mathbb{R}^n$
5 Build a set of generators $\mathfrak{G}^k$ (from Algorithm 2 or Algorithm 3)
6 **while true do**
7 Build master model m_k^f using the models m^{F_i} and (3.3)
8 **if** $\Delta_k \geq \eta_2 \|\nabla m_k^f(x^k)\|$ **then**
9 **break** (**go to** Line 20)
10 Approximately solve (3.4) to obtain s^k
11 Evaluate $f(x^k + s^k)$
12 **if** $\nabla M(x^k)\mathbf{sign}(x^k + s^k) \in \mathfrak{G}^k$ **then**
13 **if** $x^k + s^k$ *satisfies* (3.6) **then**
14 **break** (**go to** Line 20)
15 **else**
16 $s^k \leftarrow -\kappa_{\mathrm{d}}^{j^*} \Delta_k \frac{\nabla M(x^k)\mathbf{sign}(x^k+s^k)}{\|\nabla M(x^k)\mathbf{sign}(x^k+s^k)\|}$ for j^* defined in (3.7)
17 **go to** Line 11
18 **else**
19 $\mathfrak{G}^k \leftarrow \mathfrak{G}^k \cup \nabla M(x^k)\mathbf{sign}(x^k + s^k)$
20 **if** $\Delta_k < \eta_2 \|\nabla m_k^f(x^k)\|$ (*acceptable* iteration) **then**
21 Update ρ_k through (3.5)
22 **if** $\rho_k > \eta_1$ (*successful* iteration) **then**
23 $x^{k+1} \leftarrow x^k + s^k$, $\Delta_{k+1} \leftarrow \gamma_{\mathrm{inc}} \Delta_k$
24 **else**
25 $x^{k+1} \leftarrow x^k$, $\Delta_{k+1} \leftarrow \gamma_{\mathrm{dec}} \Delta_k$
26 **else**
27 $x^{k+1} \leftarrow x^k$, $\Delta_{k+1} \leftarrow \gamma_{\mathrm{dec}} \Delta_k$
28 $k \leftarrow k + 1$

It is necessary for convergence that the steps s^k achieve a Cauchy-like decrease; however, unlike in a typical trust-region method, this sufficient decrease is needed in the denominator of ρ_k, and not in the step obtained from minimizing (3.4). That is,

we require

$$\langle M(x^k) - M(x^k + s^k), \mathbf{sign}(x^k + s^k)\rangle \geq \frac{\kappa_{\rm d}}{2}\|g^k\| \min\left\{\Delta_k, \frac{\|g^k\|}{\kappa_{\rm mh}}\right\}. \tag{3.6}$$

Notice that the solution to (3.4) does not necessarily satisfy (3.6). However, a Cauchy-like point given in the following lemma always satisfies (3.6).

LEMMA 1. *For any* $\mathbf{pat}_q$ *satisfying* $\nabla M(x^k)\mathbf{pat}_q \in \mathfrak{G}_k$, *if* $M(\cdot)\mathbf{pat}_q$ *is twice continuously differentiable,* $\kappa_{\rm mh} \geq \max_{s\in\mathcal{B}(0,\Delta_k)} \left\|\nabla^2 M(x^k + s)\mathbf{pat}_q\right\|$, $\kappa_{\rm mh} > 0$, $\left\|\nabla M(x^k)\mathbf{pat}_q\right\| > 0$, *and* $\kappa_{\rm d} \in (0, 1)$, *then setting* $s^k = -\kappa_{\rm d}^{j^*}\Delta_k \frac{\nabla M(x^k)\mathbf{pat}_q}{\left\|\nabla M(x^k)\mathbf{pat}_q\right\|}$, *for*

$$j^* = \max\left\{0, \left\lceil \log_{\kappa_{\rm d}}\left(\frac{\left\|\nabla M(x^k)\mathbf{pat}_q\right\|}{\Delta_k}\kappa_{\rm mh}\right)\right\rceil\right\}, \tag{3.7}$$

satisfies $\left\|s^k\right\| \leq \Delta_k$ *and* (3.6).

Proof. The result follows immediately from [21, Lemma 4.2] and the fact that because $\nabla M(x^k)\mathbf{pat}_q \in \mathfrak{G}_k$, we must have $\|\nabla M(x^k)\mathbf{pat}_q\| \geq \|g^k\|$. ∎

Iteratively constructing the master model and identifying manifolds gives a metric $\|g^k\|$ to measure progress to Clarke stationarity, but each iteration of Algorithm 1 seeks sufficient decrease only in a manifold that lies in an approximate steepest descent direction. Note that any looping introduced by Line 17 in Algorithm 1 will terminate due to there being a finite number of manifolds in $\mathcal{B}(x^k, \Delta_k)$ (at most 3^r) and Lemma 1.

3.2. Generator Sets. We complete our description of our manifold sampling algorithm by showing how the set of generators $\mathfrak{G}^k$ is built so that $\mathbf{co}\left(\mathfrak{G}^k\right)$ approximates $\partial f(x^k)$. Given the definition of the Clarke subdifferential in (2.3) and the known form of the subdifferential in (2.4), we know that the extreme points of $\partial f(x)$ must be the limits of sequences of gradients at differentiable points from manifolds that are active at x. Therefore, the extreme points of $\partial f(x)$ are a subset of $\nabla F(x)\mathbf{pat}^q$ over $q \in \{1, \ldots, 3^r\}$.

An approximation

$$\nabla M(x) = \left[\nabla m^{F_1}(x), \ldots, \nabla m^{F_r}(x)\right]$$

of the Jacobian ∇F induces an approximation $\nabla M(x)\mathbf{pat}^q$ to $\nabla F(x)\mathbf{pat}^q$ for any $q \in \{1, \ldots, 3^r\}$. Since ∇F_i may not be known exactly, we relax the dependence on ∇F_i in (2.5) and consider $-\nabla m^{F_i}(x)$ and $\nabla m^{F_i}(x)$ for each i for which $F_i(x) = 0$.

This is the motivation for our first procedure for forming the generator set $\mathfrak{G}^k$. Algorithm 2 initializes $\mathfrak{G}^k$ with $\nabla M(x^k)\mathbf{sign}(x^k)$ and then triples the size of $\mathfrak{G}^k$ for each i satisfying $sgn(F_i(x^k)) = 0$ and $\nabla F_i(x^k) \neq 0$. Our analysis of Algorithm 1 will show that this strategy can be used to approximate the subdifferential $\partial f(x^k)$.

We also propose a second approach for constructing $\mathfrak{G}^k$ that uses sign patterns of points near x^k and not just $\mathbf{sign}(x^k)$. Although this approach is inspired by gradient sampling, we note that we are not approximating the gradient at any point other than x^k. We naturally extend our definition of active manifolds by saying that a manifold set $\mathcal{M}^q$ is active in a set $\mathcal{S}$ provided there exists $x \in \mathcal{S}$ such that $\mathcal{M}^q$ is active at x. We denote such a *sample set* at iteration k by $Y(x^k, \Delta_k) \subset \mathcal{B}(x^k, \Delta_k)$. This set can come, for example, from the set of points previously evaluated by the algorithm that lie within a distance Δ_k of x^k.

Algorithm 2: Forming generator set $\mathfrak{G}^k$ using possibly active manifolds at x^k.

Input: x^k and $\nabla M(x^k)$
$\mathfrak{G}^k \leftarrow \{\nabla M(x^k)\mathbf{sign}(x^k)\}$
for $i = 1, \ldots, r$ **do**
  if $sgn(F_i(x^k)) = 0$ **then**
    $\mathfrak{G}^k \leftarrow \mathfrak{G}^k \cup \{\mathfrak{G}^k + \nabla m^{F_i}(x^k)\} \cup \{\mathfrak{G}^k - \nabla m^{F_i}(x^k)\}$

We can now state Algorithm 3, which constructs generators based on the set of manifolds active in $Y(x^k, \Delta_k)$. Intuitively, this additional manifold information obtained from sampling can "warn" the algorithm about sudden changes in gradient behavior that may occur within the current trust region.

Algorithm 3: Forming generator set $\mathfrak{G}^k$ using possibly active manifolds in $Y(x^k, \Delta_k)$.

Input: x^k, $\nabla M(x^k)$, and $Y(x^k, \Delta_k) = \{x^k, y^2, \ldots, y^p\}$
Initialize $\mathfrak{G}^k$ using Algorithm 2 with inputs x^k and $\nabla M(x^k)$
for $j = 2, \ldots, p$ **do**
  $\mathfrak{G}^k = \mathfrak{G}^k \cup \nabla M(x^k)\mathbf{sign}(y^j)$

Other reasonable approaches for constructing $\mathfrak{G}^k$ exist. For our analysis, a requirement for $\mathfrak{G}^k$ is given in Assumption 3.

