ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Lemont, Illinois 60439

Manifold Sampling for Optimization of Nonconvex Functions that are Piecewise Linear Compositions of Smooth Components[1]

Kamil Khan, Jeffrey Larson, and Stefan M. Wild

Mathematics and Computer Science Division

Preprint ANL/MCS-P8001-0817

September 2017 (*revised April 2018*)

[1] This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-06CH11357.

Manifold Sampling for Optimization of Nonconvex Functions that are Piecewise Linear Compositions of Smooth Components

Kamil A. Khan[*] Jeffrey Larson[†‡] Stefan M. Wild[§]

April 19, 2018

Abstract

We develop a manifold sampling algorithm for the minimization of a nonsmooth composite function $f \triangleq \psi + h \circ F$ when ψ is smooth with known derivatives, h is a known, nonsmooth, piecewise linear function, and F is smooth but expensive to evaluate. The trust-region algorithm classifies points in the domain of h as belonging to different manifolds and uses this knowledge when computing search directions. Since h is known, classifying objective manifolds using only the values of F is simple. We prove that all cluster points of the sequence of the manifold sampling algorithm iterates are Clarke stationary; this holds although points evaluated by the algorithm are not assumed to be differentiable and when only approximate derivatives of F are available. Numerical results show that manifold sampling using zeroth-order information about F is competitive with algorithms that employ exact subgradient values from ∂f.

1 Introduction

This paper addresses the optimization problem $\underset{x\in\mathbb{R}^n}{\operatorname{minimize}} f(x)$ when f is of the form

$$f(x) \triangleq \psi(x) + h(F(x)), \tag{1}$$

where $\psi\colon \mathbb{R}^n \to \mathbb{R}$ is smooth with known derivatives, $h\colon \mathbb{R}^p \to \mathbb{R}$ is a nonsmooth, piecewise linear function with a known generalized Clarke subdifferential, and the function $F\colon \mathbb{R}^n \to \mathbb{R}^p$ is smooth. Specifically, we assume that h is analytically known and that one can identify the linear functions that contribute to the subdifferential of h at any point in its domain. This setting includes both the case when $\nabla_x F : \mathbb{R}^n \to \mathbb{R}^{n\times p}$ may not be available (such as in derivative-free optimization [7]) and the case when h is nonconvex.

In this paper, we develop a *manifold sampling* algorithm that overcomes the unavailability of $\nabla_x F$ by building a smooth model m^{F_i} of each component F_i of F. The collection of model gradients $\nabla_x m^{F_i}$ is used by the algorithm to approximate $\nabla_x F$. We show that by using such approximations, manifold sampling converges to generalized stationary points of f and performs well empirically. The algorithm proposed here extends that developed in [21] for the case where h in (1) is piecewise linear.

Previous efforts addressing functions of the form (1) have focused largely on cases where h is convex or where the derivatives of F are available. Foundational work for the case when h is convex was presented in [15, 34]. When ∇F is unavailable, [17] considers the composition of convex h with smooth F and derives error bounds between $h(F(x))$ and $h(M(x))$ (and between the subdifferentials of $h(F(x))$ and $h(M(x))$) when M is a sufficiently accurate model of F; therefore, the analysis in [17] largely generalizes the work of [18], which studies the case of $h(\cdot) \triangleq \max(\cdot)$, and [21], which studies $h(\cdot) \triangleq \|\cdot\|_1$. In [12], the authors study error bounds,

[*] Department of Chemical Engineering, McMaster University, Hamilton, ON (kamilkhan@mcmaster.ca).

[†] Corresponding author

[‡] Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL (jmlarson@anl.gov, wild@anl.gov).

convergence, and termination of algorithms that use Taylor-like models of (1) when ∇F is available and h is convex. Similarly, the authors of [16] address the worst-case complexity of trust-region algorithms when h is convex and a smoothing function of h exists and is known. Recent work on nonconvex h when ∇F and $\nabla \psi$ are assumed available includes analysis of prox-regular methods in [13] and analysis of quasi-Newton methods in [31]. Additional methods for optimizing nonsmooth, nonconvex objectives when gradient information is unavailable include [20], which proposes a version of gradient sampling that approximates gradients using function evaluations, and [2], which employs approximate subgradients to define descent directions.

Accompanying the developments in machine learning is a growing interest in *nonconvex* loss functions (see, e.g., [1, 4, 6, 23]). Such nonconvex loss functions have been observed to be more efficient in modeling of complex machine learning problems [3] and their use can result in other computational benefits [6]. Examples of such nonconvex h include those in [24, Appendix B] and the censored ℓ_1-loss in [33]. We use the censored ℓ_1-loss (which is piecewise linear) in the numerical experiments in Section 5 to compare manifold sampling with other algorithms for optimizing nonsmooth, nonconvex functions. In our experiments, two of the three manifold sampling implementations do not use exact gradient information, whereas other algorithms are given elements of the subdifferential of f.

An outline of the paper, which subsumes the results in [21], is as follows. Section 2 contains background information and definitions used throughout the paper and formally defines piecewise linear functions. In considering such functions, we present a significant extension of manifold sampling compared with the version developed in [21] addressing ℓ_1 functions. By couching the development of manifold sampling in terms of piecewise linear functions, we hope that extensions to other piecewise-continuous cases may more easily arise. Section 3 presents the new manifold sampling algorithm, which includes a more general trust-region acceptance test that applies to other functions as well as ℓ_1 objective functions. Section 4 analyzes the sequence of iterates generated by manifold sampling. Many of the results in Section 4 are considerable extensions over their counterpart results in [21]. For example, the proof of [21, Lemma 4] explicitly uses the fact that h is an ℓ_1 function, whereas here we address general piecewise linear functions. Section 5 compares three implementations of manifold sampling with implementations of other methods.

2 Background

Before proceeding, we present the definitions, notation, and assumptions used in this paper. All norms are assumed to be ℓ_2 norms unless otherwise stated. The *closure*, *interior*, and *convex hull* of a set $\mathcal{S}$ are denoted $\mathbf{cl}\,(\mathcal{S})$, $\mathbf{int}\,(\mathcal{S})$, and $\mathbf{co}\,(\mathcal{S})$, respectively. Given a closed convex set $\mathcal{S} \subset \mathbb{R}^n$, the *projection* of $x \in \mathbb{R}^n$ onto $\mathcal{S}$, denoted $\mathbf{proj}(x, \mathcal{S})$, is the unique element of the set $\operatorname*{arg\,min}_{y \in \mathcal{S}} \|y - x\|$. Let $\mathcal{B}(x; \Delta) \triangleq \{y : \|x - y\| \leq \Delta\}$.

2.1 Fully linear models

Manifold sampling constructs models m^{F_i} of each component F_i; in order to show convergence of our algorithm, m^{F_i} must sufficiently approximate F_i near a given point x. This property is formalized in the following standard definition [7].

Definition 1. *A function $m^{F_i} \colon \mathbb{R}^n \to \mathbb{R}$ is said to be a* fully linear *model of F_i on $\mathcal{B}(x; \Delta)$ if there exist constants $\kappa_{i,\mathrm{ef}}$ and $\kappa_{i,\mathrm{eg}}$, independent of x and Δ, so that*

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

For derivation of error bounds for f when fully linear models of each component function F_i are available and h is convex, see [17].

One can easily build fully linear component models of smooth functions F_i (including functions that satisfy our later assumption in Assumption 2), even when ∇F_i is unavailable; see, for example, [7, Chapter 6]. Since an evaluation of F provides values for all components, using a common set of points to build all component models can save significant resources when F is expensive to evaluate. Although we do not need to ensure that every component model used within our algorithm is fully linear, we make the following assumption for ease of presentation.

Assumption 1. *Each model m^{F_i} in $M \triangleq [m^{F_1}, \dots, m^{F_p}]^T$ is a fully linear model of F_i and twice continuously differentiable. Also, for $i \in \{1, \dots, p\}$ there exists $\kappa_{i,\mathrm{mH}}$ so that $\|\nabla^2 m^{F_i}(x)\| \leq \kappa_{i,\mathrm{mH}}$ for all $x \in \mathbb{R}^n$. For these constants and those in* Definition 1*, define $\kappa_{\mathrm{f}} \triangleq \sum_{i=1}^p \kappa_{i,\mathrm{ef}}$, $\kappa_{\mathrm{g}} \triangleq \sum_{i=1}^p \kappa_{i,\mathrm{eg}}$, and $\kappa_{\mathrm{mH}} \triangleq \sum_{i=1}^p \kappa_{i,\mathrm{mH}}$.*

Assumption 1 implicitly assumes that each component function is continuously differentiable; formal assumptions about the component functions are stated in Assumption 2.

2.2 Generalized derivatives

We now introduce terminology from nonsmooth analysis.

Definition 2 (from [29, 5])**.** *The* B-subdifferential *of a locally Lipschitz continuous function f at a point x is the set of all limiting gradients from differentiable points that converge to x, that is,*

$$\partial_{\mathrm{B}} f(x) \triangleq \left\{ \lim_{y^j \to x} \nabla f(y^j) : \ y^j \in \mathcal{D} \right\}, \tag{2}$$

where $\mathcal{D}$ is the set of points where f is differentiable. The generalized Clarke subdifferential *of f at x is defined as*

$$\partial_{\mathrm{C}} f(x) \triangleq \mathbf{co}\left(\partial_{\mathrm{B}} f(x)\right).$$

Definition 3. *A point x is called a* Clarke stationary *point of $f \colon X \to \mathbb{R}$ if $0 \in \partial_{\mathrm{C}} f(x)$.*

If f is scalar-valued and locally Lipschitz continuous on an open set $X \subset \mathbb{R}^n$, then Clarke stationarity is a necessary condition for local optimality.

2.3 Piecewise linear functions

Throughout this paper, affine functions are referred to as "linear"; thus, linear functions are not required to vanish at the origin.

Definition 4 (adapted from [30])**.** *A function $h \colon \mathbb{R}^p \to \mathbb{R}^q$ is* piecewise linear *if h is continuous and there exists a finite collection $\mathfrak{H} \triangleq \{h_i : i = 1, \dots, \bar{m}\}$ of affine functions that map $\mathbb{R}^p$ into $\mathbb{R}^q$, for which*

$$h(z) \in \left\{ \tilde{h}(z) : \tilde{h} \in \mathfrak{H} \right\}, \quad \forall z \in \mathbb{R}^p.$$

In this case, h is said to be a continuous selection *of $\mathfrak{H}$, and the elements of $\mathfrak{H}$ are called* selection functions *of h.*

All functions satisfying Definition 4 are Lipschitz continuous everywhere and B-differentiable everywhere. That is, $\partial_{\mathrm{B}} f$ and therefore $\partial_{\mathrm{C}} f$ (from Definition 2) are well defined everywhere [30, 5]. Although h is piecewise linear by assumption, $h \circ F$ might not be.

We now define useful sets for describing piecewise linear functions.

Definition 5 (adapted from [30])**.** *We employ the following sets for functions h satisfying* Definition 4*:*

$$\mathcal{S}_i \triangleq \{y : h(y) = h_i(y)\}, \quad \tilde{\mathcal{S}}_i \triangleq \mathbf{cl}\left(\mathbf{int}\left(\mathcal{S}_i\right)\right), \quad I_h^e(z) \triangleq \left\{ i : z \in \tilde{\mathcal{S}}_i \right\}.$$

Elements of $I_h^e(z)$ are called essentially active indices*; any function h_i for which $i \in I_h^e(z)$ is an* essentially active selection function *for h at z. A function $h_j \in \mathfrak{H}$ is an* essentially active selection function *for h, or a* linear piece *of h (without reference to a particular point z) if it is essentially active for h at some point z.*

For any $z \in \mathbb{R}^p$, we define its *manifold* to be

$$\mathcal{M}(z) \triangleq \{y \in \mathbb{R}^p : I_h^e(y) = I_h^e(z)\}.$$

Since h_i is assumed to be linear, each set $\tilde{\mathcal{S}}_i$ is a union of finitely many convex polyhedra. Thus, each manifold $\mathcal{M}(z)$ is also a convex polyhedron. Note that the manifolds do not necessarily partition the domain and that Definition 4 does not specify the collection $\mathfrak{H}$ uniquely (though the set of functions that are essentially active somewhere is uniquely defined for piecewise linear functions). If $h \triangleq \|\cdot\|_\infty$, then $h \circ F$ is a continuous selection of the $2p$ functions in $\{\pm F_1, \dots, \pm F_p\}$. If $h \triangleq \|\cdot\|_1$, then $h \circ F$ is a continuous selection of the 2^p functions in $\{\sum_{i=1}^p s_i F_i : s_i \in \{-1, 1\}\}$.

