Optimization and Engineering manuscript No.
(will be inserted by the editor)

Optimization with Hidden Constraints and Embedded Monte Carlo Computations

Xiaojun Chen · C. T. Kelley

the date of receipt and acceptance should be inserted later

Abstract In this paper we explore the convergence properties of deterministic direct search methods when the objective function contains a stochastic or Monte Carlo simulation. We present new results for the case where the objective is only defined on a set with certain minimal regularity properties. We present two numerical examples to illustrate the ideas.

Keywords Sampling Methods, Monte Carlo Simulation, Water Resource Policy, Hidden Constraints

Mathematics Subject Classification (2000) 65K05, 65K10, 90C30

1 Introduction

In this paper we consider the asymptotic convergence behavior of stencil-based direct search methods when applied to problems with "hidden constraints" for which both computation of the objective function and the determination of feasibility depend on an internal stochastic or Monte Carlo (MC) simulation. We refer to such simulations as "embedded" in the function computation. We are motivated by previous work on optimization [11, 32, 43] and nonlinear equations [46–48]. In these applications the objective function or nonlinear residual contained an embedded Monte Carlo simulation which was part, but not all, of the computation. In these problems one could control the variance in the Monte Carlo simulation by increasing the sample size, and we explored one way to manage the sample size as the overall solution progressed in [46] for nonlinear equations.

Hidden constraints are not given by explicit formulae. One detects infeasibility only when the objective function fails to return a value. Our results extend previous work [21, 30] to the case of functions with embedded Monte Carlo simulations. We discuss the example from [11] in § 3 to show how the theoretical results in this paper can be realized in the context of an application. We also know of one example [24] in which a direct search method, in this case Hooke-Jeeves [26], was used to optimize a function which was evaluated by a physical experiment. The theory we develop in this paper would apply to such cases as well.

The idea is simply to increase the sample size in the embedded Monte Carlo simulation as the optimization progresses. We acknowledge that if function evaluations are expensive, such an approach can rapidly become impossible to execute. Even so, the asymptotic results we prove do provide guidance as to how one should increase the number of samples even if one plans to stop the optimization after only a few rounds of increases in the sample sizes. One can envision, but we do not explore this in this paper, other ways to avoid this cost. One could use reduced order models in all or part of the objective function, model the Monte Carlo simulation with a response surface, or use variance reduction methods.

Xiaojun Chen
Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
E-mail: maxjchen@polyu.edu.hk

C. T. Kelley
North Carolina State University Department of Mathematics, Box 8205, Raleigh, NC 27695-8205, USA E-mail: tim_kelley@ncsu.edu

Our work extends previous work on derivative-free methods for objective functions with various types of randomized components. All these papers iteratively refine the quality of the approximation. In [31, 44] the pattern search class of direct search methods [33] is used. The authors of [31] solve the problem with a trust-region formulation of a generating set search (GSS) [33]. Aside from the trust-region formulation and a (slightly) more general search strategy, their approach is the same as ours. They increase the sample size as the GSS algorithm progresses.

Algorithms such as a generalized version of the Nelder-Mead [37] method were applied in [1]. Interpolatory models that extend the UOBYQA [38] have also been used [16, 18] with a Bayesian approach. The DIRECT [28] method, which is not one of the methods under consideration in this paper, has also been applied [17]. In the deterministic case the methods of analysis in § 2 of this paper have been applied to both DIRECT and direct search methods for functions that are not everywhere defined [21].

There are many ways to quantify the randomness in the objective function. Our approach, which we quantify at the end of this section in Assumption 1, says that the standard deviation is proportional to the square root of sample size. Our assumption is similar to the setting of [31] and the example in [44]. Those papers consider a general unconstrained stochastic optimization problem, which can be written as

$$\min_{x \in X} \{ f(x) := E[F(x, \boldsymbol{\xi})] \}. \tag{1}$$

Here $X \subseteq R^N$ is a nonempty compact set, $\boldsymbol{\xi} : \Omega \to \Xi \subseteq R^\ell$ is a random vector whose realizations are denoted by ξ, $F : X \times \Xi \to R$ and $(\Xi, \mathcal{F}, \mathcal{P})$ is the induced probability space.