ASSUMPTION 3. *Given x^k and $\Delta_k > 0$, the constructed set $\mathfrak{G}^k$ satisfies*

$$\left\{\nabla M(x^k)\mathbf{sign}(x^k) + \sum_{i:[\mathbf{sign}(x^k)]_i=0} t_i \nabla m^{F_i}(x^k) \,:\, t \in \{-1,0,1\}^r\right\} \subseteq \mathfrak{G}^k,$$

$$\mathfrak{G}^k \subseteq \left\{\nabla M(x^k)\mathbf{sign}(y) + \sum_{i:[\mathbf{sign}(y)]_i=0} t_i \nabla m^{F_i}(x^k) \,:\, t \in \{-1,0,1\}^r, y \in \mathcal{B}(x^k;\Delta_k)\right\}.$$

Clearly, a set $\mathfrak{G}^k$ produced by Algorithm 2 or Algorithm 3 will satisfy Assumption 3. Furthermore, any generator set satisfying Assumption 3 has $|\mathfrak{G}^k| \leq 3^r$.

4. Analysis.

We now analyze Algorithm 1.

4.1. Preliminaries.

We first show a result linking elements in a set similar to the form of $\mathfrak{G}^k$ to the subdifferentials of f at nearby points. Subsequent results will establish cases when our construction of the generator set $\mathfrak{G}^k$ satisfies the suppositions made in the statement of the lemma.

LEMMA 2. *Let Assumptions 1 and 2 hold, and let $x, y \in \mathbb{R}^n$ satisfy $\|x - y\| \leq \Delta_k$. Suppose that $T \subseteq T'$ for $T = \{\mathbf{pat}^{q_s} : s = 1, \ldots, j\}$ and $T' = \{\mathbf{pat}^{q'_{s'}} : s' = 1, \ldots, j'\}$ for*

$$\mathfrak{G} = \{\nabla M(x)p : p \in T\} \text{ and } \partial f(y) = \mathbf{co}\{\nabla F(y)p' : p' \in T'\}.$$

Then for each $g \in \mathbf{co}(\mathfrak{G})$, there exists $v(g) \in \partial f(y)$ satisfying

$$\|g - v(g)\| \leq (\kappa_{\mathrm{g}} + L)\Delta_k, \tag{4.1}$$

where κ_{g} and L are defined in Assumption 2 and Assumption 1, respectively.

Proof. Let $g \in \mathbf{co}\,(\mathfrak{G})$ be arbitrary. Since $\mathbf{co}\,(\mathfrak{G})$ is finitely generated (and thus compact and convex), g can be expressed as a positive convex combination of $N \leq n+1$ of its generators due to Caratheodory's theorem. Without loss of generality (by reordering as necessary), let these generators be the first N elements in $\mathfrak{G}$. That is, there exist $\lambda_{q_1}, \ldots, \lambda_{q_N} \in (0, 1]$ with $\sum_{s=1}^{N} \lambda_{q_s} = 1$ so that

$$g = \sum_{s=1}^{N} \lambda_{q_s} \nabla M(x) \mathbf{pat}^{q_s}. \tag{4.2}$$

By supposition, $\nabla F(y_s)\mathbf{pat}^{q_s} \in \partial f(y_s)$ for $s = 1, \ldots, N$. Since $\partial f(y)$ is convex, we have that $v(g) \in \partial f(y)$, where $v(g)$ is defined as

$$v(g) = \sum_{s=1}^{N} \lambda_{q_s} \nabla F(y) \mathbf{pat}^{q_s},$$

using the same λ_{q_s} as in (4.2) for $s = 1, \ldots, N$. Observe that for each s,

$$\begin{aligned} \|\nabla M(x)\mathbf{pat}^{q_s} - \nabla F(y)\mathbf{pat}^{q_s}\| &= \|(\nabla M(x) - \nabla F(x) + \nabla F(x) - \nabla F(y))\mathbf{pat}^{q_s}\| \\ &\leq \|\nabla M(x) - \nabla F(x)\| + \|\nabla F(x) - \nabla F(y)\| \\ &\leq (\kappa_{\mathrm{g}} + L)\Delta_k. \end{aligned}$$

Applying the definitions of g and $v(g)$ and recalling that $\sum_{s=1}^{N} \lambda_{q_s} = 1$ yields the expression (4.1). ∎

The approximation property in Lemma 2 can be used to motivate the use of the master model gradient in (3.1); as we shall see in Section 4.2, descent directions for the smooth master model will eventually identify descent directions for the nonsmooth function f.

4.2. Analysis of Algorithm 1. The next lemma demonstrates that because the master model gradient is chosen as g^k from (3.1), the sufficient decrease condition in (3.6) ensures a successful iteration, provided Δ_k is sufficiently small.

LEMMA 3. *Let Assumptions 1 and 2 hold. If*

$$\Delta_k < \min\left\{\frac{\kappa_{\mathrm{d}}(1-\eta_1)}{4\kappa_{\mathrm{f}}}, \eta_2\right\} \|g^k\|, \tag{4.3}$$

then iteration k of Algorithm *1 is successful.*

Proof. Notice that, whenever $g^k \neq 0$, the bound on Δ_k is positive by the algorithmic parameter assumptions in Algorithm 1 and that $\Delta_k < \eta_2 \|g^k\|$ ensures that the iteration is acceptable. Suppressing superscripts on x^k, s^k, and w^k for space, the definition of ρ_k in (3.5) yields

$$\begin{aligned} &|\rho_k - 1| \\ &= \left| \frac{\langle F(x), \mathbf{sign}(x+s)\rangle - \langle F(x+s), \mathbf{sign}(x+s)\rangle}{\langle M(x), \mathbf{sign}(x+s)\rangle - \langle M(x+s), \mathbf{sign}(x+s)\rangle} - 1 \right| \\ &= \left| \frac{\langle F(x) - M(x), \mathbf{sign}(x+s)\rangle - \langle F(x+s) - M(x+s), \mathbf{sign}(x+s)\rangle}{\langle M(x) - M(x+s), \mathbf{sign}(x+s)\rangle} \right|. \end{aligned} \tag{4.4}$$

The numerator of (4.4) satisfies

$$\begin{aligned}
&|\langle F(x) - M(x), \mathbf{sign}(x+s)\rangle - \langle F(x+s) - M(x+s), \mathbf{sign}(x+s)\rangle| \\
&\leq |\langle F(x) - M(x), \mathbf{sign}(x+s)\rangle| + |\langle F(x+s) - M(x+s), \mathbf{sign}(x+s)\rangle| \\
&\leq 2\kappa_{\mathrm{f}}\Delta_k^2. \qquad (4.5)
\end{aligned}$$

By construction, the denominator of (4.4) always satisfies (3.6). Therefore, using (4.5) and (3.6) in (4.4) yields

$$|\rho_k - 1| \leq \frac{4\kappa_{\mathrm{f}}\Delta_k^2}{\kappa_{\mathrm{d}}\|g^k\| \min\left\{\Delta_k, \frac{\|g^k\|}{\kappa_{\mathrm{mh}}}\right\}} \leq \frac{4\kappa_{\mathrm{f}}\Delta_k}{\kappa_{\mathrm{d}}\|g^k\|} < 1 - \eta_1,$$

where the second inequality is implied by the algorithmic parameter assumption $\frac{1}{\eta_2} > \kappa_{\mathrm{mh}}$ and the last inequality is a result of (4.3). Thus, $\rho_k > \eta_1$, and iteration k is successful. ∎

The next result shows that the trust-region radius converges to zero.

LEMMA 4. *Let Assumptions 1 and 2 hold. If $\{x^k, \Delta_k\}$ is generated by* Algorithm *1, then $\Delta_k \to 0$.*

Proof. On successful iterations k, $\rho_k > \eta_1$ and thus,

$$\begin{aligned}
\langle F(x) - F(x+s), \mathbf{sign}(x+s)\rangle &> \eta_1 \left(\langle M(x) - M(x+s), \mathbf{sign}(x+s)\rangle\right) \\
&\geq \eta_1 \frac{\kappa_{\mathrm{d}}}{2}\|g^k\| \min\left\{\Delta_k, \frac{\|g^k\|}{\kappa_{\mathrm{mh}}}\right\} \\
&\geq \eta_1 \frac{\kappa_{\mathrm{d}}}{2}\|g^k\|\Delta_k \\
&> \frac{\eta_1\kappa_{\mathrm{d}}}{2\eta_2}\Delta_k^2,
\end{aligned}$$

where the second inequality follows from (3.6), and the last two inequalities follow from the algorithmic parameter assumption $\frac{1}{\eta_2} > \kappa_{\mathrm{mh}}$ and acceptability of all successful iterations. If there are infinitely many successful iterations, let $\{k_j\}$ index them. Notice that on any iteration,

$$\langle F(x^k), \mathbf{sign}(x^k + s^k)\rangle \leq \langle F(x^k), \mathbf{sign}(x^k)\rangle = f(x^k),$$

since, for any i, $F_i(x^k)[\mathbf{sign}(x^k)]_i = |F_i(x^k)| \geq t_i F_i(x^k)$ for all $t_i \in \{-1, 0, 1\}$.