To show that cluster points of the sequence of iterates from the manifold sampling algorithm are Clarke stationary, we make the following assumptions on ψ, h, and F.

Assumption 2. *Suppose that the set $\mathcal{L} \triangleq \left\{x : f(x) \le f(x^0)\right\}$ is bounded and the function f is of the form* (1) *where h is a piecewise linear* (Definition 4) *selection of $\mathfrak{H} \triangleq \{h_1, \dots, h_{\bar{m}}\}$, where $h_i : z \in \mathbb{R}^p \mapsto \langle a_i, z\rangle + b_i$ for each i. Define $L_h \triangleq \max\{\|a_i\| : i = 1, \dots, \bar{m}\}$; observe that L_h is a Lipschitz constant for h. Suppose that the essentially active index set $I_h^e(z)$* (Definition 5) *can be computed for each $z \in \mathbb{R}^p$.*

For a constant $\Delta_{\max} > 0$ define $\mathcal{L}_{\max} = \bigcup_{x \in \mathcal{L}} \mathcal{B}(x; \Delta_{\max})$. Suppose that each F_i is continuously differentiable on $\mathcal{L}_{\max}$ and that ∇F is Lipschitz continuous on $\mathcal{L}_{\max}$ with a Lipschitz constant $L_{\nabla F}$. Similarly, suppose that ψ is twice continuously differentiable on $\mathcal{L}_{\max}$ and that $\nabla\psi$ is Lipschitz continuous on $\mathcal{L}_{\max}$ with a Lipschitz constant $L_{\nabla\psi}$.

Define $\kappa_{\mathrm{fH}} \triangleq \max_{x \in \mathcal{L}_{\max}} \left\{\left\|\nabla^2\psi(x)\right\|\right\} + L_h \kappa_{\mathrm{mH}}$, and observe that κ_{fH} is finite.

3 Algorithmic Framework

This section provides a rough outline of the manifold sampling algorithm (presented in Algorithm 1) for optimizing a function f of the form (1) subject to Assumptions 1 and 2. Our algorithm builds component models m^{F_i} of each F_i at a point x and places the first-order terms of each model in the ith column of the matrix $\nabla M(x) \in \mathbb{R}^{n \times p}$. That is,

$$\nabla M(x) \triangleq \left[\nabla m^{F_1}(x), \dots, \nabla m^{F_p}(x)\right].$$

At the kth iteration of the algorithm, the current iterate x^k is known. The p component function values $F_i(x^k)$ can be computed, and the set of essentially active indices $I_h^e(F(x^k))$ is available. Then, p component models m^{F_i} that approximate F_i near x^k are built. Using the elements of $I_h^e(F(x^k))$, we infer a set of generators $\mathfrak{G}^k$ using the manifolds that are potentially active at x^k. Elements of $\mathfrak{G}^k$ will be of the form $\nabla\psi(x^k) + \nabla M(x^k)\, a_i$ for suitable manifolds $\langle a_i, x\rangle + b_i$ and $\mathbf{co}\left(\mathfrak{G}^k\right)$ can then be used as an approximation to $\partial_{\mathrm{C}} f(x^k)$.

3.1 Master model

If $\mathfrak{G}^k$ contains t elements, Algorithm 1 uses these t generators to infer that the corresponding gradients $\{a_{j_1}, \dots, a_{j_t}\}$ of selection functions of h may be active at (or relatively near) the current iterate x^k. The minimum-norm element of $\mathfrak{G}^k$ is denoted

$$g^k \triangleq \mathbf{proj}\left(0, \mathbf{co}\left(\mathfrak{G}^k\right)\right) \in \mathbf{co}\left(\mathfrak{G}^k\right). \tag{3}$$

We let $\lambda^* \in [0,1]^t$ with $\sum_i \lambda_i^* = 1$ denote the coefficients of the convex combination $g^k = G^k\lambda^*$, where the columns of G^k are the generators in $\mathfrak{G}^k$. The coefficients λ^* may be obtained by solving the quadratic optimization problem

$$\begin{array}{ll} \underset{\lambda}{\text{minimize}} & \dfrac{1}{2}\lambda^T (G^k)^T G^k \lambda \\ \text{subject to} & e^T\lambda = 1,\ \lambda \ge 0. \end{array} \tag{4}$$

These coefficients λ^* will be used to combine the p component models m^{F_i} into a smooth model of f.

Since the qth generator in $\mathfrak{G}^k$ (alternatively, the qth column of G^k) is given by $\nabla\psi(x^k) + \nabla M(x^k)a_{j_q}$, we have $G^k = \nabla\psi(x^k)\, e^T + \nabla M(x^k)A^k$, where

$$A^k \triangleq \begin{bmatrix} | & & | \\ a_{j_1} & \cdots & a_{j_t} \\ | & & | \end{bmatrix}.$$

To ensure that the smooth *master model* $m_k^f \colon \mathbb{R}^n \to \mathbb{R}$ has a gradient equal to g^k from (3), we consider the set of weights $w^k = A^k\lambda^*$ and define

$$m_k^f(x) \triangleq \psi(x^k) + \sum_{i=1}^p w_i^k m^{F_i}(x) + \sum_{i=1}^p \lambda_i^* b_{j_i}. \tag{5}$$

Note that the last term in (5) is constant and does not affect the model's minimizer. It is included to ensure a direct correspondence between m^{F_i} and F_i.

Observe that by construction,

$$\begin{aligned} \nabla m_k^f(x^k) &= \nabla\psi(x^k) + \sum_{i=1}^p w_i^k \nabla m^{F_i}(x^k) \\ &= \nabla\psi(x^k) + \sum_{i=1}^p \nabla m^{F_i}(x^k)(A^k\lambda^*)_i \\ &= \nabla\psi(x^k)(e^T\lambda^*) + \nabla M(x^k)A^k\lambda^* = G^k\lambda^* = g^k. \end{aligned}$$

Note that $w_i^k \in [-L_h, L_h]$ for each $i \in \{1,\dots,t\}$ due to the fact that $\lambda^* \in [0,1]^t$ and $\sum_i \lambda_i^* = 1$. If G^k contains exactly one generator (i.e., $t = 1$), then $\lambda^* = 1$, and the master model is simply

$$m_k^f(x) = \psi(x) + \sum_{i=1}^p (a_{j_1})_i m^{F_i}(x) + b_{j_1} = \psi(x) + \langle M(x), a_{j_1}\rangle + b_{j_1}.$$

3.2 Sufficient decrease condition

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

$$\text{minimize}\left\{ m_k^f(x^k + s) : s \in \mathcal{B}(0;\Delta_k) \right\}. \tag{6}$$

As with traditional trust region methods, this problem does not have to be solved exactly. Rather, the solution s^k of (6) needs only to satisfy the sufficient decrease condition

$$\psi(x^k) - \psi(x^k + s^k) + \left\langle M(x^k) - M(x^k + s^k), a^{(k)} \right\rangle \geq \frac{\kappa_{\mathrm{d}}}{2}\|g^k\| \min\left\{ \Delta_k, \frac{\|g^k\|}{\kappa_{\mathrm{mH}}} \right\}, \tag{7}$$

where $a^{(k)}$ is the gradient of some selection function $h^{(k)} \in \mathfrak{H}$ satisfying

$$h^{(k)}(F(x^k)) \leq h(F(x^k)) \qquad \text{and} \qquad h^{(k)}(F(x^k + s^k)) \geq h(F(x^k + s^k)). \tag{8}$$

Note that the sufficient decrease condition (7) differs from the typical trust-region method; instead of measuring the decrease in m_k^f between x^k and $x^k + s^k$, (7) measures the decrease using a selection function $h^{(k)}$. The sufficient decrease condition (7) extends the approach from [21], where $h = \|\cdot\|_1$ and decrease is measured by using the sign pattern of F at $x^k + s^k$. Lemma 2 will show that an $h^{(k)} \in \mathfrak{H}$ satisfying (8) exists when h is piecewise linear. Lemma 3 will show that an analogue of [21, Lemma 1] (replacing $\mathbf{pat}_q$ with $a^{(k)}$) guarantees the existence of a relatively easy-to-find Cauchy point that satisfies (7).

If several selection functions $h^{(k)} \in \mathfrak{H}$ satisfy (8), then any may be chosen. Where possible, our experience to date suggests choosing a selection function that maximizes the descent in f from x^k to $x^k + s^k$ while satisfying (8). Since the function ψ does not affect the selection functions, maximizing descent amounts to choosing

$$h^{(k)} \in \operatorname*{arg\,max}_{h_i \in \mathfrak{H}} \{h_i(F(x^k)) - h_i(F(x^k + s^k)) : h_i \text{ satisfies (8)}\}. \tag{9}$$

3.3 ρ_k test

In common with other trust-region methods, manifold sampling uses a ρ_k test to measure whether the master model m_k^f sufficiently approximates the function f within the trust region. Manifold sampling measures this agreement using a specific element $h^{(k)}$ of the set $\mathfrak{H}$ that defines h instead of using h itself. The value of ρ_k can therefore be considered the ratio of actual decrease to predicted decrease in f using a selection function $h^{(k)}$. Before ρ_k can be calculated, manifold information from $h^{(k)}$ must be included in $\mathfrak{G}^k$. Therefore, $\mathfrak{G}^k$ may be augmented after a putative step s^k has been computed and $F(x^k + s^k)$ has been evaluated. Although adding manifold information to $\mathfrak{G}^k$ may result in a given iteration having more than one trust-region subproblem and therefore more than one evaluation of F per iteration, in practice the number of function evaluations per iteration is rarely more than 1. This process of adding elements to $\mathfrak{G}^k$ will not cycle indefinitely because the number of manifolds defining h is finite.

Explicitly, given a selection function $h^{(k)} \in \mathfrak{H}$ (with gradient $a^{(k)}$) satisfying (8), ρ_k is the ratio

$$\rho_k \triangleq \frac{\psi(x^k) - \psi(x^k + s^k) + h^{(k)}(F(x^k)) - h^{(k)}(F(x^k + s^k))}{\psi(x^k) - \psi(x^k + s^k) + \langle M(x^k) - M(x^k + s^k), a^{(k)} \rangle}. \tag{10}$$

The point $x^k + s^k$ is chosen to be the next iterate only if ρ_k is sufficiently large.

3.4 Generator set $\mathfrak{G}^k$

We complete our discussion of the manifold sampling algorithm by showing how, in the kth iteration of Algorithm 1, the set $\mathfrak{G}^k$ of generators is built. We ultimately show that the generated $\mathbf{co}\left(\mathfrak{G}^k\right)$ approximates $\partial_{\mathrm{C}} f(x^k)$ sufficiently well in order to ultimately guarantee convergence of our algorithm.

Several approaches for constructing $\mathfrak{G}^k$ are possible; we impose the following requirement for $\mathfrak{G}^k$.