Suppose that we have independent sample $\xi^1, \ldots, \xi^{N_{MC}}$ of N_{MC} realizations of the random vector $\boldsymbol{\xi}$. The Sample Average Approximation (SAA) replaces the expectation in (1) with the sum

$$\tilde{f}(x, N_{MC}) = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} F(x, \xi^i).$$

Suppose $F(x, \xi)$ is Normal$(f(x), \sigma^2)$. Then we have that $\sqrt{N_{MC}} |\tilde{f}(x, N_{MC}) - f(x)|$ is Normal$(0, \sigma^2)$, and hence the Chebyshev inequality [40] implies that

$$Prob\left(|\tilde{f}(x, N_{MC}) - f(x)| \geq \frac{\sigma k}{\sqrt{N_{MC}}} \right) \leq \frac{1}{k^2} \tag{2}$$

for any $k > 1$.

In the remainder of this section we describe stencil-based search methods in § 1.1, briefly discuss hidden constraints in § 1.2, and then we describe our setting for functions with embedded Monte Carlo simulations and hidden constraints in § 1.3. Our setting is similar to those for unconstrained problems in [31, 44]. We differ from [31] in the algorithmics, even for the unconstrained case.

In § 2 we present our convergence results.

For the unconstrained cases where f is everywhere defined, the asymptotic results state that if the sample size within the MC simulation is increased to infinity as the size of the stencil is decreased to zero, then the sequence of generalized gradients of f at x_n has a subsequence which converges to zero with probability one.

We caution the reader that asymptotic convergence results for direct search methods provide theoretical guidance, but do not precisely predict performance in the situations where the methods are used. In practice, unlike gradient-based methods for smooth problems, one cannot continue the iteration long enough to observe the asymptotic predictions of the theory. The reasons for this include expensive function evaluations, large amounts of noise in the function evaluations, and limits on the degree to which the accuracy in the function can be improved as the iteration progresses.

1.1 Stencil-Based Direct Search Methods

A stencil centered at x with scale h is the set of points $\{x + hv_i\}_{i=1}^K \cup \{x\}$, where $v_i \in V$. V is a set of directions $V = \{v_i\}_{i=1}^K$ which guides the evaluations of the objective function as the optimization progresses. While the stencil directions may vary as the iteration progresses, at least a subset of the stencil is determined by the algorithm and not generated at random. A few random directions are useful for some problems, and we discuss that in § 2 and § 3.

Direct search methods are often used as components of other methods which improve the search by using surrogate models of the function. This strategy uses the surrogate model to explore the optimization landscape in a way that may be independent of the stencil, hoping to get a better point. The stencil-based part of the optimization is used to provide convergence theory and robustness. The example in § 3 uses the implicit filtering algorithm [30], which augments the search with a quasi-Newton method.

We begin by reviewing basic stencil-based direct search methods in the context of the unconstrained optimization problem

$$\min_{x \in R^N} f(x). \tag{3}$$

Stencil-based methods evaluate the objective function on a stencil, scaled by a factor h and centered at the current point. As the iteration progresses, f is evaluated at the points on the stencil and either a new point is taken or the stencil size is reduced. Algorithm **basic_sample** is a simple case.

```
**basic_sample**$(x, f, h, V)$
  **for** forever **do**
    $f_{base} = f(x)$
    $f_{min} = \min_{1 \le i \le K} f(x + hv_i)$
    $x_t = x + hv^*$ where $v^* \in \{v_i \mid f(x + hv_i) = f_{min}\}$
    **if** $f_{min} \ge f_{base}$ **then**
      $h \leftarrow h/2$
    **else**
      $x \leftarrow x_t$
    **end if**
  **end for**
```