Since f is bounded below by zero by Assumption 1 and $f(x^k)$ is nonincreasing in k, having infinitely many successful iterations implies that

$$\begin{aligned}
\infty > \sum_{j=0}^{\infty} f(x^{k_j}) - f(x^{k_j} + s^{k_j}) &> \sum_{j=0}^{\infty} \langle F(x^{k_j}) - F(x^{k_j} + s^{k_j}), \mathbf{sign}(x^{k_j} + s^{k_j})\rangle \\
&> \sum_{j=0}^{\infty} \frac{\eta_1\kappa_{\mathrm{d}}}{2} \left\|g^{k_j}\right\| \Delta_{k_j} \\
&> \sum_{j=0}^{\infty} \frac{\eta_1\kappa_{\mathrm{d}}}{2\eta_2}\Delta_{k_j}^2. \qquad (4.6)
\end{aligned}$$

Thus, $\Delta_{k_j} \to 0$ provided $\{k_j\}$ is an infinite subsequence of successful iterations. Since Δ_k increases by γ_{inc} on successful iterations, for any successful iterate k_i,

$\gamma_{\text{inc}}\Delta_{k_i} \geq \Delta_j \geq \Delta_{k_{i+1}}$ for all $k_i < j \leq k_{i+1}$. Therefore, $\Delta_k \to 0$ if the number of successful iterations is infinite.

If there are only finitely many successful iterations, then there is a last successful iteration $k_{j'}$, and the update rules of the algorithm will monotonically decrease Δ_k on all iterations $k > k_{j'}$.

Regardless of whether Algorithm 1 has an infinite or a finite number of successful iterations, $\Delta_k \to 0$. ☐

The next lemma is a liminf-type result for the master model gradients.

LEMMA 5. *Let Assumptions 1 and 2 hold. If $\{x^k, \Delta_k\}$ is generated by* Algorithm *1, then for all $\epsilon > 0$, there exists a $k(\epsilon)$ such that $\|g^{k(\epsilon)}\| \leq \epsilon$. That is, $\liminf_{k\to\infty} \|g^k\| = 0$.*

Proof. To arrive at a contradiction, suppose that there exist j and $\epsilon > 0$ so that for all $k \geq j$, $\|g^k\| \geq \epsilon$. By Lemma 3, since Algorithm 1 requires that $\nabla M(x^k)\mathbf{sign}(x^k + s^k) \in \mathfrak{G}^k$ before s^k can possibly be accepted, any iteration satisfying $\Delta_k \leq C\|g^k\|$ will be successful, where

$$C = \min\left\{\frac{\kappa_{\text{d}}(1-\eta_1)}{4\kappa_{\text{f}}}, \eta_2\right\}.$$

Hence, by the contradiction hypothesis, any $k \geq j$ satisfying $\Delta_k \leq C\epsilon$ is guaranteed to be successful and $\Delta_{k+1} = \gamma_{\text{inc}}\Delta_k \geq \Delta_k$. Therefore $\Delta_k \geq \gamma_{\text{dec}}C\epsilon$ for all k, contradicting Lemma 4. ☐

Before showing that every cluster point of $\{x^k\}$ is a Clarke stationary point, we recall basic terms and a theorem.

Motivated by the subdifferential operator $\partial f(x)$, we first formalize the notion of a limit superior of a set mapping (i.e., one that maps a vector to a set of vectors). For a set mapping $D : \mathbb{R}^n \to \mathbb{R}^n$, the *limit superior* of D as $x \to \bar{x}$ is defined by the set mapping

$$\limsup_{x\to\bar{x}} D(x) = \{y : \exists\{x^k : k \geq 1\} \to \bar{x} \text{ and } \{y^k : k \geq 1\} \to y \text{ with } y^k \in D(x^k)\}.$$

A set mapping D is said to be *outer semicontinuous at* $\bar{x}$ provided

$$\limsup_{x\to\bar{x}} D(x) = D(\bar{x}).$$

The following result is given as Proposition 7.1.4 in [7].

THEOREM 6. *If a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous, then the set mapping $\partial f(x)$ is everywhere outer semicontinuous.*

LEMMA 7. *Let Assumptions 1–3 hold and take $\{x^k, \Delta_k, g^k\}$ to be a sequence generated by* Algorithm *1. For any subsequence of acceptable iterations $\{k_j\}$ such that*

$$\lim_{j\to\infty} \|g^{k_j}\| = 0,$$

and $\left\{x^{k_j}\right\} \to x^*$ for some point x^*, then $0 \in \partial f(x^*)$.

Proof. Since $\Delta_k \to 0$ by Lemma 4 and $\left\{x^{k_j}\right\}$ converges to x^*, for k sufficiently large, only manifolds active at x^* are in $\mathfrak{G}^k$ by Assumption 3. Setting T (in Lemma 2) to be the sign patterns in $\mathfrak{G}^k$ and T' to be the sign patterns active at $\partial f(x^*)$, Assumption 3 and Lemma 4 thus guarantee $T \subseteq T'$ when k is sufficiently large. Therefore, by Lemma 2, there exists $v(g^{k_j}) \in \partial f(x^*)$ for each g^{k_j} so that

$$\|g^{k_j} - v(g^{k_j})\| \leq (\kappa_{\text{g}} + L)\Delta_{k_j}.$$

Thus, by the acceptability of every iteration indexed by k_j,

$$\|g^{k_j} - v(g^{k_j})\| \leq (\kappa_{\mathrm{g}} + L)\eta_2 \|g^{k_j}\|,$$

and so

$$\|v(g^{k_j})\| \leq (1 + (\kappa_{\mathrm{g}} + L)\eta_2)\, \|g^{k_j}\|.$$

Since $\left\|g^{k_j}\right\| \to 0$ by assumption, therefore $v(g^{k_j}) \to 0$. Since ∂f is everywhere outer semicontinuous by Theorem 6, $0 \in \partial f(x^*)$. ∎

We now prove the promised result.

THEOREM 8. *Let Assumptions 1–3 hold. If x^* is a cluster point of a sequence $\{x^k\}$ generated by* Algorithm *1, then $0 \in \partial f(x^*)$.*

Proof. Suppose that there are only finitely many successful iterations, with k' being the last.

To establish a contradiction, suppose $0 \notin \partial f(x^{k'})$. By the continuity of each component F_i granted by Assumption 1, there exists $\bar{\Delta} > 0$ so that for all $\Delta \in [0, \bar{\Delta})$, the manifold sets active in $\mathcal{B}(x^{k'}, \bar{\Delta})$ are precisely the manifold sets active at $x^{k'}$; that is,

$$\left\{ \lim_{y^j \to x^{k'}} \mathbf{sign}(y^j) : \lim_{j\to\infty} y^j = x^{k'},\ \{y^j : j \geq 1\} \subset \mathcal{M}^q \right\} = \left\{ \mathbf{sign}(y) : y \in \mathcal{B}(x^{k'}, \Delta) \right\}.$$

for all $\Delta \leq \bar{\Delta}$.

Since every iteration after k' is assumed to be unsuccessful, $\Delta_{k'}$ decreases by a factor of γ_{dec} in each subsequent iteration and there exists a least $k'' \geq k'$ so that $\Delta_{k''} < \bar{\Delta}$. Therefore, by Assumption 3, $\nabla M(x^k)\mathbf{sign}(x^k + s^k) \in \mathfrak{G}^k$ holds the first time Line 12 of Algorithm 1 is reached in iteration $k \geq k''$. Consequently, the conditions for Lemma 2 hold; and thus, for each $k \geq k''$, there exists $v(g^k) \in \partial f(x^{k'})$ so that $\|v(g^k) - g^k\| \leq (\kappa_{\mathrm{g}} + L)\Delta_k$. By supposition, since $0 \notin \partial f(x^{k'})$, there is a nonzero minimum-norm element $v^* \in \partial f(x^{k'})$. We thus conclude the following:

$$\|g^k\| \geq \|v(g^k)\| - (\kappa_{\mathrm{g}} + L)\Delta_k \geq \|v^*\| - (\kappa_{\mathrm{g}} + L)\Delta_k \qquad \text{for all } k \geq k''. \tag{4.7}$$

Since every iteration after k'' is unsuccessful, Δ_k will decrease by a factor of γ_{dec} and $x^{k+1} = x^{k'}$ for each $k \geq k''$.