Assumption 3. *At iteration k of* Algorithm 1*, the constructed set $\mathfrak{G}^k$ satisfies*

$$\begin{gathered}\left\{\nabla\psi(x^k) + \nabla M(x^k)\, a_i : i \in I_h^e(F(x^k))\right\} \subseteq \mathfrak{G}^k \textit{ and} \\ \mathfrak{G}^k \subseteq \left\{\nabla\psi(x^k) + \nabla M(x^k)\, a_i : y \in \mathcal{B}\left(x^k; \Delta_k\right), i \in I_h^e(F(y))\right\}.\end{gathered}$$

In practice, the set $\bigcup_{y \in \mathcal{B}(x^k;\Delta_k)} I_h^e(F(y))$ in Assumption 3 may be difficult to evaluate, since, in the derivative-free case, F is available only through sampling. Instead, the following two choices for $\mathfrak{G}^k$ are consistent with this assumption and may be constructed in practice:

- $\left\{\nabla\psi(x^k) + \nabla M(x^k)\, a_i : i \in I_h^e(F(x^k))\right\}$ and
- $\left\{\nabla\psi(x^k) + \nabla M(x^k)\, a_i : i \in I_h^e(F(y)),\ y \in Y\right\}$, for some finite set $Y \subset \mathcal{B}(x^k; \Delta_k)$.

We suggest constructing the set Y using points where f has been evaluated in previous iterations. These points could be any point evaluated before iteration k (including, for example, points evaluated while constructing component models m^{F_i}). In the numerical results in Section 5, we compare both of the above approaches for building generator sets.

Observe that the second of these approaches uses manifold information from points near x^k. This is similar to the approach of gradient sampling [19] but has two key differences: we do not assume that h is differentiable at any of the sampled points, and we are not approximating the gradient at any point other

Algorithm 1: Manifold sampling for piecewise linear compositions

```
1  Set $\eta_1 \in (0,1)$, $\kappa_{\rm d} \in (0,1)$, $\kappa_{\rm mH} \geq 0$, $\frac{1}{\eta_2} \in (\kappa_{\rm mH}, \infty)$, $\gamma_{\rm dec} \in (0,1)$, $\gamma_{\rm inc} \geq 1$, and $\Delta_{\max} > 0$
2  Choose initial iterate $x^0$ and trust-region radius $\Delta_0$ satisfying $\Delta_{\max} \geq \Delta_0 > 0$
3  for $k = 0, 1, 2, \ldots$ do
4      Build $p$ component models $m_k^{F_i}$ that are fully linear and satisfy $\sum_{i=1}^{p} \left\| \nabla^2 m_k^{F_i} \right\| \leq \kappa_{\rm mH}$ on $\mathcal{B}(x^k; \Delta_k)$
5      Form $\nabla M(x^k)$ using $\nabla m_k^{F_i}(x^k)$
6      Construct $\mathfrak{G}^k \subset \mathbb{R}^n$ satisfying Assumption 3
7      $\rho_k \leftarrow -\infty$
8      while $\rho_k = -\infty$ do
9          Build master model $m_k^f$ using (5)
10         if $\Delta_k < \eta_2 \|\nabla m_k^f(x^k)\|$ (acceptability criterion) then
11             Approximately solve (6) to obtain $s^k$
12             Evaluate $F(x^k + s^k)$ and set $h^{(k)}$ satisfying (8)
13             if $(\nabla \psi(x^k) + \nabla M(x^k)\, a^{(k)}) \in \mathfrak{G}^k$ then
14                 if $s^k$ does not satisfy (7) then
15                     Approximately solve (6) to obtain a new $s^k$ satisfying (7)
16                     Evaluate $F(x^k + s^k)$ and set $h^{(k)}$ satisfying (8)
17                 Calculate $\rho_k$ using (10)
18             else
19                 $\mathfrak{G}^k \leftarrow \mathfrak{G}^k \cup \{\nabla \psi(x^k) + \nabla M(x^k)\, a^{(k)}\}$ and update $m_k^{F_i}$
20         else
21             break out of while loop; iteration is unacceptable
22     if $\rho_k > \eta_1 > 0$ (successful iteration) then
23         $x^{k+1} \leftarrow x^k + s^k$, $\Delta_{k+1} \leftarrow \min\{\gamma_{\rm inc}\Delta_k, \Delta_{\max}\}$
24     else
25         $x^{k+1} \leftarrow x^k$, $\Delta_{k+1} \leftarrow \gamma_{\rm dec}\Delta_k$
```

than x^k where we approximate the minimum-norm element of $\partial_{\rm C} f(x^k)$. 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.

We conclude this section by giving our algorithmic framework and restrictions on algorithmic parameters in Algorithm 1, which employs various intermediate constructions described in this section. Note the following aspects of the algorithm.

Line 8: Algorithm 1 will stay in this while loop fewer than $|\mathfrak{H}| - \left| I_h^e \left(F(x^k) \right) \right|$ times, where I_h^e is defined in Definition 5. The reason is that

$$\left\{ \nabla \psi(x^k) + \nabla M(x^k) a_i : a_i \in \mathfrak{H} \right\} \supseteq \mathfrak{G}^k \supseteq \left\{ \nabla \psi(x^k) + \nabla M(x^k) a_i : i \in I_h^e \left(F(x^k) \right) \right\}$$

and each time Line 13 is visited, the cardinality of $\mathfrak{G}^k$ has increased by one. In the worst case, information from all elements of $\mathfrak{H}$ must be added to $\mathfrak{G}^k$ before ρ_k is calculated.

Line 12: The existence of at least one such $h^{(k)}$ is guaranteed by Lemma 2. Furthermore, Lemma 2 even guarantees that there exists a linear component that is essentially active somewhere in $\textbf{co}\left(\left\{ F(x^k), F(x^k + s^k) \right\} \right)$; our analysis does not require that the selected $h^{(k)}$ is active in $\textbf{co}\left(\left\{ F(x^k), F(x^k + s^k) \right\} \right)$.

Line 15: That such an $x^k + s^k$ can be computed in a straightforward manner is shown in Lemma 3.

Acceptable iterations: As defined in Line 10, acceptable iterations occur when $\Delta_k < \eta_2 \|\nabla m_k^f(x^k)\| = \eta_2\|g^k\|$, and so the norm of the master model gradient is sufficiently large to consider taking a step s^k. Note that on these iterations

$$\|g^k\| \geq \min\left\{\kappa_{\mathrm{mH}}\Delta_k, \|g^k\|\right\} \geq \kappa_{\mathrm{mH}} \min\left\{\Delta_k, \eta_2 \left\|g^k\right\|\right\} = \kappa_{\mathrm{mH}}\Delta_k. \tag{11}$$

Successful iterations: As defined in Line 22, successful iterations are those acceptable iterations for which $\rho_k > \eta_1$ and $x^{k+1} \leftarrow x^k + s^k$. Note that on every successful iteration,

- $(\nabla\psi(x^k) + \nabla M(x^k)a^{(k)}) \in \mathfrak{G}^k$ and
- the decrease condition in (7) is satisfied.

Line 23: By construction, all points evaluated by the algorithm are in the set $\mathcal{L}_{\max}$ defined in Assumption 2.

4 Analysis of Manifold Sampling

To study Algorithm 1, we first show some preliminary results in Section 4.1 and then analyze the algorithm's sequence of iterates in Section 4.2.

4.1 Preliminaries

We now show a result linking elements in $\mathbf{co}\left(\mathfrak{G}^k\right)$ 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 Lemma 1.

Lemma 1. *Let* Assumptions 1 *and* 2 *hold, and let $x, y \in \mathbb{R}^n$ satisfy $\|x - y\| \leq \Delta$. Choose any subsets $I, J \subseteq \{1, \ldots, |\mathfrak{H}|\}$ for which $I \subseteq J$, and define*

$$\mathfrak{G} \triangleq \{\nabla\psi(x) + \nabla M(x)\, a_i : i \in I\} \text{ and } \mathcal{H} \triangleq \mathbf{co}\left\{\nabla\psi(y) + \nabla F(y)\, a_j : j \in J\right\}.$$

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

$$\|g - v(g)\| \leq c_2\Delta, \tag{12}$$

where c_2 is defined by

$$c_2 \triangleq L_{\nabla\psi} + L_h(L_{\nabla F} + \kappa_{\mathrm{g}}), \tag{13}$$

for $L_{\nabla\psi}$ L_h, and $L_{\nabla F}$ defined in Assumption 2 *and κ_{g} defined in* Assumption 1.

Proof (adapted from [21]). Any $g \in \mathbf{co}(\mathfrak{G})$ may be expressed as

$$g = \nabla\psi(x) + \sum_{i \in I} \lambda_i \nabla M(x)\, a_i, \tag{14}$$

where $\sum_{i\in I} \lambda_i = 1$ and $\lambda_i \geq 0$ for each i.

By supposition, $(\nabla\psi(y) + \nabla F(y)\, a_i) \in \mathcal{H}$ for all $i \in I$. For

$$v(g) \triangleq \nabla\psi(y) + \sum_{i\in I} \lambda_i \nabla F(y)\, a_i,$$

using the same λ_i as in (14) for $i \in I$, convexity of $\mathcal{H}$ implies that $v(g) \in \mathcal{H}$. Since $y \in \mathcal{B}(x; \Delta)$, the triangle inequality and Assumptions 1 and 2 give

$$\begin{aligned}\|\nabla M(x)\, a_i - \nabla F(y)\, a_i\| \leq\ & \|\nabla F(y) - \nabla F(x)\| \|a_i\| + \|\nabla F(x) - \nabla M(x)\| \|a_i\| \\ \leq\ & (L_h L_{\nabla F} + \kappa_{\mathrm{g}} L_h)\Delta,\end{aligned}$$

for each i. Using this along with (14) and the definition of $v(g)$ yields

$$\begin{aligned}\|g - v(g)\| &\leq \left\| \nabla\psi(x) - \nabla\psi(y) + \sum_{i\in I} [\lambda_i \nabla M(x)\, a_i - \lambda_i \nabla F(y)\, a_i] \right\| \\ &\leq \|\nabla\psi(x) - \nabla\psi(y)\| + \sum_{i\in I} \lambda_i \|\nabla M(x)\, a_i - \nabla F(y)\, a_i\| \leq c_2\Delta.\end{aligned}$$

□

The approximation property in Lemma 1 can be used to motivate the use of the master model gradient in (3).

Before proceeding, we prove the following lemma, which shows that there always exists a selection function $h^{(k)}$ satisfying (8).

Lemma 2. *Consider a piecewise linear function $\phi\colon [\ell, u] \subset \mathbb{R} \to \mathbb{R}$. There exists a linear function $\bar{\phi}\colon \mathbb{R} \to \mathbb{R}$ satisfying all the following conditions:*

- *$\bar{\phi}$ is an essentially active selection function for ϕ at some $x \in [\ell, u]$,*
- *$\bar{\phi}(\ell) \leq \phi(\ell)$, and*
- *$\bar{\phi}(u) \geq \phi(u)$.*

In particular, if Assumption 2 holds, then for any $x^k, s^k \in \mathbb{R}^n$, there exist $y \in \mathbf{co}\left(\{F(x^k), F(x^k + s^k)\}\right)$ and a selection function $h^{(k)} \in I_h^e(y) \subset \mathfrak{H}$ satisfying (8).

Proof. The case when $l = u$ is trivial. Therefore, let us assume that $\ell < u$. For some $p \in \mathbb{N}$, there exist $\ell \triangleq t_0 < t_1 < \ldots < t_p \triangleq u$ for which ϕ is linear on $[t_{k-1}, t_k]$ for each $k \in \{1, \ldots, p\}$. Choose $a_k, b_k \in \mathbb{R}$ for which $\phi(x) = a_k x + b_k \triangleq \phi_k(x)$ for each $x \in [t_{k-1}, t_k]$ and $k \in \{1, \ldots, p\}$. The lemma will be proved by induction on p.