Convergence proofs are based on choosing the stencil directions so that if the current point is the best point in the stencil (called *stencil failure* in [29, 30]), then some type of approximate necessary condition holds. Let $V = \{v_i\}_{i=1}^K$ be the stencil directions and let h be the search distance along those directions. Stencil failure at a point x with scale h means that x is the best point in the stencil, *i.e.*

$$f(x) \le f(x + hv_i), \text{ for } i = 1, \dots K. \tag{4}$$

The set of directions V can be fixed throughout the iteration, see, for example, [19, 23, 33, 43]. That strategy works fine for unconstrained problems where f is everywhere defined and smooth, if the set of directions is constructed properly. In this case, the proof that stencil failure implies the approximate first-order necessary conditions for optimality,

$$\|\nabla f(x)\| = O(h),$$

requires that the direction set contains a positive basis [14, 33, 35, 50]. This means that every vector in R^N can be represented as a non-negative linear combination of the directions set.

In the constrained case, one must either add directions near the tangent space of active constraints or, when the constraints are not explicitly given by formulae, enrich the stencil either by adding random directions [30] or using a deterministic scheme to make sure no directions of descent can be missed [4]. We will discuss this point in more detail in § 2.

1.2 Hidden Constraints

We consider problems in this paper in which the objective function may not be everywhere defined. Such problems arise, for example, when internal computations within f fail [10, 12, 15, 43] or when the evaluation of f itself tests for feasibility conditions which are not given by closed form inequalities [11, 32]. We refer to these constraints which are internal to f as "hidden constraints". One typically handles hidden constraints by setting a flag or returning NaN or ∞ when f is called. A second, and very different, case is when explicit constraints, i.e. those given by inequalities, are handled with the extreme barrier approach [3, 4, 34, 36]. This means that one does not attempt to capture tangent directions or even use the functions which define the constraints. Instead one treats the constraints the same as if they were hidden constraints.

We follow the notation of [30] to describe the deterministic case. Here f is defined and Lipschitz continuous on a set $\mathcal{D} \subset R^N$ and we have no information about $\mathcal{D}$ other than f does not return a value when evaluated at $x \notin \mathcal{D}$. For the theoretical results we set $f(x) = \infty$ if $x \notin \mathcal{D}$ so we can obtain the inequalities we need for the necessary conditions. In practice, we use NaN instead of ∞.

Suppose $\mathcal{D}$ is nonempty and closed. The distance function related to $\mathcal{D}$, defined by

$$d_{\mathcal{D}}(x) = \min\{\|x - z\| \mid z \in \mathcal{D}\}$$

is Lipschitz continuous. The Clarke tangent cone $T_{\mathcal{D}}(x)$ to $\mathcal{D}$ at point x in $\mathcal{D}$ is defined as follows [13]:

$$T_{\mathcal{D}}(x) = \{v \in R^N \mid \limsup_{\substack{y \to x \\ \tau \downarrow 0}} \frac{d_{\mathcal{D}}(y + \tau v) - d_{\mathcal{D}}(y)}{\tau} = 0\}.$$

An intrinsic characterization of $T_{\mathcal{D}}(x)$ is that a vector $v \in T_{\mathcal{D}}(x)$ for $x \in \mathcal{D}$ if and only if for any sequences $y^k \in \mathcal{D}, y^k \to x$ and $\tau^k > 0, \tau^k \to 0$ as $k \to \infty$, there is a sequence $v^k \to v$ such that $y^k + \tau^k v^k \in \mathcal{D}$ for all $k \geq 0$.

The contingent cone $T^c_{\mathcal{D}}(x)$ (called the tangent cone in [39] is the set of vectors v such that there are sequences $y^k \in \mathcal{D}, y^k \to x$ and $\tau^k > 0, \tau^k \to 0$ as $k \to \infty$ such that

$$v = \lim_{k \to \infty} \frac{y^k - x}{\tau^k}.$$

Obviously $T_{\mathcal{D}}(x) \subseteq T^c_{\mathcal{D}}(x)$. When $T_{\mathcal{D}}(x) = T^c_{\mathcal{D}}(x)$, we say $\mathcal{D}$ is regular at x. Moreover, we say a set $\mathcal{D}$ is regular if $\mathcal{D}$ is regular at every point $x \in \mathcal{D}$.