Define the constant

$$c = \|v^*\| \min\left\{ \frac{(1-\eta_1)}{4\frac{\kappa_{\mathrm{f}}}{\kappa_{\mathrm{d}}} + (\kappa_{\mathrm{g}} + L)(1-\eta_1)}, \frac{1}{\kappa_{\mathrm{mh}} + \kappa_{\mathrm{g}} + L} \right\}.$$

By Lemma 3, success is guaranteed within $t = \left\lceil \log_{\gamma_{\mathrm{dec}}} \frac{c}{\Delta_{k''}} \right\rceil$ many iterations after iteration k'' since (4.7) and the definition of c imply that

$$\Delta_{k''+t} \leq \min\left\{ \frac{\kappa_{\mathrm{d}}(1-\eta_1)}{4\kappa_{\mathrm{f}}}, \eta_2 \right\} \|g^{k''+t}\|.$$

This is a contradiction, thus proving the result when there are finitely many successful iterations.

Now suppose there are infinitely many successful iterations. We will demonstrate that there exists a subsequence of successful iterations $\{k_j\}$ that simultaneously satisfies both

$$x^{k_j} \to x^* \text{ and } \|g^{k_j}\| \to 0.$$

Suppose first that $x^k \to x^*$. Then, every subsequence $x^{k_j} \to x^*$, and so we can use the sequence of subgradient approximations $\{g^{k_j}\}$ from Lemma 5, and we have the desired subsequence.

Now, suppose $x^k \not\to x^*$. We will show that $\liminf \max\{\|x^k - x^*\|, \|g^k\|\} = 0$. We proceed by contradiction. The contradiction hypothesis, along with the definition of x^* being a cluster point of $\{x^k\}$, implies that there exists $\bar{\nu} > 0$ and iteration $\bar{k}$, so that for the infinite set $K = \{k : k \geq \bar{k}, \|x^k - x^*\| \leq \bar{\nu}\}$, $\{x^k\}_{k\in K} \to x^*$ and $\|g^k\| > \bar{\nu}$ for all $k \in K$. From (4.6), we have that

$$\frac{\eta_1 \kappa_{\mathrm{d}}}{2} \sum_{k\in K} \|g^k\| \|x^{k+1} - x^k\| \leq \frac{\eta_1 \kappa_{\mathrm{d}}}{2} \sum_{k=0}^{\infty} \|g^k\| \|x^{k+1} - x^k\| < \infty, \tag{4.8}$$

since on successful iterations, $\|x^{k+1} - x^k\| \leq \Delta_k$, while on unsuccessful iterations, $\|x^{k+1} - x^k\| = 0$. Since $\|g^k\| > \bar{\nu}$ for all $k \in K$, then we conclude from (4.8) that

$$\sum_{k\in K} \|x^{k+1} - x^k\| < \infty. \tag{4.9}$$

Since $x^k \not\to x^*$, there exists some $\hat{\nu} > 0$ so that for any $k_{\text{start}} \in K$ satisfying $\|x^{k_{\text{start}}} - x^*\| \leq \bar{\nu}$, there a first index $k_{\text{end}} > k_{\text{start}}$ satisfying $\|x^{k_{\text{end}}} - x^{k_{\text{start}}}\| > \hat{\nu}$ and $\{k_{\text{start}}, k_{\text{start}} + 1, \ldots, k_{\text{end}} - 1\} \subset K$.

By (4.9), for $\hat{\nu}$ there exists $N \in \mathbb{N}$ such that

$$\sum_{\substack{k\in K \\ k\geq N}} \left\|x^{k+1} - x^k\right\| \leq \hat{\nu}.$$

Taking $k_{\text{start}} \geq N$, by the triangle inequality, we have

$$\hat{\nu} < \|x^{k_{\text{end}}} - x^{k_{\text{start}}}\| \leq \sum_{i\in\{k_{\text{start}}, k_{\text{start}}+1, \ldots, k_{\text{end}}-1\}} \|x^{i+1} - x^i\| \leq \sum_{\substack{k\in K \\ k\geq N}} \left\|x^{k+1} - x^k\right\| \leq \hat{\nu}.$$

Thus, $\hat{\nu} < \hat{\nu}$, a contradiction, and therefore $\liminf \max\{\|x^k - x^*\|, \|g^k\|\} = 0$.

Therefore, by Lemma 7, in either the case where $x^k \to x^*$ or $x^k \not\to x^*$, there exists a subsequence of subgradients satisfying $\|v(g^{k_j})\| \to 0$ and $x^{k_j} \to x^*$. Since $\partial f(x^*)$ is outer semicontinuous, we have that $0 \in \partial f(x^*)$. ∎

4.3. Concerning Termination Certificates. One would hope that (on acceptable iterations), a small master model gradient norm would signal that necessary conditions for proximity to stationarity are satisfied, analogous to how a small gradient norm serves as such a signal in smooth optimization. Although this is not always so, we provide exact theoretical conditions under which a similar statement holds for Algorithm 1. This approach emulates a stopping criterion used in [13], which likewise cannot generally be shown to be a necessary condition of proximity to stationarity.

LEMMA 9. *Let Assumptions 1 and 2 hold. Suppose that at iteration k, Line 21 of* Algorithm *1 is reached with* $\|g^k\| < \epsilon$ *for some* $\epsilon > 0$ *and also let*

$\mathfrak{G}^k = \{\nabla M(x^k)\mathbf{pat}^{q_s} : s = 1, \ldots, j\}$ *be the generator set at iteration* k. *Additionally, suppose there exists* $y \in \mathcal{B}(x^k, \Delta_k)$ *so that* $\partial f(y) = \mathbf{co}\left\{\nabla F(y)\mathbf{pat}^{q'_{s'}} : s' = 1, \ldots, j'\right\}$ *with* $\{\mathbf{pat}^{q_1}, \ldots, \mathbf{pat}^{q_j}\} \subseteq \{\mathbf{pat}^{q'_1}, \ldots, \mathbf{pat}^{q'_{j'}}\}$. *Then,*

$$\min_{v \in \partial f(y)} \|v\| < (1 + (\kappa_{\mathrm{g}} + L)\eta_2)\epsilon,$$

where κ_{g} *is as in Assumption 2 and* L *is as in Assumption 1.*

Proof. By the algorithmic parameter assumptions in Algorithm 1, $\eta_2 > 0$. Since Line 21 is reached with $\|g^k\| < \epsilon$, it must be that $\frac{\Delta_k}{\eta_2} \leq \|g^k\| < \epsilon$. By Lemma 2 and the suppositions, there exists $v(g^k) \in \partial f(y)$ so that

$$\|g^k - v(g^k)\| \leq (\kappa_{\mathrm{g}} + L)\Delta_k < (\kappa_{\mathrm{g}} + L)\eta_2\epsilon.$$

Applying the triangle inequality yields

$$\|v(g^k)\| < (1 + (\kappa_{\mathrm{g}} + L)\eta_2)\epsilon,$$

from which the desired result follows. ∎

We note that the assumptions that the iteration is acceptable and that y lies within Δ_k of x^k directly tie the result to both the master model gradient $\|g^k\|$ and the trust-region radius Δ_k. A termination certificate consisting of these two quantities is analogous to the "optimality certificates" used in [3]. In both cases, the certificate can be interpreted as indicating that two of three necessary conditions are satisfied so that there is a point $y \in \mathcal{B}(x^k, \Delta_k)$ so that an element of $\partial f(y)$ is as small in norm as suggested in Lemma 9. The third necessary condition, which is not as straightforward to check, is that the algorithm's iterates have become sufficiently clustered around y so that the manifolds active at y are a superset of the manifolds active in $\mathcal{B}(x^k, \Delta_k)$.

We remark here that this dependence on knowing all the manifolds active in a given trust region is what makes a straightforward analysis of rates based on $\|g_k^f\|$ elusive. Therefore, a comparison of the theoretical worst-case complexity of Algorithm 1 with the rates proven for the nonsmooth methods in [10] is currently elusive.

5. Manifold Sampling as a Stochastic Algorithm. Thus far, no restrictions have been placed on the sample set $Y(x^k, \Delta_k)$, apart from its containment in $\mathcal{B}(x^k, \Delta_k)$. In this section, we consider what happens when $Y(x^k, \Delta_k)$ includes stochastically sampled points, a strategy that results in Section 6 show is fruitful when $\mathfrak{G}^k$ is built by using Algorithm 3.