As the base case of the inductive argument, if $p \triangleq 1$, then $\phi \equiv \phi_1$ on $[\ell, u]$, and $\bar{\phi} \triangleq \phi_1$ satisfies each condition trivially.

As the inductive step, suppose that the lemma's claims have been established when $p \triangleq q$. Now consider the case in which $p \triangleq q + 1$. By construction, $\phi_1(\ell) = \phi(\ell)$. If $a_1 \geq \frac{\phi(u)-\phi(\ell)}{u-\ell}$, then $\phi_1(u) \geq \phi(u)$, and so ϕ_1 is the required selection function $\bar{\phi}$. Next, suppose that $a_1 < \frac{\phi(u)-\phi(\ell)}{u-\ell}$. The inductive assumption applies to ϕ on the subdomain $[t_1, u]$; thus, there exists $\kappa \in \{2, \ldots, q+1\}$ for which $\phi_\kappa(u) \geq \phi(u)$ and $\phi_\kappa(t_1) \leq \phi(t_1)$. It suffices to show that $\phi_\kappa(\ell) \leq \phi(\ell)$; to obtain a contradiction, suppose that $\phi_\kappa(\ell) > \phi(\ell)$, in which case

$$a_\kappa = \frac{\phi_\kappa(t_1) - \phi_\kappa(\ell)}{t_1 - \ell} < \frac{\phi(t_1) - \phi(\ell)}{t_1 - \ell} = a_1.$$

Consequently,

$$\begin{aligned}\phi(u) - \phi(\ell) &\leq (\phi_\kappa(u) - \phi_\kappa(t_1)) + (\phi_1(t_1) - \phi_1(\ell)) \\ &= a_\kappa(u - t_1) + a_1(t_1 - \ell) < a_1(u - \ell) < \phi(u) - \phi(\ell),\end{aligned}$$

which is a contradiction. Therefore, the claimed conditions are all satisfied when $\bar{\phi} \triangleq \phi_\kappa$.

Now, suppose that Assumption 2 holds, and choose any fixed $x^k, s^k \in \mathbb{R}^n$. The obtained result may be applied to the piecewise linear mapping:

$$\phi : [0, 1] \to \mathbb{R} : t \mapsto (1-t)\psi(x^k) + t\psi(x^k + s^k) + h\left((1-t)F(x^k) + tF(x^k + s^k)\right),$$

whose essentially active selection functions at any $\tilde{t} \in [0, 1]$ may all be chosen to take the form

$$\tilde{\phi} : [0, 1] \to \mathbb{R} : t \mapsto (1-t)\psi(x^k) + t\psi(x^k + s^k) + \tilde{h}\left((1-t)F(x^k) + tF(x^k + s^k)\right),$$

for $\tilde{h} \in I_h^e(\tilde{y})$, where $\tilde{y} := (1 - \tilde{t})F(x^k) + \tilde{t}F(x^k + s^k)$. The final claimed result follows immediately. □

We now use Lemma 2 to show that certain regularity assumptions guarantee that Line 15 in Algorithm 1 is satisfiable. We note that Line 15 is not reached if $0 \in \mathfrak{G}^k$ by virtue of the acceptability criterion.

Lemma 3. *For any a_q satisfying $0 \neq (\nabla\psi(x^k) + \nabla M(x^k)a_q) \in \mathfrak{G}^k$ upon reaching* Line 15 *of* Algorithm 1*, if M and f satisfy* Assumptions 1 *and* 2*, $\kappa_\mathrm{d} \in (0,1)$, κ_fH is as defined in* Assumption 2*, and*

$$j^* \triangleq \max\left\{0, \left\lceil \log_{\kappa_\mathrm{d}}\left(\frac{\|\nabla\psi(x^k) + \nabla M(x^k)a_q\|}{\kappa_\mathrm{fH}\Delta_k}\right)\right\rceil\right\}, \tag{15}$$

then

$$\hat{s}^{j^*} \triangleq -\kappa_\mathrm{d}^{j^*}\Delta_k \frac{\nabla\psi(x^k) + \nabla M(x^k)a_q}{\|\nabla\psi(x^k) + \nabla M(x^k)a_q\|} \tag{16}$$

satisfies $\left\|\hat{s}^{j^*}\right\| \leq \Delta_k$ and (7) *(in place of s^k).*

Proof. First note that

$$\begin{aligned}\kappa_\mathrm{fH} \geq \left\|\nabla^2\psi(x)\right\| + \sum_{i=1}^p \|a_i\| \left\|\nabla^2 m^{F_i}(x)\right\| &\geq \left\|\nabla^2\psi(x)\right\| + \sum_{i=1}^p [a_q]_i \left\|\nabla^2 m^{F_i}(x)\right\| \\ &\geq \left\|\nabla^2\psi(x) + \sum_{i=1}^p [a_q]_i \nabla^2 m^{F_i}(x)\right\|\end{aligned} \tag{17}$$

for any $x \in \mathcal{L}_\mathrm{max}$ and any a_q.

Because $\kappa_\mathrm{d} \in (0,1)$, $\kappa_\mathrm{d}^{i+1} \leq \kappa_\mathrm{d}^{\lceil i \rceil} \leq \kappa_\mathrm{d}^i$ for any $i \geq 0$, Equation (15) implies that

$$\kappa_\mathrm{d} \min\left\{1, \frac{\|\nabla\psi(x^k) + \nabla M(x^k)a_q\|}{\kappa_\mathrm{fH}\Delta_k}\right\} \leq \kappa_\mathrm{d}^{j^*} \leq \frac{\|\nabla\psi(x^k) + \nabla M(x^k)a_q\|}{\kappa_\mathrm{fH}\Delta_k}. \tag{18}$$

Note that $\left\|\hat{s}^{j^*}\right\| \leq \Delta_k$ follows from $\kappa_\mathrm{d} \in (0,1)$. Since whenever x^k is updated, the denominator of (10) is positive and $\rho_k > \eta_1 > 0$,

$$\begin{aligned}0 &< \psi(x^k) - \psi(x^k + s^k) + h^{(k)}(F(x^k)) - h^{(k)}(F(x^k + s^k)) \\ 0 &< \psi(x^k) - \psi(x^k + s^k) + h(F(x^k)) - h(F(x^k + s^k)) \\ f(x^k + s^k) &< f(x^k).\end{aligned}$$

Therefore, $x^k \in \mathcal{L}$ for all iterations and $x^k + \hat{s}^{j^*} \in \mathcal{L}_\mathrm{max}$, where $\mathcal{L}$ and $\mathcal{L}_\mathrm{max}$ are defined in Assumption 2.

For $s \in \mathbb{R}^n$ and any fixed a_q satisfying the hypotheses of the lemma, define

$$\hat{m}(s) \triangleq \psi(x^k + s) + \left\langle M(x^k + s), a_q \right\rangle.$$

Since ψ and M are twice continuously differentiable (Assumption 2 and Assumption 1, respectively) on $\mathcal{L}_\mathrm{max}$, Taylor's theorem provides an $\xi \in (x^k, x^k + s)$ so that

$$\begin{aligned}\hat{m}(0) - \hat{m}(s) &= -\langle \nabla\psi(x^k) + \nabla M(x^k)a_q, s\rangle - \frac{1}{2}\langle s, \nabla^2\psi(\xi)s + \sum_{i=1}^p [a_q]_i \nabla^2 m^{F_i}(\xi)s\rangle \\ &\geq -\langle \nabla\psi(x^k) + \nabla M(x^k)a_q, s\rangle - \frac{1}{2}\|s\|^2\kappa_\mathrm{fH},\end{aligned}$$

because $\xi \in \mathcal{L}_\mathrm{max}$ and κ_fH satisfies (17). Setting $s \triangleq \hat{s}^{j^*}$ in the last expression yields

$$\begin{aligned}&-\langle \nabla\psi(x^k) + \nabla M(x^k)a_q, \hat{s}^{j^*}\rangle - \frac{1}{2}\|\hat{s}^{j^*}\|^2\kappa_\mathrm{fH} \\ &= \kappa_\mathrm{d}^{j^*}\Delta_k\left(1 - \frac{\kappa_\mathrm{d}^{j^*}\kappa_\mathrm{fH}\Delta_k}{2\left\|\nabla\psi(x^k) + \nabla M(x^k)a_q\right\|}\right)\left\|\nabla\psi(x^k) + \nabla M(x^k)a_q\right\| \\ &\geq \frac{1}{2}\kappa_\mathrm{d}^{j^*}\Delta_k\left\|\nabla\psi(x^k) + \nabla M(x^k)a_q\right\|,\end{aligned}$$

where the last term is obtained from the upper bound in (18). If $j^* = 0$, then (7) immediately follows from noting that $\nabla\psi(x^k) + \nabla M(x^k)a_q \in \mathfrak{G}^k$ implies that $\|\nabla\psi(x^k) + \nabla M(x^k)a_q\| \geq \|g^k\|$.

If $j^* \geq 1$, then $j^* = \left\lceil \log_{\kappa_{\rm d}} \left(\frac{\left\|\nabla\psi(x^k)+\nabla M(x^k)a_q\right\|}{\kappa_{\rm fH}\Delta_k} \right) \right\rceil$. For $\kappa_{\rm d} \in (0,1)$ and any $c \geq 0$, it follows that $\kappa_{\rm d}^{\lceil c \rceil} \geq \kappa_{\rm d}^{c+1}$ and thus $\kappa_{\rm d}^{j^*} \geq \kappa_{\rm d} \frac{\left\|\nabla\psi(x^k)+\nabla M(x^k)a_q\right\|}{\kappa_{\rm fH}\Delta_k}$. Using this relation and again noting that $\|\nabla\psi(x^k) + \nabla M(x^k)a_q\| \geq \|g^k\|$ therefore yields

$$\begin{aligned}\frac{1}{2}\kappa_{\rm d}^{j^*}\Delta_k \left\|\nabla\psi(x^k) + \nabla M(x^k)a_q\right\| &\geq \frac{1}{2\kappa_{\rm fH}}\kappa_{\rm d}\left\|\nabla\psi(x^k) + \nabla M(x^k)a_q\right\|^2 \\ &\geq \frac{1}{2\kappa_{\rm fH}}\kappa_{\rm d}\left\|g^k\right\|^2,\end{aligned}$$

which completes the proof. □

4.2 Stationarity of cluster points

We now prove that cluster points of the sequence of iterates generated by Algorithm 1 are Clarke stationary; the proof used the following sequence of results:

Lemma 4 shows that when the trust-region radius Δ_k is a sufficiently small multiple of the norm of the master model gradient, $\left\|g^k\right\|$, the iteration is guaranteed to be successful.

Lemma 5 shows that $\lim\limits_{k\to\infty} \Delta_k = 0$.

Lemma 6 shows that a subsequence of master model gradients g^k must go to zero as well, as $k \to \infty$.

Lemma 7 shows that zero is in the generalized Clarke subdifferential $\partial_{\rm C} f(x^*)$ of any cluster point x^* of any subsequence of iterates where the master model gradients go to zero.

Theorem 4.1 shows that $0 \in \partial_{\rm C} f(x^*)$ for any cluster point x^* of the sequence of iterates generated by Algorithm 1.

We demonstrate in the following lemma that building a master model gradient g^k from a particular combination of the component model gradients ensures a successful iteration if Δ_k is sufficiently small. (The particular combination is defined by manifolds that are active in the trust region.)