We illustrate the idea with some examples. If a set $\mathcal{C}$ is convex, then it is regular at every point x and

$$T_{\mathcal{C}}(x) = cl\{\tau(y - x) \mid \tau \geq 0, y \in \mathcal{C}\}.$$

The set

$$\mathcal{C}_1 = \{(x_1, x_2)^T \in R^2 \mid x_2 \leq \sqrt{|x_1|}\}$$

is not regular at $x = (0, 0)^T$ because

$$T_{\mathcal{C}_1}(x) = \{(0, \alpha)^T \in R^2 \mid \alpha \in R\} \neq T^c_{\mathcal{C}_1}(x) = R^2.$$

The set

$$\mathcal{C}_2 = \{(x_1, x_2)^T \in R^2 \mid x_2 \geq -|x_1|\}$$

is not regular at $x = (0, 0)^T$ because

$$T_{\mathcal{C}_2}(x) = \{(x_1, x_2)^T \in R^2 \mid x_2 \geq |x_1|\} \neq T^c_{\mathcal{C}_2}(x) = \mathcal{C}_2.$$

Our objective for a direct search method is to asymptotically satisfy the necessary conditions [13, 27]

$$f^\circ(x^*; v) \geq 0 \text{ for all } v \in T_{\mathcal{D}}(x^*), \tag{5}$$

for $x^* \in \mathcal{D}$ being a local minimizer of

$$\min_{x \in \mathcal{D}} f(x), \tag{6}$$

where the Clarke-Jahn [13, 27] directional derivative of f at a point $x \in \mathcal{D}$ in the direction v is

$$f^\circ(x; v) = \limsup_{\substack{y \to x,\ y \in \mathcal{D} \\ t \downarrow 0,\ y + tv \in \mathcal{D}}} \frac{f(y + tv) - f(y)}{t}.$$

We point out that if f is Lipschitz continuously differentiable and $\mathcal{D}$ is determined by smooth inequality constraints then (5) is equivalent to the standard first order necessary conditions for inequality constrained optimization problems [13].

1.3 Embedded Monte-Carlo Functions within the Objective Function

The central question in this paper is how one should interpret stencil failure when the evaluation of the objective function depends on a MC simulation. We will use notation similar to that in [46, 48]. We will approximate the evaluation of an objective function f with a randomized simulation using a variable sample size N_{MC} and let $\tilde{f}(x, N_{MC})$ be the outcome of the simulation for f with sample size N_{MC}. It is important to note that the results of the MC simulation may not be reproducible. We have encountered such problems in some previous work [11, 23, 32, 43], but used deterministic theory for guidance and did not deal with the MC nature of the objective function at all.

We now state our assumptions on $\tilde{f}$. The assumption is motivated by the discussion earlier on the problem (1), but extends the role of the Chebyshev inequality to cover the hidden constraints.

Assumption 1 *There are functions $\phi, c_F : (0, \infty) \to (0, \infty)$ with $\lim_{z\to\infty} \phi(z) = 0$, such that for all $x \in \mathcal{D}$ and $\delta > 0$*

$$Prob\left(|f(x) - \tilde{f}(x, N_{MC})| > \frac{c_F(\delta)}{\sqrt{N_{MC}}} \right) \leq \delta, \tag{7}$$

and

$$Prob\left(\tilde{f}(x, N_{MC}) = \infty \right) \leq \phi(N_{MC}). \tag{8}$$

For all $x \notin \mathcal{D}$,

$$Prob\left(\tilde{f}(x, N_{MC}) < \infty \right) \leq \phi(N_{MC}). \tag{9}$$

Assumption 1 has two parts. We assume, as we did for nonlinear equations in [46], that one can use a Chebyshev type inequality to estimate the error in f. This estimate for error in the function, (7), is equivalent to (2), with $\delta = 1/k^2$ and $c_F = \sigma k = \sigma/\sqrt{\delta}$. Hence problem (1) represents a class of examples for (7).