If random points are added to $Y(x^k, \Delta_k)$, the sequence of generator sets $\{\mathfrak{G}^k : k \geq 1\}$ will be a realization of a random variable denoted $\{\tilde{\mathfrak{G}}^k : k \geq 1\}$. Consequently, the algorithm will be inherently stochastic; the sequence of iterates produced by Algorithm 1 will be random variables $\{\tilde{x}^k : k \geq 1\}$ with realizations $\{x^k : k \geq 1\}$. Similarly, we denote by $\{\tilde{\Delta}_k : k \geq 1\}$, $\{\tilde{s}^k : k \geq 1\}$, and $\{\tilde{g}^k : k \geq 1\}$ sequences of random trust-region radii, trial steps, and master model gradients with respective realizations $\{\Delta_k : k \geq 1\}$, $\{s^k : k \geq 1\}$, and $\{g^k : k \geq 1\}$.

We show in Theorem 10 that the results from Section 4 hold for *any* realization of Algorithm 1. Note that Assumption 1 is unaffected by stochasticity in $Y(x^k, \Delta_k)$ (and $\mathfrak{G}^k$ is similarly unaffected on any iteration). In particular, we note that Assumption 2 ensures that the quality of the component models holds in a deterministic fashion. Assumption 3 has the stochastic analogue of assuming that every iterate in any realization satisfies the deterministic Assumption 3. This is nonrestrictive since

Assumption 3 requires that $\mathfrak{G}^k$ satisfy conditions depending not on $Y(x^k, \Delta_k)$ but only on the sign patterns in $\mathcal{B}(x^k, \Delta_k)$, which is a deterministic set at any iteration.

THEOREM 10. *Let X^* denote the union of all cluster points x^* over all realizations $\{(x^k, \Delta_k, s^k, \mathfrak{G}^k, g^k) : k \geq 1\}$ in the σ-algebra generated by $\{(\tilde{x}^k, \tilde{\Delta}_k, \tilde{s}^k, \tilde{\mathfrak{G}}^k, \tilde{g}^k) : k \geq 1\}$. Let Assumptions 1 and 2 hold, let Assumption 3 hold for each (x^k, Δ_k) in any realization $\{(x^k, \Delta_k) : k \geq 1\}$, and let* Algorithm *1 be initialized with $(\tilde{x}^0, \tilde{\Delta}_0) = (x^0, \Delta_0)$. Then, for every $x^* \in X^*$, $0 \in \partial f(x^*)$.*

Proof. The proof follows the same argument as in Section 4.

Let $\{(x^k, \Delta_k, s^k, \mathfrak{G}^k, g^k) : k \geq 1\}$ be an arbitrary realization of the random sequence $\{(\tilde{x}^k, \tilde{\Delta}_k, \tilde{s}^k, \tilde{\mathfrak{G}}^k, \tilde{g}^k) : k \geq 1\}$ produced by the stochastic algorithm.

Lemma 2 is independent of the realization, and Theorem 6 holds independently of Algorithm 1. Lemma 3 holds deterministically for any k where Δ_k and $\|g^k\|$ (produced by $\mathfrak{G}^k$) satisfy $\Delta_k < C\|g^k\|$.

Lemma 4 depends only on f being bounded below (a result of Assumption 1), and thus we get $\Delta_k \to 0$ for the arbitrary realization.

Lemma 5 holds if $Y(x^k, \Delta_k)$ is stochastic, thereby producing stochastic $\mathfrak{G}^k$, because the realization $\{(\Delta_k, g^k) : k \geq 1\}$ having $\liminf_{k\to\infty} \|g^k\| \neq 0$ would similarly contradict $\Delta_k \to 0$ (Lemma 4).

We can now prove the theorem. Suppose $\{x^k : k \geq 1\}$ has a cluster point x^*. Then, having proved that all the lemmata hold for the arbitrary realization, a direct application of Theorem 8 to that particular realization gives us that $0 \in \partial f(x^*)$. Since the realization of $\{(\tilde{x}^k, \tilde{\Delta}_k, \tilde{s}^k, \tilde{\mathfrak{G}}_k, \tilde{g}^k) : k \geq 1\}$ was arbitrary, we have shown the desired result. ∎

6. Numerical Results. We now examine the performance of variations of the manifold sampling algorithm outlined in Algorithm 1. Throughout this section, we use $x^{(j,p,s)}$ to denote the jth point evaluated on a problem p by an optimization solver s. For derivative-based versions of Algorithm 1, such points correspond solely to the trust-region subproblem solutions (Line 10) and points possibly sampled when constructing the generator set (Line 5); for derivative-free versions, evaluated points may additionally include evaluations performed to ensure that the component models are fully linear in the current trust region (Line 4). We drop the final superscript (s) when the point is the same for all solvers.

6.1. Implementations. In our first tests, we focus on the derivative-free case, when only zeroth-order information (function values) of F is provided to a solver; we view such problems as a more challenging test of the manifold sampling algorithm, since the component model approximations are not directly obtained from derivative information.

In our implementations of Algorithm 1, the sampling set $Y(x^k, \Delta_k)$ is used for construction of both the component models and, when using Algorithm 3, the generator sets. All our implementations employ linear models, m^{F_i}, of each component function F_i. The linear models are constructed so that m^{F_i} interpolates F_i at x^k and is the least-squares regression model for the remainder of the sampling set, $Y(x^k, \Delta_k)\backslash x^k$. At the beginning of iteration k, we set $Y(x^k, \Delta_k)$ to be all points previously evaluated by the algorithm that lie in $\mathcal{B}(x^k, \Delta_k)$. If this results in an underdetermined interpolation (i.e., $\text{rank}\big(Y(x^k, \Delta_k) - x^k\big) < n$), then additional points are added to $Y(x^k, \Delta_k)$ as described below.

We tested four variants of Algorithm 1, which differ from one another in how they construct the generator set $\mathfrak{G}^k$ and how they add points to the sampling set $Y(x^k, \Delta_k)$:

Center Manifold Sampling (CMS): Uses Algorithm 2 to build the generator set $\mathfrak{G}^k$; this generator set does not depend on the sampling set $Y(x^k, \Delta_k)$, which is used solely for constructing the component models. For building these models, the set of scaled coordinate directions, $\{x^k+\Delta_k e_1, \ldots, x^k+\Delta_k e_n\}$, are added to the sample set $Y(x^k, \Delta_k)$ in cases of underdetermined interpolation.

Greedy Deterministic Manifold Sampling (GDMS): Uses Algorithm 3 to build the generator set $\mathfrak{G}^k$. Additional points are not added to the sample set $Y(x^k, \Delta_k)$ unless the linear regression is underdetermined. In the underdetermined case, $n-\text{rank}(Y(x^k, \Delta_k) - x^k)$ directions D in the null space of $Y(x^k, \Delta_k) - x^k$ are generated by means of a (deterministic) QR factorization. After evaluating F along these scaled directions, the associated points $x^k + \Delta_k D$ are added to the sample set $Y(x^k, \Delta_k)$.

Deterministic Manifold Sampling (DMS): Uses Algorithm 3 to build the generator set and adds scaled coordinate directions, $\{x^k + \Delta_k e_1, \ldots, x^k + \Delta_k e_n\}$, to the sample set $Y(x^k, \Delta_k)$ every iteration.

Stochastic Manifold Sampling (SMS): Uses Algorithm 3 to build the generator set and adds a set of n points randomly generated from a uniform distribution on $\mathcal{B}(x^k, \Delta_k)$ to the sample set $Y(x^k, \Delta_k)$ every iteration.

The strategy used to add points to the sample set $Y(x^k, \Delta_k)$ in the deterministic variants ensures the full linearity of the models required in Assumption 2. For the stochastic variant SMS, however, the realized sample set $Y(x^k, \Delta_k)$ results in models that do not necessarily satisfy Assumption 2. Consequently, Theorem 10 may not hold for our implementation since such a sample set does not guarantee that the realized models are fully linear (see, e.g., [5, 18]).

In all cases, the weights defining the master model in Line 7 of Algorithm 1 are calculated by solving the quadratic program (3.2) via the subproblem solver used in [6], which is based on a specialized active set method proposed in [15].

We compared the above variants with a modified version of the minimax method of Grapiglia et al. [11], which we denote GYY. We adjusted the code used in [11] to solve ℓ_1-problems by changing the nonsmooth subproblem linear program (2.6). We also adjusted stopping tolerances to prevent early termination: we decreased the minimum trust-region radius to 10^{-32} and removed the default criterion of stopping after 10 successive iterations without a decrease in the objective.