Lemma 4. *Let* Assumptions 1 *and* 2 *hold. If an iteration is acceptable and*

$$\Delta_k < \frac{\kappa_{\rm d}(1-\eta_1)}{4\kappa_{\rm f} L_h}\|g^k\|, \tag{19}$$

then $\rho_k > \eta_1$ *in* Algorithm 1*, and the iteration is successful.*

Proof. Since the iteration is acceptable, $g^k \neq 0$, and the bound on Δ_k is positive. By Lemma 2, there exists some $h^{(k)}$ with gradient $a^{(k)}$ satisfying (8). Therefore,

$$1 - \rho_k \leq |\rho_k - 1| = \left| \frac{\psi(x^k) - \psi(x^k + s^k) + h^{(k)}(F(x^k)) - h^{(k)}(F(x^k + s^k))}{\psi(x^k) - \psi(x^k + s^k) + \langle M(x^k) - M(x^k + s^k), a^{(k)} \rangle} - 1 \right| \tag{20}$$

Since M is a fully linear model of F on $\mathcal{B}(x^k;\Delta_k)$ by Assumption 1, F and h satisfy Assumption 2, and $\left\|s^k\right\| \leq \Delta_k$, combining the two terms on the right-hand side of (20) produces a numerator that satisfies

$$\begin{aligned}&\left| h^{(k)}(F(x^k)) - h^{(k)}(F(x^k + s^k)) - \left\langle M(x^k) - M(x^k + s^k), a^{(k)} \right\rangle \right| \\ &= \left| \left\langle F(x^k), a^{(k)} \right\rangle - \left\langle F(x^k + s^k), a^{(k)} \right\rangle - \left\langle M(x^k) - M(x^k + s^k), a^{(k)} \right\rangle \right| \\ &\leq \left\| F(x^k) - M(x^k) \right\| \left\| a^{(k)} \right\| + \left\| F(x^k + s^k) - M(x^k + s^k) \right\| \left\| a^{(k)} \right\| \\ &\leq 2\kappa_{\rm f} L_h \Delta_k^2. \end{aligned} \tag{21}$$

Applying (21) to the numerator of (20) and (7) to the denominator of (20), we have

$$\begin{aligned} 1-\rho_k &\leq \frac{2\kappa_{\mathrm{f}} L_h \Delta_k^2}{\psi(x^k)-\psi(x^k+s^k)+\langle M(x^k)-M(x^k+s^k), a^{(k)}\rangle} \\ &\leq \frac{4\kappa_{\mathrm{f}} L_h \Delta_k^2}{\kappa_{\mathrm{d}}\left\|g^k\right\| \min\left\{\Delta_k, \frac{\|g^k\|}{\kappa_{\mathrm{mH}}}\right\}} \\ &\leq \frac{4\kappa_{\mathrm{f}} L_h \Delta_k^2}{\kappa_{\mathrm{d}}\left\|g^k\right\| \Delta_k}, \end{aligned} \tag{22}$$

where the last inequality comes from (11). Applying (19) to (22) leaves

$$1-\rho_k \leq \frac{4\kappa_{\mathrm{f}} L_h \Delta_k}{\kappa_{\mathrm{d}}\left\|g^k\right\|} < 1-\eta_1.$$

Thus, $\rho_k > \eta_1$ if Δ_k satisfies (19), and the iteration is successful. □

The next lemma shows that the sequence of manifold sampling trust-region radii converges to zero.

Lemma 5. *Let* Assumptions 1 *and* 2 *hold. If* $\{x^k, \Delta_k\}_{k\in\mathbb{N}}$ *is generated by* Algorithm 1*, then the sequence* $\{f(x^k)\}_{k\in\mathbb{N}}$ *is nonincreasing, and* $\lim\limits_{k\to\infty} \Delta_k = 0$.

Proof. If iteration k is unsuccessful, then $\Delta_{k+1} < \Delta_k$, and $x^{k+1} = x^k$; therefore, $f(x^{k+1}) = f(x^k)$. On successful iterations k, $\rho_k > \eta_1$ and $\|g^k\| > 0$, and s^k satisfies (7) by construction. Using the definitions of $h^{(k)}$ and ρ_k and equations (7) and (11), we have

$$\begin{aligned} f(x^k)-f(x^{k+1}) &= \psi(x^k)-\psi(x^k+s^k)+h(F(x^k))-h(F(x^k+s^k)) \\ &\geq \psi(x^k)-\psi(x^k+s^k)+h^{(k)}(F(x^k))-h^{(k)}(F(x^k+s^k)) \\ &= \rho_k(\psi(x^k)-\psi(x^k+s^k)+\langle M(x^k)-M(x^k+s^k), a^{(k)}\rangle) \\ &\geq \eta_1\frac{\kappa_{\mathrm{d}}}{2}\|g^k\| \min\left\{\Delta_k, \frac{\|g^k\|}{\kappa_{\mathrm{mH}}}\right\} \\ &\geq \eta_1\frac{\kappa_{\mathrm{d}}}{2}\|g^k\|\Delta_k > 0. \end{aligned} \tag{23}$$

Thus, the sequence $\{f(x^k)\}_{k\in\mathbb{N}}$ is nonincreasing.

To show that $\Delta_k \to 0$, we consider the cases in which there are infinitely or finitely many successful iterations separately. First, suppose that there are infinitely many successful iterations, indexed by $\{k_j\}_{j\in\mathbb{N}}$. Since $f(x^k)$ is nonincreasing in k and f is bounded below (by Assumption 2), the sequence $\{f(x^k)\}_{k\in\mathbb{N}}$ converges to some limit $f^* \leq f(x^0)$. Thus, from (23), having infinitely many successful iterations (indexed $\{k_j\}_{j\in\mathbb{N}}$) implies that

$$\infty > f(x^0)-f^* \geq \sum_{j=0}^{\infty} f(x^{k_j})-f(x^{k_j+1}) > \sum_{j=0}^{\infty} \eta_1\Delta_{k_j}\left\|g^{k_j}\right\| > \sum_{j=0}^{\infty}\frac{\eta_1}{\eta_2}\Delta_{k_j}^2, \tag{24}$$

where the last inequality comes from the acceptability of all successful iterations. It follows that $\Delta_{k_j} \to 0$ for the sequence of successful iterations. Observe that $\Delta_{k_j+1} \leq \gamma_{\mathrm{inc}}\Delta_{k_j}$ and that $\Delta_{k+1} = \gamma_{\mathrm{dec}}\Delta_k < \Delta_k$ if iteration k is unsuccessful. Thus, for any unsuccessful iteration $k > k_j$, $\Delta_k \leq \gamma_{\mathrm{inc}}\Delta_q$, where $q \triangleq \max\{k_j : j \in \mathbb{N},\, k_j < k\}$. It follows immediately that

$$0 \leq \lim_{k\to\infty}\Delta_k \leq \gamma_{\mathrm{inc}}\lim_{j\to\infty}\Delta_{k_j} = 0,$$

and so $\Delta_k \to 0$ as required.

Next, suppose there are only finitely many successful iterations; let $\nu \in \mathbb{N}$ be the number of successful iterations. Since $\gamma_{\mathrm{dec}} < 1 \leq \gamma_{\mathrm{inc}}$, it follows that $0 \leq \Delta_k \leq \gamma_{\mathrm{inc}}^{\nu}\gamma_{\mathrm{dec}}^{k-\nu}\Delta_0$ for each $k \in \mathbb{N}$. Thus, $\Delta_k \to 0$ as required. □

We now show that the norms of the master model gradients are not bounded away from zero.

Lemma 6. *Let* Assumptions 1 *and* 2 *hold. If the sequence* $\{x^k, \Delta_k\}_{k\in\mathbb{N}}$ *is generated by* Algorithm 1*, then* $\liminf_{k\to\infty} \|g^k\| = 0$.

Proof. To obtain a contradiction, suppose there is an iteration j and some $\epsilon > 0$ for which $\|g^k\| \geq \epsilon$, for all $k \geq j$. Algorithm 1 guarantees that $\Delta_j \geq \gamma_{\text{dec}}^j \Delta_0 > 0$. Moreover, any iteration where $\Delta_k \leq C \left\|g^k\right\|$ for

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

will be successful because the conditions of Lemma 4 are then satisfied. Therefore, by the contradiction hypothesis, any $k \geq j$ satisfying $\Delta_k \leq C\epsilon$ is guaranteed to be successful, in which case $\Delta_{k+1} = \gamma_{\text{inc}}\Delta_k \geq \Delta_k$. On the other hand, if $\Delta_k \geq C\epsilon$, then $\Delta_{k+1} \geq \gamma_{\text{dec}}\Delta_k$. A straightforward inductive argument then yields $\Delta_k \geq \min(\gamma_{\text{dec}}C\epsilon, \Delta_j) > 0$ for all $k \geq j$, contradicting Lemma 5. Thus, no such (j, ϵ) pair exists, and so $\liminf_{k\to\infty} \|g^k\| = 0$. □

The next lemma shows that subsequences of iterates with master model gradients converging to 0 have cluster points that are Clarke stationary (Definition 3). Algorithm 1 generates at least one such subsequence of iterates by Lemma 6.

Lemma 7. *Let* Assumptions 1–3 *hold, and let* $\{x^k, \Delta_k, g^k\}_{k\in\mathbb{N}}$ *be a sequence generated by* Algorithm 1*. For any subsequence* $\{k_j\}_{j\in\mathbb{N}}$ *of acceptable iterations such that both*

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

and $\{x^{k_j}\}_{j\in\mathbb{N}} \to x^*$ *for some cluster point* x^**, then* $0 \in \partial_{\text{C}} f(x^*)$.

Proof. Let I^k contain the indices of selection functions of h represented in $\mathfrak{G}^k$, and let $J^k \triangleq I_h^e(F(x^*))$. Since $\Delta_k \to 0$ by Lemma 5, $\{x^{k_j}\}_{j\in\mathbb{N}}$ converges to x^* by assumption, and piecewise linear functions are piecewise differentiable in the sense of Scholtes [30], Assumption 3 implies that, for k sufficiently large, only selection functions that are essentially active at x^* are represented in $\mathfrak{G}^k$. Consequently, $I^{k_j} \subseteq J^{k_j}$. By Lemma 1 with $I \leftarrow I^{k_j}$, $J \leftarrow J^{k_j}$, $x \leftarrow x^{k_j}$, $y \leftarrow x^*$, and $\Delta \leftarrow \Delta_{k_j}$, there exists $v(g^{k_j}) \in \partial_{\text{C}} f(x^*)$ for each g^{k_j} so that

$$\|g^{k_j} - v(g^{k_j})\| \leq c_2 \Delta_{k_j},$$

with c_2 defined by (13). Thus, by the acceptability of every iteration indexed by k_j,

$$\|g^{k_j} - v(g^{k_j})\| \leq c_2\eta_2 \|g^{k_j}\|,$$

and so

$$\|v(g^{k_j})\| \leq (1 + c_2\eta_2)\|g^{k_j}\|.$$

Since $\|g^{k_j}\| \to 0$ by assumption, therefore $\left\|v(g^{k_j})\right\| \to 0$. Proposition 7.1.4 in [14] then yields the claimed result, by establishing that $\partial_{\text{C}} f$ is *outer-semicontinuous* and therefore $0 \in \partial_{\text{C}} f(x^*)$. □

Theorem 4.1. *Let* Assumptions 1–3 *hold. If* x^* *is a cluster point of a sequence* $\{x^k\}$ *generated by* Algorithm 1*, then* $0 \in \partial_{\text{C}} f(x^*)$.

Proof. First, suppose that there are only finitely many successful iterations and k' is the last.