We must also handle hidden constraints, which is the second part of the assumption. In the case considered here, one determines that a point $x \notin \mathcal{D}$ by a failure in f, *i.e.* $f(x) = \infty$. Equations (8) and (9) say that the probability that the evaluation of f does not correctly reflect feasibility is small and model that with a condition similar to (7).

2 Convergence Theory

One cannot keep the stencil directions constant in this case without running the risk that the iteration will terminate at a point which does not satisfy the necessary conditions (5) (see Example 4.2 in [30]). The resolution of this is to use a sequence of stencils with an asymptotically dense set of directions [4, 21, 30, 45] and only use the sample points from $\mathcal{D}$ (i.e. those for which f returns a value) in the search or poll phases.

We will follow [42, 49] and use the abbreviation "w.p.1" for "with probability one".

We will let the direction set for iteration n be

$$V_n = \{v_i^n\}_{i=1}^{K_n}.$$

Let $\mathcal{V} = \{V_n\}$ be the sequence of direction sets. We say that $\mathcal{V}$ is **rich** if for any unit vector $v \in R^N$ and any subsequence $\mathcal{W} = \{W_{n_j}\}$ of $\mathcal{V}$

$$\liminf_{j\to\infty} \min_{w\in W_{n_j}} \|w - v\| = 0 \text{ w.p.1.} \tag{10}$$

The MADS approach [4] generates a rich set of directions with either a deterministic algorithm or randomly, and (10) is a consequence of the direction generation algorithm. Another way [5, 21], which we use in § 3, is to augment a fixed stencil with one or more random directions. In this latter case, the sequence of stencils will be rich w.p.1.

In [21, 30] we extended the **basic_sample** algorithm by selecting a new stencil from a rich set $\mathcal{V}$ of directions at each iterate. The analysis set $f(x) = \infty$ for $x \notin \mathcal{D}$ and required that the initial iterate be in $\mathcal{D}$. Our generalization will require that as well. The convergence result stated that any limit point of the subsequence of the iterates at which stencil failure occurred satisfied (5).

Our new algorithm is **general_mc**. Here we must let N_{MC} depend on h in a nontrivial way. The reason for this is that we wish to prove a first-order necessary condition (5) and so the errors or variances in the function evaluation must decrease faster than h. Algorithm `general_mc` is a formal description of the iteration.

```
**general_mc**($x, f, h, \mathcal{V}, N_{MC}$)
  **for** forever **do**
    $\tilde{f}_{base} = \tilde{f}(x, N_{MC})$;
    $\tilde{f}_{min} = \min_{1 \le i \le K} \tilde{f}(x + hv_i, N_{MC})$
    $x_t = x + hv^*$, where $v^* \in \{v_i \mid \tilde{f}_{min} = \tilde{f}(x + hv_i, N_{MC})\}$
    **if** $\tilde{f}_{min} \ge \tilde{f}_{base}$ **then**
      $h \leftarrow h/2$; $N_{MC} \leftarrow N_{MC}(h)$.
    **else**
      $x = x_t$
    **end if**
  **end for**
```

We will call the subsequence of iterates $\{x_m\}$ that triggers stencil failure *major iterates.* Note that the scale counter m increments only when the scale h changes after a stencil failure. After increasing m one changes the sample size N_{MC} and the scale h. In the fully deterministic case, all iterates are feasible. In the MC case, however, we would need both

$$\tilde{f}(x_m, N_{MC}^m) < \infty \tag{11}$$

and

$$\tilde{f}(x_m, N_{MC}^{m+1}) < \infty. \tag{12}$$

Equation (11) follows from the feasibility of the major iterates before index m and (12) is needed to guarantee the feasibility of major iterate $m+1$. For the theory, we will assume that either (12) holds or the poll finds a better point on the stencil. This assumption will guarantee that the iteration following x_m is feasible w.p.1.