As a baseline, we also tested two trust-region algorithms for smooth optimization. The codes L-DFOTR and Q-DFOTR are implementations of the algorithm described in [5] using, respectively, linear and quadratic regression models defined by an appropriately sized, deterministic sample set. These implementations may not converge to a stationary point since they assume a smooth objective function, but they serve as important comparators since they are more efficient at managing their respective sample sets.

By design, all seven of the codes tested employ a trust-region framework, and thus the parameters across the methods can be set equal. The parameter constants were selected to be $\Delta_0 = \max\left\{1, \left\|x^{(0,p)}\right\|_\infty\right\}$, $\eta_1 = 0.25$, $\eta_2 = 1$, $\gamma_{\text{dec}} = 0.5$, and $\gamma_{\text{inc}} = 2$.

6.2. Test Problems. We consider the ℓ_1 test problems referred to as the "piecewise smooth" test set in [19]. This synthetic test set was selected in part because of the availability of the Jacobian $\nabla F(x)$ for each problem, and thus the subdifferential in (2.4); this is useful for benchmarking purposes. The set is composed of 53 problems of the form (1.1) ranging in dimension from $n = 2$ to $n = 12$, with the number of

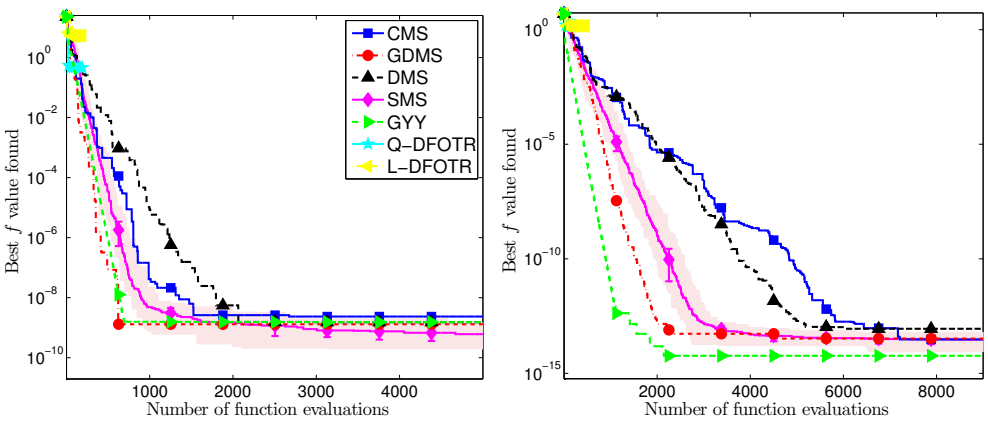

FIG. 1. *Sample trajectories of function values on problem 11 (left), which has $n = r = 4$, and problem 52 (right), which has $n = r = 8$. In* SMS*, 30 runs were performed: the upper band shows the largest function value obtained at the indicated number of function evaluations and the lower band shows the least function value obtained; the median values are indicated in the line segment connecting the 25th and 75th quantiles of the function values.*

component functions, r, ranging from n to 65.

A standard starting point, $x^{(0,p)}$, is provided for each problem p in the test set. We note that the objective f is nondifferentiable at $x^{(0,p)}$ for five problems (numbers 9, 10, 29, 30, and 52).

For all test problems, there is neither a guarantee that there is a unique minimizer x^* with $0 \in \partial f(x^*)$ nor a guarantee that $f(x^*) = f(y^*)$ for all x^*, y^* with $0 \in \partial f(x^*) \cap \partial f(y^*)$.

A budget of $1000(n+1)$ function evaluations was given to each solver for each n-dimensional problem. Solvers were terminated short of this budget only when Δ_k fell below 10^{-32}. We note, however, that for virtually every solver and problem, no successful iterations were found after Δ_k fell below 10^{-18}; this result is unsurprising given that the experiments were run in double precision.

Figure 1 shows typical trajectories of the best function value found on two problems where $\min f(x) = 0$. The sole stochastic solver (SMS) was run 30 times; Figure 1 shows that there is little variability in the value of the solution found by SMS across the 30 instances (on these two problems). This is not the case in Figure 2 (left), which shows the trajectory on problem 31; here we see that at least one instance of SMS finds a best function value different from that of the majority of SMS instances. In each of these instances, the smooth solvers L-DFOTR and Q-DFOTR struggle to find solutions with function values comparable to those found by the nonsmooth solvers.

6.3. Measuring Stationarity. The behavior seen in Figure 2 (left) suggests that the function values found by a solver may not indicate whether the solver has found a stationary point. We now measure the ability of a solver to identify points close to Clarke stationarity.

Lemma 9 does not guarantee that $(\Delta_k, \|g^k\|)$ provides a measure of stationarity. Instead, we will employ the stationarity measure used for nonsmooth composite optimization in [23] and more recently in [11]. This measure considers the maximum decrease obtained from directional linearizations of f at x,

$$\Psi(x) = \max_{d:\|d\|\leq 1} f(x) - \|F(x) + \nabla F(x)^T d\|_1. \tag{6.1}$$

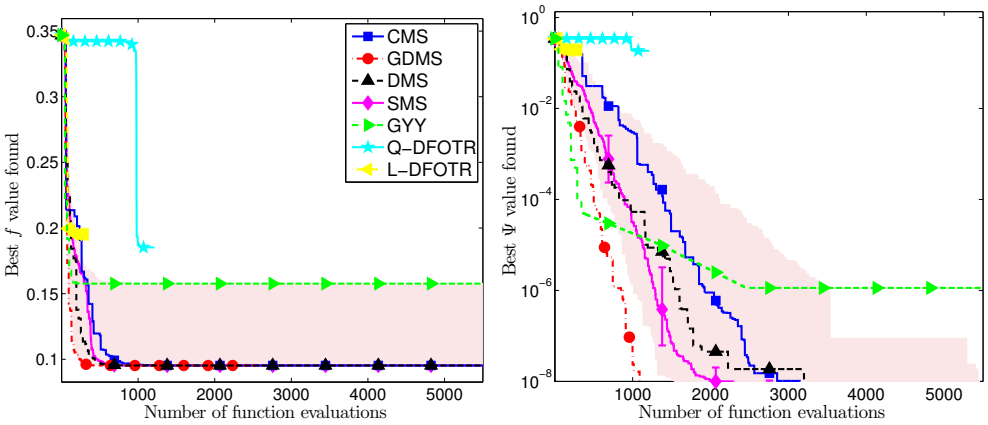

FIG. 2. *Best function value (left) and stationary measure $\Psi(x)$ (right) found in terms of the number of function evaluations performed for problem 31, a problem with $n = 8$ dimensions and $r = 8$ components. The stationary measure indicates that all instances of* SMS *find a stationary point, despite the function values associated with these stationary points being different.*

From the fact that f is Clarke regular (see, e.g., [4]), a cluster point x^* of $\{x^k\}$ being Clarke stationary is equivalent to the condition $\liminf_{k\to\infty} \Psi(x^k) = 0$. Such a stationary measure is also readily computed for our benchmark problems since the Jacobian ∇F is known. For example, when the Euclidean norm on d in (6.1) is changed to an ℓ_∞ norm, $\Psi(x^k)$ can be obtained by solving the linear optimization problem

$$\underset{d,s}{\text{minimize}} \left\{ e^T s : \; s \geq F(x^k) + \nabla F(x^k)^T d, \; s \geq -F(x^k) - \nabla F(x^k)^T d, \; d \in [-1, \, 1]^n \right\}.$$

The importance of using the stationary measure Ψ is highlighted in Figure 2, where we see the performance of seven algorithms on problem 31. Most of the manifold sampling implementations find the same (and largest) amount of function-value decrease, but GYY and an SMS instance converge to a point with a relatively worse function value. Even so, these points all have similar stationary behavior. L-DFOTR and Q-DFOTR fail to find a stationary point.

6.4. Measuring Performance across the Set. For comparing the performance of algorithms across the entire test set, we use the data profiles described in [19]. Let S denote the set of solvers we wish to compare, and let P denote the set of test problems. Let $t_{p,s}$ denote the number of function evaluations required for solver $s \in S$ to satisfy a convergence criterion on problem $p \in P$. We use the convention that $t_{p,s} = \infty$ if the convergence criterion is not satisfied within the budget of evaluations.

For $\kappa \geq 0$, the data profile for solver s is then defined by

$$d_s(\kappa) = \frac{1}{|P|} \left| \{ p \in P : t_{p.s} \leq \kappa(n_p + 1) \} \right|,$$

where n_p is the dimension of problem p.