Suppose for contradiction that $0 \notin \partial_{\text{C}} f(x^{k'})$. By continuity of F_i (Assumption 2), there exists $\bar{\Delta} > 0$ so that for all $\Delta \in [0, \bar{\Delta}]$, the manifolds active in $\mathcal{B}(x^{k'}; \bar{\Delta})$ are precisely the manifold active at $x^{k'}$; that is,

$$I_h^e(x^{k'}) = \bigcup_{y\in\mathcal{B}(x^{k'};\Delta)} I_h^e(y) \qquad \text{for all } \Delta \leq \bar{\Delta}.$$

By assumption, Δ_k decreases by a factor of γ_{dec} in each iteration after k' since every iteration after k' is unsuccessful. There is therefore a least iteration $k'' \geq k'$ so that $\Delta_{k''} \leq \bar{\Delta}$. By Assumption 3, for each $k \geq k''$, $\mathfrak{G}^k$ contains all manifolds at $x^{k'}$, and therefore $(\nabla\psi(x^k) + \nabla M(x^k)a^{(k)}) \in \mathfrak{G}^k$. Since k' is the last successful iteration, $x^k = x^{k'}$ for all $k \geq k'' \geq k'$. Consequently, the conditions for Lemma 1 hold for $x \leftarrow x^k$, $y \leftarrow x^{k'}$, (noting that $x^k = x^{k'}$) $\Delta \leftarrow 0$, $\mathfrak{G} \leftarrow \mathfrak{G}^k$, and $\mathcal{H} \leftarrow \partial_{\mathrm{C}} f(x^{k'})$; thus, for each $k \geq k''$, $g^k \in \partial_{\mathrm{C}} f(x^{k'})$.

Since $0 \notin \partial_{\mathrm{C}} f(x^{k'})$ by supposition, $v^* \triangleq \mathbf{proj}(0, \partial_{\mathrm{C}} f(x^{k'}))$ is nonzero, and so

$$\|g^k\| \geq \|v^*\| > 0 \qquad \text{for all } k \geq k''. \tag{25}$$

Since $\Delta_k \to 0$, Δ_k will satisfy the conditions of Lemma 4 for k sufficiently large: there will be a successful iteration contradicting k' being the last.

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

$$x^{k_j} \to x^* \text{ and } \|g^{k_j}\| \to 0. \tag{26}$$

If the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges, then the subsequence $\left\{x^{k_j}\right\}_{j\in\mathbb{N}}$ from Lemma 6 satisfies (26). Otherwise, if the sequence $\{x^k\}_k$ is not convergent, we will show that $\liminf_{k\to\infty}(\max\{\|x^k - x^*\|, \|g^k\|\}) = 0$ for each cluster point x^*. Suppose for contradiction that there exist $\bar{\nu} > 0$, an iteration $\bar{k}$, and a cluster point x^* of the sequence $\{x^k\}$ with the following property. Considering the infinite set

$$\mathcal{K} \triangleq \{k : k \geq \bar{k}, \|x^k - x^*\| \leq \bar{\nu}\},$$

suppose that the subsequence $\left\{x^k\right\}_{k\in\mathcal{K}}$ converges to x^* and that $\|g^k\| > \bar{\nu}$ for all $k \in \mathcal{K}$. From (24), we have that

$$\eta_1 \sum_{k\in\mathcal{K}} \|g^k\| \|x^{k+1} - x^k\| \leq \eta_1 \sum_{k=0}^{\infty} \|g^k\| \|x^{k+1} - x^k\| < \infty, \tag{27}$$

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 $k \in \mathcal{K}$, we conclude from (27) that

$$\sum_{k\in\mathcal{K}} \|x^{k+1} - x^k\| < \infty. \tag{28}$$

Since $x^k \not\to x^*$, there exists some $\hat{\nu} \in (0, \bar{\nu})$ for which, for each $k' \in \mathcal{K}$, there exists

$$q(k') \triangleq \min\{\kappa \in \mathbb{N} : \kappa > k', \quad \|x^\kappa - x^{k'}\| > \hat{\nu}\}.$$

From this construction, since $\hat{\nu} < \bar{\nu}$, then $\{k', k'+1, \ldots, q(k') - 1\} \subset \mathcal{K}$.

By (28), for $\hat{\nu}$ there exists $N \in \mathbb{N}$ such that

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

Taking $k' \geq N$, by the triangle inequality, we have

$$\hat{\nu} < \|x^{q(k')} - x^{k'}\| \leq \sum_{i\in\{k',k'+1,\ldots,q(k')-1\}} \|x^{i+1} - x^i\| \leq \sum_{\substack{k\in\mathcal{K}\\ k\geq N}} \left\|x^{k+1} - x^k\right\| \leq \hat{\nu}.$$

Therefore, $\hat{\nu} < \hat{\nu}$, a contradiction. Therefore $\liminf_{k\to\infty}(\max\{\|x^k - x^*\|, \|g^k\|\}) = 0$ for all cluster points x^*, and there is a subsequence satisfying (26). By Lemma 7, $0 \in \partial_{\mathrm{C}} f(x^*)$ for all such subsequences. □

5 Implementation and Experimentation

We implemented manifold sampling for piecewise linear functions (MS4PL) in MATLAB. The parameters used in the MS4PL implementation of Algorithm 1 were $\eta_1 = 0.05$, $\kappa_{\rm d} = 10^{-4}$, $\eta_2 = 10^4$, $\gamma_{\rm dec} = 0.5$, $\gamma_{\rm inc} = 2$, $\Delta_{\max} = 10^8$, and $\Delta_0 = 0.1$. Fully linear quadratic models of the component functions F_i were formed at Line 4 of Algorithm 1 using the routine from POUNDerS [32], whereby an explicit value of $\kappa_{\rm mH}$ is not used in model building. In the implementation tested here, $\kappa_{\rm mH}$ is also not used when checking for descent; this can be viewed as effectively setting $\kappa_{\rm mH}$ to zero in (7). We consider two options for constructing the initial generator set at Line 6: what we denote as MS4PL-1 uses

$$\mathfrak{G}^k \triangleq \left\{ \nabla\psi(x^k) + \nabla M(x^k)\, a_i : i \in I_h^e(F(x^k)) \right\},$$

and what we denote as MS4PL-2 uses

$$\mathfrak{G}^k \triangleq \left\{ \nabla\psi(x^k) + \nabla M(x^k)\, a_i : i \in I_h^e(F(y)),\, y \in Y \right\},$$

where Y is all points in $\mathcal{B}(x^k; \Delta_k)$ that have already been evaluated during the given run. (Such points are past iterates of the algorithm as well as points evaluated in order to construct models of the components of F.) MS4PL then determines the minimum-norm element of $\mathfrak{G}^k$ by solving (4) using a specialized active set method from [11]. These weights define the master model of f via (5). We then solve our trust-region subproblems on $\mathcal{B}(x^k; \Delta_k)$ using GQT [26].

The determination of $h^{(k)}$ in MS4PL requires some care; for many problems, $|\mathfrak{H}|$ is so large that evaluating $h_i(F(x^k))$ and $h_i(F(x^k + s^k))$ for all $h_i \in \mathfrak{H}$ is unnecessarily expensive. Often, evaluating h at $F(x)$ may return only the value of $h(F(x))$ and $I_h^e(F(x))$ (for I_h^e defined in Definition 5). Therefore, MS4PL uses the following procedure to identify $h^{(k)}$. The implementation first checks whether any selection function h_i for i in $I_h^e\left(F(x^k)\right) \cup I_h^e\left(F(x^k + s^k)\right)$ satisfies (8), picking the function that predicts the largest decrease in f between x^k and $x^k + s^k$. That is,

$$h^{(k)} \leftarrow \operatorname*{arg\,max}_{i \in I_h^e(F(x^k)) \cup I_h^e(F(x^k+s^k))} \{h_i(F(x^k)) - h_i(F(x^k + s^k)) : h_i \text{ satisfies (8)}\},$$

breaking ties arbitrarily if necessary. If no selection function active at $F(x^k)$ or $F(x^k + s^k)$ satisfies (8), then determining $h^{(k)}$ requires considering selection functions that are active at neither $F(x^k)$ nor $F(x^k + s^k)$. In this case, we evaluate points on an increasingly refined grid between $F(x^k)$ and $F(x^k + s^k)$. If evaluating h at 1,024 evenly spaced points between $F(x^k)$ and $F(x^k + s^k)$ does not identify any h_i satisfying (8), only then will MS4PL resort to completely enumerating all elements of $\mathfrak{H}$.

Since $\Delta_{k+1} \geq \gamma_{\rm dec}\Delta_k$ for each k, the convergence rate of any implementation is fundamentally limited in some sense by the chosen values for $\gamma_{\rm dec}$ and $\gamma_{\rm inc}$. A complete study of this effect has not been performed.

We compared the performance of MS4PL with SLQP-GQ, an implementation gradient sampling, and GRANSO, a quasi-Newton method that uses a steering strategy to update the penalty parameter. Both solve sequential quadratic programs, and both have convergence results for nonconvex, nonsmooth functions [8, 9, 10, 25]. Both SLQP-GS and GRANSO are given true gradient values; MS4PL-1 and MS4PL-2 do not receive this information in our computational experiments. Even though SLQP-GS and GRANSO are given this (significant) additional information, we measure progress of all methods in terms of the number of function evaluations. That is, although we record the number of times SLQP-GS and GRANSO request an element of the subdifferential of f at a point, both are getting gradient information for free when measuring performance. As a point of reference, we also modified MS4PL-1 to use linear models built using ∇F (i.e., similar gradient information available to SLQP-GS and GRANSO). This implementation is denoted MS4PL-1-grad.

For our final comparison, we use a modified version of POUNDerS [32], which we denote PLC. This implementation is a model-based trust-region method that builds models of each component F_i around each iterate x^k and then combines this information into a smooth master model of f by using a single element of $\partial_{\rm B} f(x^k)$. This PLC implementation was run with $\eta_1 = 0.05$, $\gamma_{\rm dec} = 0.5$, $\gamma_{\rm inc} = 2$, $\Delta_{\max} = 10^8$, and $\Delta_0 = 0.1$. The trust-region subproblems in PLC were solved by using MINQ [28] on an ∞-norm trust region.

5.1 Test Problems

We benchmark all methods on a set of censored ℓ_1-loss functions. Given data $d \in \mathbb{R}^p$, censors $c \in \mathbb{R}^p$, and the mapping $F : \mathbb{R}^n \to \mathbb{R}^p$, we define

$$f(x) = \sum_{i=1}^{p} |d_i - \max\{F_i(x), c_i\}| .$$

In other words, ψ is the zero function, and $h(z) = \sum_{i=1}^{p} |d_i - \max\{z_i, c_i\}|$. This nonconvex, piecewise-linear loss function is discussed in [33]; the loss function penalizes deviation of $F_i(x)$ from target data d_i, but only if $F_i(x)$ is larger than the censor value c_i.

For these problems, the gradients of the selection functions that are active at $z \in \mathbb{R}^p$ are given by

$$\nabla h_i = \begin{cases} \mathbf{sign}\,(z_i - d_i) & \text{if } z_i > c_i \\ \{0, \mathbf{sign}\,(z_i - d_i)\} & \text{if } z_i = c_i \\ 0 & \text{if } z_i < c_i, \end{cases} \qquad i = 1, \ldots, p,$$

where

$$\mathbf{sign}\,(z) = \begin{cases} 1 & \text{if } z > 0 \\ \{-1, 1\} & \text{if } z = 0 \\ -1 & \text{if } z < 0. \end{cases}$$

To generate different problem instances, we define F by the 53 vector mappings in [27, Section 4], which satisfy $2 \leq n \leq 12$ and $2 \leq p \leq 45$. We then define the data d and censors c in a manner that attempts to introduce points of nondifferentiability to the problem. For components $2 \leq i \leq p$, we draw c_i uniformly from $[l_i, u_i]$ where

$$l_i = \min\left\{F_i(x^0), F_i(x^*)\right\} \qquad \text{and} \qquad u_i = \max\left\{F_i(x^0), F_i(x^*)\right\},$$

where x^0 is a starting point from [27] and x^* is a known approximate minimizer to the problem

$$\underset{x}{\text{minimize}}\, f(x) \triangleq \sum_{i=1}^{p} \|F_i(x)\| . \tag{29}$$

If we make the (crude) assumption that $F_i(x)$ is also drawn uniformly from $[l_i, u_i]$, then $\max\{c_i, z_i\}$ follows the modified beta distribution $(u_i - l_i) * \beta(2, 1) + l_i$; for the $p - 1$ components components in question, we therefore draw d_i from this distribution. For each of the 53 problems, we draw 10 random instances of c_i and d_i from their distributions resulting in 530 benchmark problems. We cannot censor all components F_i in this fashion because if $F_i(x^0) \leq F_i(x^*)$ for all i, every component of $F_i(x^0)$ will be censored, thereby causing the starting point x^0 to be Clarke stationary. We therefore set $c_1 = -\infty$ and $d_1 = 0$. The 10 sets of c and d for each problem are available at [22].