We first examine the convergence criterion in [19], which is based on the best function value found by an algorithm. In particular, given a tolerance $\tau > 0$, we will say that solver s has converged on problem p when an $x^{(j,p,s)} \in \mathbb{R}^{n_p}$ has been found such that

$$f(x^{(j,p,s)}) \leq f_p + \tau(f(x^{(0,p)}) - f_p), \tag{6.2}$$

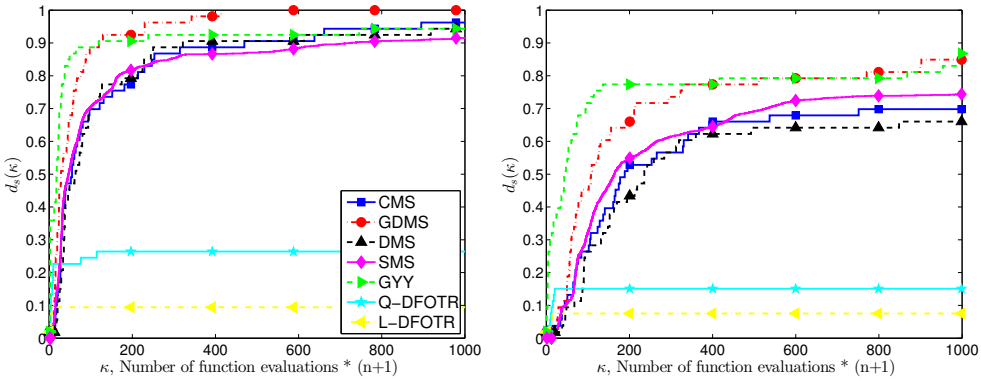

FIG. 3. *Data profiles based on the function value convergence measure* (6.2) *for* $\tau = 10^{-3}$ *(left) and* $\tau = 10^{-7}$ *(right).*

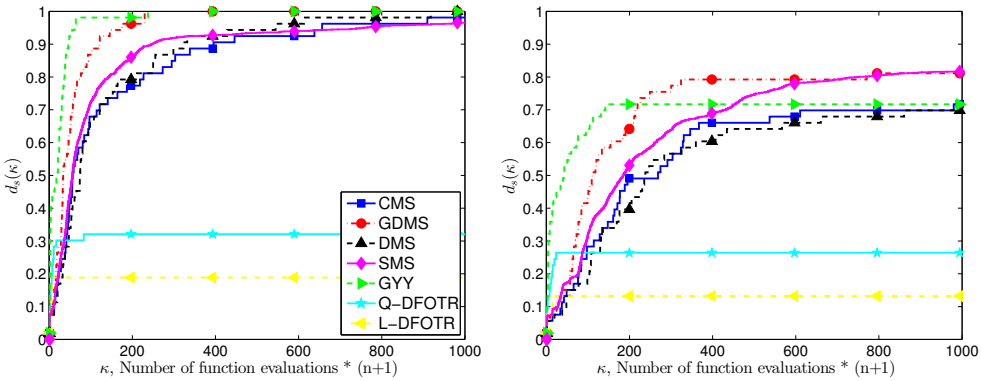

FIG. 4. *Data profiles based on the* Ψ *convergence measure* (6.3) *for* $\tau = 10^{-3}$ *(left) and* $\tau = 10^{-7}$ *(right).*

where f_p is the least function value obtained across all evaluations of all solvers in S for problem p and where $x^{(0,p)}$ is an initial point common to all solvers. The parameter τ determines how accurate one expects a solution to be in terms of the achievable decrease $f(x^{(0,p)}) - f_p$.

Figure 3 (left) shows that all manifold sampling implementations and GYY successfully find points with function values better than 99.9% of the best-found decrease for over 90% of the problems. For the smaller τ, the solvers' performances are more distinguishable. SMS finds decrease at least as good as $(1 - 10^{-7})\%$ of the best-performing method on 75% of the problems, while GDMS and GYY do so for 85% of the problems. The relative success of GDMS over the other deterministic solvers highlights the importance of judiciously using the budget of function evaluations. The quick plateau behavior for the smooth solvers L-DFOTR and Q-DFOTR indicate that these solvers are efficient on the problems that they are able to solve.

Our next convergence criterion relates to the stationarity measure Ψ defined in (6.1). Given a tolerance $\tau > 0$, we say that convergence has occurred when

$$\Psi(x^{(j,p,s)}) \leq \tau \Psi(x^{(0,p)}). \tag{6.3}$$

Using (6.3) to test for convergence in Figure 4, we gain additional insight into the

performance of the solvers. Figure 4 (left) shows that Q-DFOTR and L-DFOTR do not find points with Ψ values less than one-thousandth of the stationary measure at $x^{(0,p)}$ on a majority of the benchmark problems, while the other solvers do so for over 95% of the problems. For the more restrictive τ, CMS and DMS perform nearly identically, while SMS and GDMS are shown to be even more robust. GYY is relatively faster at finding small Ψ values in the initial $150(n+1)$ function evaluations. Note that the data profiles for SMS are improved when moving from function value measures (Figure 3) to stationarity measures (Figure 4). We attribute this behavior to the fact that some stochastic instances find stationary points with relatively worse function values (recall Figure 2).

Before proceeding, we note that although these tests show that GDMS is efficient when evaluations are performed sequentially, the other variants have the ability to utilize $n+1$ evaluations concurrently and therefore might prove more useful in a parallel setting.

6.5. Comparison with Gradient Sampling. We also compare the performance between a variant of manifold sampling that uses some gradient information (SMS-G) and GRAD-SAMP, a MATLAB implementation of gradient sampling from [3] that uses gradient information at every evaluated point. This code was run with its default settings and a budget of $1000(n+1)$ Jacobian (and hence gradient) evaluations. Since GRAD-SAMP does not proceed from a nondifferentiable initial point, the five problems with nondifferentiable starting points were perturbed by machine epsilon. We also extended the set of sampling radii in GRAD-SAMP from $\{10^{-4}, 10^{-5}, 10^{-6}\}$ to $\{10^{-4}, 10^{-5}, \ldots, 10^{-16}\}$ to avoid early termination.

As suggested by its name, SMS-G is SMS from Section 6.1 with the following modifications. The model building step, Line 4 in Algorithm 1, directly uses the Jacobian $\nabla F(x^k)$ and thus $M(x) = F(x^k) + \nabla F(x^k)(x - x^k)$. Since the default settings in GRAD-SAMP samples $\min(n+10, 2n)$ gradients per iteration, SMS-G has this manifold sampling rate as opposed to the n points sampled each iteration by SMS.

Notice that because SMS-G computes a new Jacobian only immediately following a successful iteration, it incurs at most one Jacobian evaluation per iteration. The manifold sampling step and the evaluation of the trial point in SMS-G require only function evaluations (i.e., not Jacobian evaluations), and so SMS-G incurs at most $\min(n+11, 2n+1)$ function evaluations per iteration. On the other hand, GRAD-SAMP can require a bundle of $\min(n+11, 2n+1)$ gradient (and hence Jacobian) and corresponding function evaluations per iteration. Thus, in our data profiles measured in terms of function evaluations, every function evaluation used by GRAD-SAMP entails a Jacobian evaluation; SMS-G function evaluations include a Jacobian only evaluation for a fairly small (always less than 20%) proportion of the function evaluations. That is, each function evaluation within gradient sampling includes a "free" Jacobian evaluation that is not accounted for in the presented data profiles. SMS-G uses a "free" Jacobian evaluation on fewer than 20% of the function evaluations.

Data profiles are shown in Figure 5 and Figure 6 for an experiment where 30 stochastic runs were performed for both solvers. We also compare results with 30 instances of the Jacobian-free SMS described in Section 6.1. The gradient sampling method performs significantly worse than either manifold sampling method. Furthermore, the similar performance exhibited by the Jacobian-based SMS-G and the Jacobian-free SMS indicates that the performance of SMS-G would likely further im-

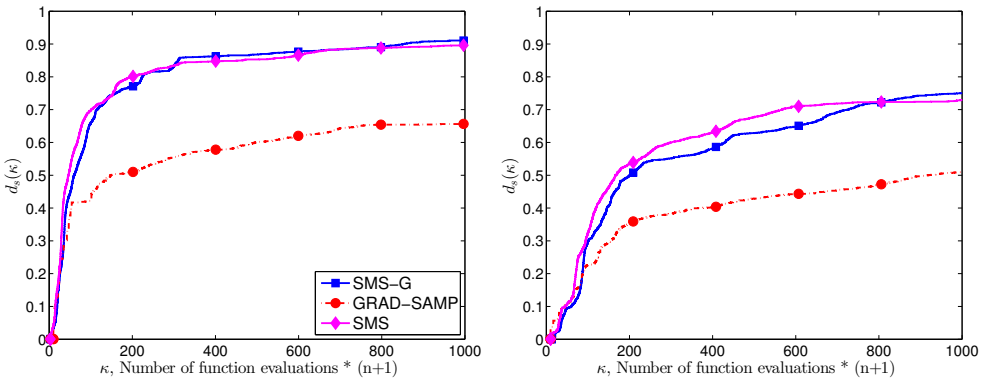

FIG. 5. *Data profiles based on the function value convergence measure* (6.2) *for* $\tau = 10^{-3}$ *(left) and* $\tau = 10^{-7}$ *(right). Note that* SMS *samples* n *points per iteration while* SMS-G *and* GRAD-SAMP *sample* $\min\{n+10, 2n\}$ *points per iteration.*