These problems have the useful property that elements of $\partial_{\mathrm{B}} f$ are fairly easy to calculate. Therefore, we provide this information to SLQP-GS and GRANSO. We also use gradient information when benchmarking, namely the results in Figures 3 and 4 below.

Figure 1 shows contour plots for the tenth benchmark instance of three of the two-dimensional benchmark problems. We record the manifolds observed when evaluating the 200×200 points in the contour plots and then uniquely number each manifold. This allows us to visualize (a subset of) the manifolds present.

5.2 Comparison of algorithms

We ran MS4PL-1, MS4PL-2, MS4PL-1-grad, PLC, SLQP-GS, and GRANSO on the 530 benchmark problem instances outlined above. All implementations were given $500(n_p + 1)$ function evaluations where n_p is the dimension of problem p. We use data profiles to compare their ability to solve the benchmark problems. Data

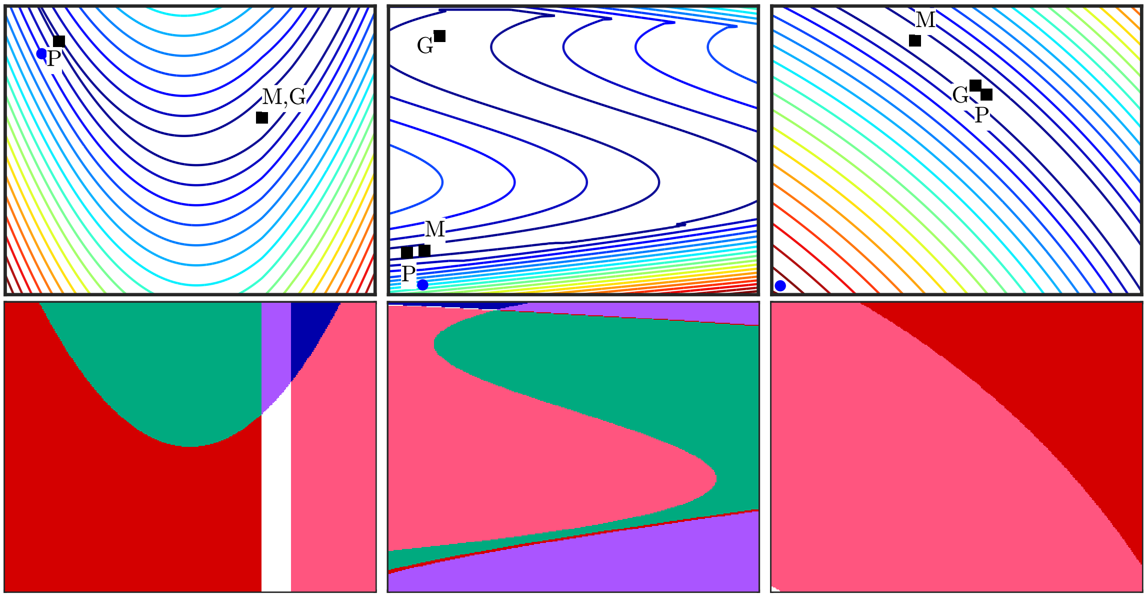


Figure 1: Contour manifold plots for the tenth instance of three of the two-dimensional test problems. A blue circle denotes the problem starting point; and M, G, and P denote the approximate solutions returned by MS4PL-1, GRANSO, and PLC, respectively. From left to right, the number of manifolds shown are 6, 6, and 3, respectively.

profiles show the fraction of problems solved to some level τ after a given number of function evaluations. In an attempt to normalize for problems with higher dimension n_p (as such problems are assumed to be more difficult), the number of function evaluations are grouped into $n_p + 1$ batches.

Formally, if $t_{p,s}$ is the number of function evaluations required for implementation s to solve problem p in some set of problems P, then the data profile is

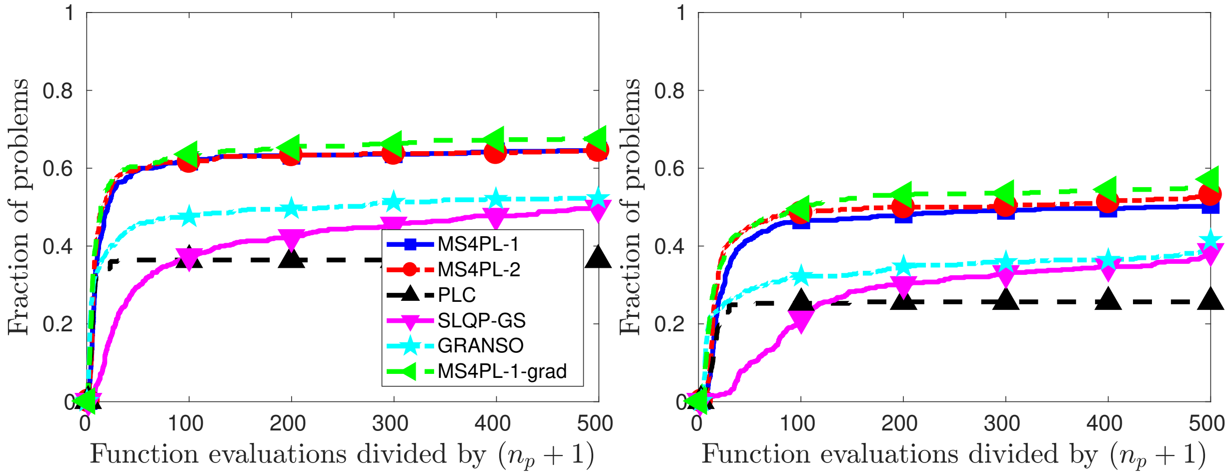
$$d_s(\alpha) = \frac{\left|\left\{p : \frac{t_{p,s}}{n_p+1} \leq \alpha\right\}\right|}{|P|}.$$

All that remains is to define when we consider an implementation to have solved a problem p to a level τ. We first examine convergence by observing how the sequence of objective function values approach the best-found function value by any of the implementations. We consider an implementation s to have solved problem p to a level τ after j evaluations if

$$f(x^0) - f(x^j) \geq (1-\tau)(f(x^0) - \tilde{f}_p), \tag{30}$$

where x^0 is the problem's starting point, x^j is the jth point evaluated by a given implementation, and $\tilde{f}_p$ is the best-found function value by any implementation on problem p. For example, if $\tau = 0.1$, the convergence test in (30) considers an implementation to have solved problem p when a point is evaluated with 90% of the possible decrease on the problem (for the implementations being compared). Figure 2 shows data profiles for all implementations for two values of τ.

The data profile values at $500(n_p + 1)$ show that all of the theoretically convergent implementations find 99% of the best-found decrease on at least half of the benchmark problems. Given that SLQP-GS and GRANSO are given exact gradient information whenever they request it, the success of MS4PL-1 and MS4PL-2 is especially stark. (SLQP-GS requests gradient information on approximately 80% of its function evaluations; GRANSO requests gradient information on every function evaluation.) Examination of the



(a) $\tau = 10^{-2}$

(b) $\tau = 10^{-5}$

Figure 2: Data profiles for 6 solvers for convergence test (30) measuring success in terms of function values.

iterates produced by SLQP-GS appears to suggest that a re-starting mechanism causes iterations to jump to different parts of the domain. The other implementations do not display this behavior.

Only slight differences are observed between initializing $\mathfrak{G}^k$ with information at x^k or information at all previously evaluated points in $\mathcal{B}(x^k; \Delta_k)$ (MS4PL-1 and MS4PL-2, respectively). The additional gradient information available to MS4PL-1-grad only slightly improves the performance of manifold sampling in this metric. The smooth method PLC that is considering only a single manifold when constructing its smooth master model performs noticeably worse than the other implementations.

Considering only function values when comparing the performance of implementations of local optimization algorithms on nonconvex benchmark problems may be deceiving. An implementation may be stopping at a point with worse function value because it has converged to a stationary point. We therefore show data profiles based on measuring the approximate stationarity of points evaluated by each implementation.

For convex h, Yuan [35] presents a useful measure of stationarity at any point. Unfortunately, this metric is inappropriate for nonconvex h because a large decrease in objective value may exist arbitrarily close to a stationary point to which some implementation has converged. We therefore resort to sampling gradients in a ball around each point evaluated by each implementation and then computing a minimum-norm element in the convex hull of these points. Specifically, we evaluate the generalized subdifferential of h at 30 points drawn uniformly in a ball of radius 10^{-8} around each point evaluated by each implementation. We set $\tilde{g}(x^j)$ to be the minimum-norm element of the union of these 30 generalized subdifferentials around each point x^j evaluated by each algorithm. That is,

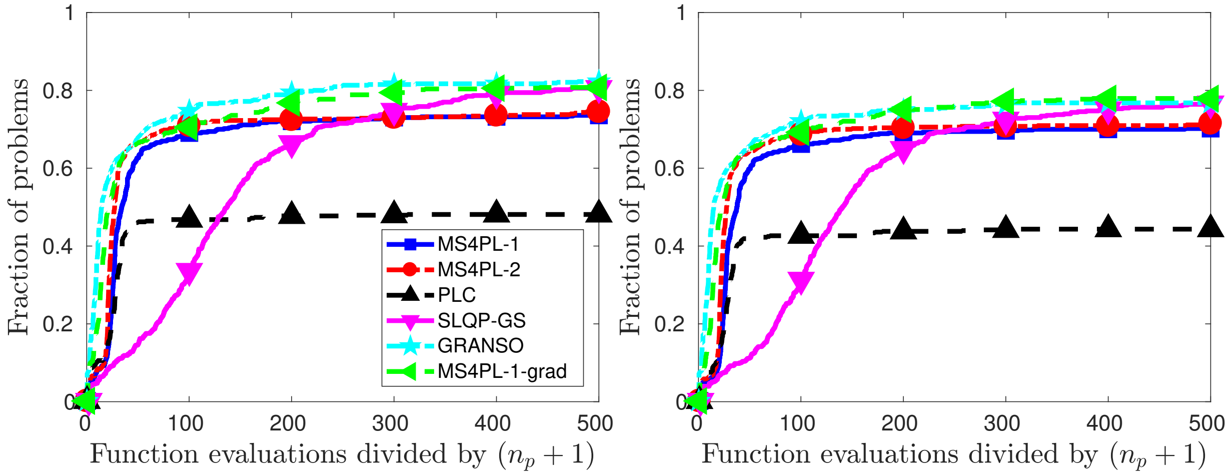
$$\tilde{g}(x^j) = \mathbf{proj}\left(0, \bigcup_{l=1}^{30} \left\{\nabla F(x^l) \nabla h_i(F(x^l)) : x^l \in \mathcal{B}(x^j; 10^{-8}), i \in I_h^e(F(x^l))\right\}\right).$$

(For all observed cases, $\left\| I_h^e(F(x^l)) \right\| = 1$.) We consider an implementation to have solved a problem to a level τ after j function evaluations if

$$\left\| \tilde{g}(x^j) \right\| \leq \tau \left\| \tilde{g}(x^0) \right\|. \tag{31}$$

Note that knowledge of $\partial_{\mathrm{B}} f$ is used to benchmark all implementations; only SLQP-GS and GRANSO use elements of $\partial_{\mathrm{B}} f$ when running.