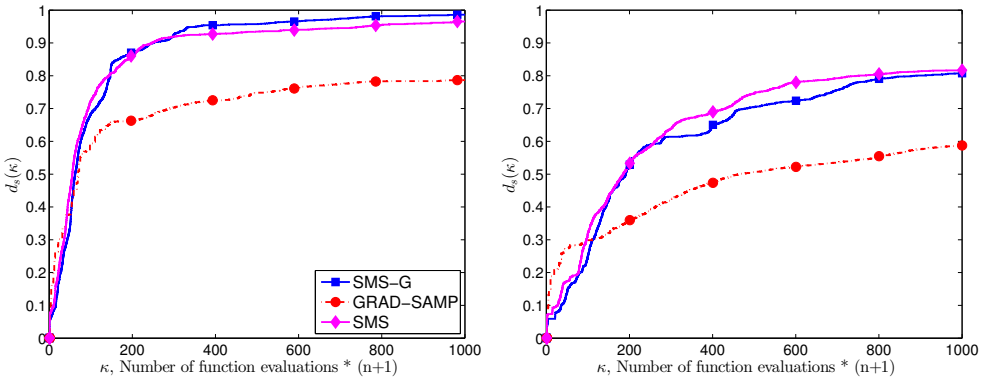

FIG. 6. *Data profiles based on the* Ψ *convergence measure* (6.3) *for* $\tau = 10^{-3}$ *(left) and* $\tau = 10^{-7}$ *(right).*

prove if the sampling rate were reduced.

7. Discussion. The driving force behind the proposed manifold sampling algorithm is that search directions are computed by using a finitely generated set,

$$\mathbf{co}\left\{\nabla M(x^k)\mathbf{sign}(y^j) + \sum_{i:[\mathbf{sign}(y^j)]_i=0} t_i \nabla m^{F_i}(x^k) \;:\; t \in \{-1,0,1\}^r, y^j \in Y(x^k;\Delta_k)\right\},$$

which differs from the finitely generated set used by gradient sampling,

$$\mathbf{co}\left\{\nabla M(y^j)\mathbf{sign}(y^j) \;:\; y^j \in \mathcal{D} \cap Y(x^k;\Delta_k)\right\}.$$

Our tests on ℓ_1 functions show that the manifold sampling strategies compare favorably with a gradient sampling approach.

Our presentation in Sections 3 and 4 focused on the case of (1.2) when $f_s = 0$. Since the presence of a nontrivial f_s does not affect the manifolds of f, such f_s can naturally be addressed by a shift of the generator set to $\nabla f_s(x^k)+\mathfrak{G}_k$, inclusion of f_s in the

master model (e.g., $\nabla m^f(x^k+s) = \left\langle s, \mathbf{proj}\Big(0, \nabla f_s(x^k) + \nabla M\left(x^k\right) \partial h\left(F\left(x^k\right)\right)\Big)\right\rangle$), and an analogous to ρ_k.

Furthermore, although the present work targets composite problems (1.2) for the particular nonsmooth function $h(u) = \|u\|_1$, the approach can be extended to other functions h, provided one can classify points into smooth manifolds (either with zeroth-order information or inexact first-order information). In the case of ℓ_1 functions, this classification was determined by the trivial evaluation of the sign pattern of $F(y)$ for a sample $y \in Y(x, \Delta)$. In the case of minimax objective functions of the form

$$h(u) = \max_{i=1,\ldots,r} u_i \qquad \text{or} \qquad \max_{i=1,\ldots,r} |u_i|,$$

this classification is determined by what Hare and Nutini refer to as the "active set" at a point [13]. In general, in any setting where the form of the subdifferential ∂h at any point is known, a setting that subsumes much of the work in the nonsmooth composite optimization literature, an analogous version of Algorithm 1 can be proposed.

In particular, the manifold sampling approach does not rely on convexity of the function h. This is in contrast to methods that solve the nonsmooth subproblem (2.6). An example of such a method is the GYY code modified from [11], which we showed can slightly outperform manifold sampling on ℓ_1 problems.

We are also interested in efficient and greedy updates of sample sets for manifolds and/or models in both settings where function (and Jacobian) evaluations are performed sequentially and concurrently. Our implementation of GDMS is a first step in this direction. Furthermore, natural questions arise about the tradeoff between the richness of manifold information required to reach early termination of the inner while loop in Algorithm 1 and the efficiency to guarantee that the sample set is, for example, well poised for model building.

Acknowledgments. This material is based upon work supported by the U.S. Department of Energy, Office of Science, under Contract DE-AC02-06CH11357. We are grateful to Kamil Khan for helpful discussions, Katya Scheinberg for the code we refer to as DFOTR, Geovani Grapiglia for the GYY implementation, and to Frank Curtis and Xiaocun Que for the use of their specialized active set method in solving our minimum-norm element QP subproblems. Wild is grateful to Aswin Kannan for early numerical experiments testing methods for composite black-box optimization. We are grateful to two referees for their comments, which lead to an improved manuscript.

REFERENCES

[1] C. Audet and J. E. Dennis, Jr., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 17 (2006), pp. 188–217.

[2] J. V. Burke, *Descent methods for composite nondifferentiable optimization problems*, Mathematical Programming, 33 (1985), pp. 260–279.

[3] J. V. Burke, A. S. Lewis, and M. L. Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM Journal on Optimization, 15 (2005), pp. 751–779.

[4] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, 1983.

[5] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, PA, 2009.

[6] F. E. Curtis and X. Que, *An adaptive gradient sampling algorithm for non-smooth optimization*, Optimization Methods and Software, 28 (2013), pp. 1302–1324.

[7] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer-Verlag New York, Inc., New York, NY, 2003.

[8] R. Fletcher, *A model algorithm for composite nondifferentiable optimization problems*, in Nondifferential and Variational Techniques in Optimization, D. C. Sorensen and R. J.-B. Wets, eds., vol. 17 of Mathematical Programming Studies, Springer Berlin Heidelberg, 1982, pp. 67–76.

[9] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, second ed., 1987.

[10] R. Garmanjani, D. Júdice, and L. N. Vicente, *Trust-region methods without using derivatives: Worst case complexity and the non-smooth case*, Preprint 15-03, Dept. Mathematics, University of Coimbra, March 2015. To appear in SIAM J. Optimization.

[11] G. N. Grapiglia, J. Yuan, and Y.-x. Yuan, *A derivative-free trust-region algorithm for composite nonsmooth optimization*, Computational and Applied Mathematics, (2014), pp. 1–25.

[12] A. Griewank, A. Walther, S. Fiege, and T. Bosse, *On Lipschitz optimization based on gray-box piecewise linearization*, Mathematical Programming, (2015), pp. 1–33.

[13] W. Hare and J. Nutini, *A derivative-free approximate gradient sampling algorithm for finite minimax problems*, Computational Optimization and Applications, 56 (2013), pp. 1–38.

[14] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, vol. 1133 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1985.

[15] ———, *A method for solving certian quadratic programming problems arising in nonsmooth optimization*, IMA Journal of Numerical Analysis, (1986), pp. 137–152.

[16] ———, *Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM Journal on Optimization, 18 (2007), pp. 379–388.

[17] ———, *A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM Journal on Optimization, 20 (2010), pp. 1983–1994.

[18] J. Larson and S. C. Billups, *Stochastic derivative-free optimization using a trust region framework*, Computational Optimization and Applications, (2016).

[19] J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM Journal on Optimization, 20 (2009), pp. 172–191.

[20] D. Popovic and A. R. Teel, *Direct search methods for nonsmooth optimization*, in Proceedings of the 43rd IEEE Conference on Decision and Control, 2004.

[21] S. M. Wild, *Derivative-free optimization algorithms for computationally expensive functions*, ph.d. thesis, Cornell University, 2009.

[22] S. M. Wild and C. A. Shoemaker, *Global convergence of radial basis function trust-region algorithms for derivative-free optimization*, SIAM Review, 55 (2013), pp. 349–371.

[23] Y.-x. Yuan, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Mathematical Programming, (1985), pp. 220–228.