Data profiles using the stationarity measure convergence test (31) are shown in Figure 3. For a tight tolerance, $\tau = 10^{-7}$, the theoretically convergent implementations find stationary points for between 70%



(a) $\tau = 10^{-3}$

(b) $\tau = 10^{-7}$

Figure 3: Data profiles for 6 solvers for convergence test (31) measuring success in terms of approximate subdifferentials.

and 80% of the benchmark problems. We note that increasing τ to 10^{-3} results in each implementation solving only approximately 5% more problems as measured by (31). This fact suggests that the benchmark problems are either being solved or not. In general, the performance of all implementations is relatively robust with respect to the level τ, which is an endorsement for the relative quality of benchmark problem set with respect to the convergence test (31).

Even though PLC is building smooth models m^{F_i} and combining them with an arbitrary element of $\partial_{\mathrm{B}} f$, it is unable to find stationary points at the largest tolerance τ on even half of the problems. Note that the difference between PLC and the other implementations is much larger than the differences between any of the manifold sampling or gradient sampling implementations. When gradient information is available (i.e., MS4PL-1-grad), manifold sampling's ability to find stationary points is further improved, and comparable with GRANSO.

Overall, Figures 2 and 3 show that the manifold sampling implementations that have access only to values of F but utilize information about the manifolds of h are competitive with gradient-based methods that are given access to true elements of $\partial_{\mathrm{B}} f$. For the problems considered, we find that when the derivative of F is unavailable, exploiting knowledge of h is nearly as valuable as having access to subgradients of f.

Data profiles built using function values (Figure 2) reveal complementary information to data profiles built using approximate gradient values (Figure 3) for some problems. For example, Figure 4 shows how function values and normalized gradient-norm values progress as MS4PL-1, MS4PL-2, and PLC are run on a single benchmark problem. The best function value found by MS4PL-1 is 0.4; but after $500(n+1)$ function evaluations, $\left\|\tilde{g}(x^j)\right\|$ has not approached zero. For this problem, MS4PL-2 has identified a point x^{70} satisfying $\left\|\tilde{g}(x^{70})\right\| = 2.6 \times 10^{-8}$ and $f(x^{70}) = 1.2$. (The common starting point for this problem satisfies $f(x^0) = 1.3 \times 10^3$ and $\left\|\tilde{g}(x^0)\right\| = 241.2$.)

6 Discussion

We are interested in generalizing the convergence results for manifold sampling to the case where h is a selection of a finite set of continuous, but not necessarily piecewise linear, functions. Extending the above analysis hinges critically on showing that some selection function (or combination of selection functions) can suitably approximate the behavior of h within the trust region. If h is a selection of functions $\mathfrak{H}$ that are

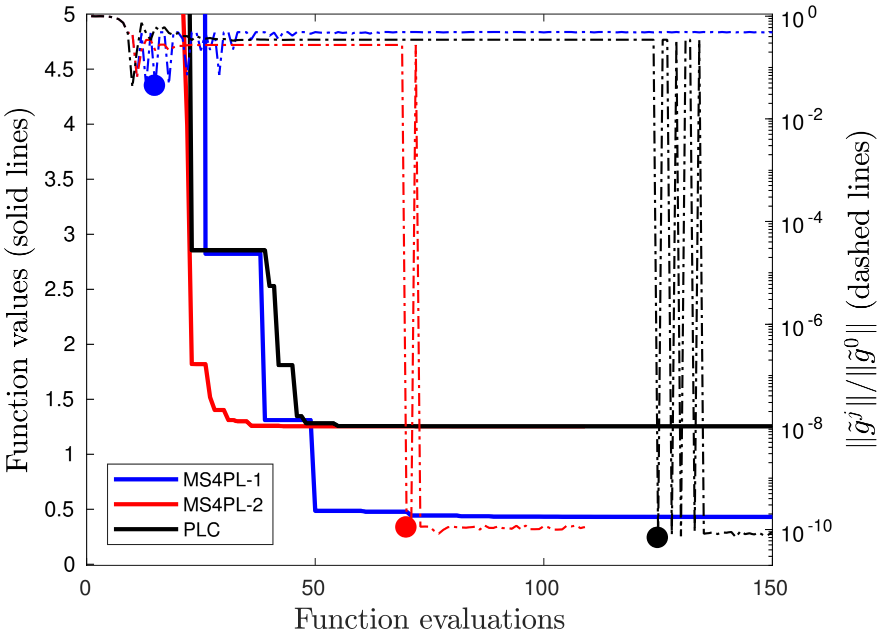


Figure 4: Progress of $f(x^j)$ and normalized $\|\tilde{g}(x^j)\|$ values for three implementations on the first instance of problem 8. Dots appear at the minimum of $\|\tilde{g}(x^j)\|$ for each method.

continuous but not necessarily affine, simple cases exist where no $h^{(k)} \in \mathfrak{H}$ satisfies (8). We therefore lack a selection function to use within the definition of ρ_k in (10).

We have considered analyzing methods that require knowledge of which functions in $\mathfrak{H}$ are essentially active between x^k and $x^k + s^k$; this requirement seems unreasonable since F is assumed to have a relatively unknown structure. Whereas the current approach requires information only about the selection functions at $F(x^k)$ and $F(x^k + s^k)$ (or possibly information on the line $\left[F(x^k), F(x^k + s^k)\right]$), learning information about F on the line $\left[x^k, x^k + s^k\right]$ could require significantly more evaluations of F and therefore add significant computational expense when F is expensive to evaluate. Therefore, our current research effort has focused on determining a theoretically suitable and computationally practical $h^{(k)}$ and ρ_k that do not require significant information about the behavior of F between x^k and $x^k + s^k$.

Establishing convergence rates for manifold sampling currently remains elusive because of the nonconvexity in h. The results in [16] critically rely on the convexity of h. Also, deterministic rates would seem to require knowledge of all manifolds that are active in each trust region radius. Probabilistic rates may be possible.

Acknowledgments

This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-06CH11357. We are grateful to the anonymous referees for their thoughtful suggestions.

References

[1] A. Aravkin, J. V. Burke, A. Chiuso, and G. Pillonetto, *Convex vs non-convex estimators for*

regression and sparse estimation: The mean squared error properties of ARD and GLasso, Journal of Machine Learning Research, 15 (2014), pp. 217–252, http://jmlr.org/papers/v15/aravkin14a.html.

[2] A. M. Bagirov, B. Karasözen, and M. Sezer, *Discrete gradient method: Derivative-free method for nonsmooth optimization*, Journal of Optimization Theory and Applications, 137 (2008), pp. 317–334, https://doi.org/10.1007/s10957-007-9335-5.

[3] Y. Bengio and Y. LeCun, *Scaling learning algorithms towards AI*, in Large-Scale Kernel Machines, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, eds., MIT Press, 2007, ch. 14, pp. 321–360.

[4] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi, *Dropping convexity for faster semi-definite optimization*, in 29th Annual Conference on Learning Theory, V. Feldman, A. Rakhlin, and O. Shamir, eds., vol. 49 of Proceedings of Machine Learning Research, Columbia University, New York, NY, 2016, pp. 530–582, http://proceedings.mlr.press/v49/bhojanapalli16.html.

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

[6] R. Collobert, F. Sinz, J. Weston, and L. Bottou, *Trading convexity for scalability*, in Proceedings of the 23rd international conference on Machine Learning, ACM Press, 2006, pp. 201–208, https://doi.org/10.1145/1143844.1143870.

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

[8] F. E. Curtis, *SLQP-GS*, 2017. http://coral.ise.lehigh.edu/frankecurtis/software.

[9] 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 (2017), pp. 148–181, https://doi.org/10.1080/10556788.2016.1208749.

[10] F. E. Curtis and M. L. Overton, *A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization*, SIAM Journal on Optimization, 22 (2012), pp. 474–500, https://doi.org/10.1137/090780201.

[11] F. E. Curtis and X. Que, *A quasi-Newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees*, Mathematical Programming Computation, 7 (2015), pp. 399–428, https://doi.org/10.1007/s12532-015-0086-2.

[12] D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis, *Nonsmooth optimization using Taylor-like models: error bounds, convergence, and termination criteria*, Tech. Report 1610.03446, ArXiv, Oct. 2016, https://arxiv.org/abs/1610.03446.

[13] D. Drusvyatskiy and A. S. Lewis, *Error bounds, quadratic growth, and linear convergence of proximal methods*, Mathematics of Operations Research, (2018), https://doi.org/10.1287/moor.2017.0889.

[14] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vol. II, Springer-Verlag, New York, 2003.

[15] 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, https://doi.org/10.1007/BFb0120959.

[16] R. Garmanjani, D. Júdice, and L. N. Vicente, *Trust-region methods without using derivatives: Worst case complexity and the nonsmooth case*, SIAM Journal on Optimization, 26 (2016), pp. 1987–2011, https://doi.org/10.1137/151005683.

[17] W. Hare, *Compositions of convex functions and fully linear models*, Optimization Letters, (2017), https://doi.org/10.1007/s11590-017-1117-x.

[18] 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, https://doi.org/10.1007/s10589-013-9547-6.

[19] K. C. Kiwiel, *Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM Journal on Optimization, 18 (2007), pp. 379–388, https://doi.org/10.1137/050639673.

[20] K. C. Kiwiel, *A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM Journal on Optimization, 20 (2010), pp. 1983–1994, https://doi.org/10.1137/090748408.

[21] J. Larson, M. Menickelly, and S. M. Wild, *Manifold sampling for L1 nonconvex optimization*, SIAM Journal on Optimization, 26 (2016), pp. 2540–2563, https://doi.org/10.1137/15M1042097.

[22] J. Larson and S. M. Wild, *Censored L1 problem specifications*, 2018. http://www.mcs.anl.gov/~jlarson/MS4PL.

[23] Y. LeCun, *Who is afraid of non-convex loss functions?*, 2007, https://www.cs.nyu.edu/~yann/talks/lecun-20071207-nonconvex.pdf. Presented at NIPS Workshop on Efficient Machine Learning.

[24] P.-L. Loh, *Statistical consistency and asymptotic normality for high-dimensional robust M-estimators*, The Annals of Statistics, 45 (2017), pp. 866–896, https://doi.org/10.1214/16-aos1471.

[25] T. Mitchell, *GRANSO*, 2017. https://gitlab.com/timmitchell/GRANSO.

[26] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572, https://doi.org/10.1137/0904038.

[27] J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM Journal on Optimization, 20 (2009), pp. 172–191, https://doi.org/10.1137/080724083.

[28] A. Neumaier, *MINQ*, 2017. http://www.mat.univie.ac.at/~neum/software/minq.

[29] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Mathematics of Operations Research, 18 (1993), pp. 227–244, https://doi.org/10.1287/moor.18.1.227.

[30] S. Scholtes, *Introduction to Piecewise Differentiable Equations*, Springer New York, 2012, https://doi.org/10.1007/978-1-4614-4340-7.

[31] L. Stella, A. Themelis, and P. Patrinos, *Forward-backward quasi-Newton methods for nonsmooth optimization problems*, Computational Optimization and Applications, 67 (2017), pp. 443–487, https://doi.org/10.1007/s10589-017-9912-y.

[32] S. M. Wild, *Solving derivative-free nonlinear least squares problems with POUNDERS*, in Advances and Trends in Optimization with Engineering Applications, T. Terlaky, M. F. Anjos, and S. Ahmed, eds., SIAM, 2017, pp. 529–540, http://www.mcs.anl.gov/papers/P5120-0414.pdf.

[33] R. S. Womersley, *Censored discrete linear l_1 approximation*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 105–122, https://doi.org/10.1137/0907008.

[34] R. S. Womersley and R. Fletcher, *An algorithm for composite nonsmooth optimization problems*, Journal of Optimization Theory and Applications, 48 (1986), pp. 493–523, https://doi.org/10.1007/bf00940574.

[35] Y.-X. Yuan, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Mathematical Programming, 31 (1985), pp. 220–228, https://doi.org/10.1007/bf02591750.