Unlike the deterministic case, the iteration could break down in the following way. One could have $\tilde{f}(x_n, N_{MC}^n) < \infty$ but have the function evaluation fail at every point of the stencil on the subsequent iteration. In this event, the iteration would terminate with failure after finitely many calls to $\tilde{f}$. We must assume that the iteration does not break down in order to even state an asymptotic convergence result. While the assumption that stencil failure occurs infinitely often implies that the iteration is infinite, and hence does not break down, we will state that explicitly in the statement of Theorem 1.

As we said in the introduction, the asymptotic convergence result requires that the number of samples in the Monte Carlo simulation increase as the optimization progresses. We can now make this precise by expressing N_{MC} as a function of the scale h. We will need

$$\lim_{h \to 0} (h\sqrt{N_{MC}(h)})^{-1} = 0. \tag{13}$$

As we said in the introduction, increasing the number of samples rapidly enough to satisfy (13) is problematic if function evaluations are expensive. Even so, Theorem 1 provides useful guidance for increasing N_{MC} even in the early parts of an iteration, where a small value could be useful when h is large and the stencil only gives coarse-grained information, but more samples would be needed as h is reduced and more accuracy in $\tilde{f}$ is needed to resolve differences of function values on smaller stencils.

Theorem 1 *Let Assumption 1 hold and assume that the iteration does not break down. Let $\mathcal{V}$ be a rich sequence of stencils and let f be defined and Lipschitz continuous on $\mathcal{D}$. Assume that the set $\mathcal{D}$ is regular. Let $\{x_m\}$, $\{h_m\}$, $\{N_{MC}^m = N_{MC}(h_m)\}$, be the subsequences of iterates, scales, and numbers of samples generated by* **general_mc** *algorithm at stencil failure.*

Assume that stencil failure occurs infinitely often, that (12) *holds infinitely often and that* (13) *holds. Then every accumulation point of the sequence $\{x_m\}$ satisfies* (5) w.p.1.

Proof Note that $\{x_m\}$ is a subsequence of the entire iteration. At these "major iterates" several things happen:

– The scale changes, so $h_{m+1} = h_m/2$.
– The sample size N^m_{MC} changes; $N^{m+1}_{MC} = N_{MC}(h_{m+1})$.
– $\tilde{f}(x_m, N^m_{MC}) < \infty$.

The last item follows from our assumptions that guarantee that all iterates, in particular major iterates, are feasible w.p.1.

Let $x^* \in \mathcal{D}$ be any limit point of $\{x_m\}$ and let $v \in T_{\mathcal{D}}(x^*)$ be such that $x^* + tv \in \mathcal{D}$ for t sufficiently small. Since $\mathcal{V}$ is rich we can, by taking a subsequence of the sequence of major iterates, find a sequence of directions $\{v_m\}$ such that $v_m \in V^m$ for all major iterates m and $v_m \to v$. We will denote this subsequence of major iterates with the index m as well.

Lipschitz continuity of f and convergence of v_m to v and $h_m \to 0$ imply that

$$f(x_m + h_m v_m) - f(x_m + h_m v) = O(h_m \|v - v_m\|) = o(h_m).$$

Therefore, the definition of the generalized directional derivative and the convergence of x_m to x^* imply that

$$f^o(x^*; v) \geq \limsup_{m \to \infty} \frac{f(x_m + h_m v_m) - f(x_m)}{h_m}. \tag{14}$$

We will be done if we can show that

$$\varDelta_m \equiv f(x_m + h_m v_m) - f(x_m) \geq o(h_m) \qquad \text{w.p.1.} \tag{15}$$

which, together with (14), will imply that

$$f^o(x^*; v) \geq 0, \qquad \text{w.p.1.}$$

as asserted.

We will consider two cases. If $x_m + h_m v_m \in \mathcal{D}$ infinitely often, we may refine the subsequence and have $x_m, x_m + h_m v_m \in \mathcal{D}$ for all m. We will compare $\varDelta_m$ to

$$\tilde{\varDelta}_m = \tilde{f}(x_m + h_m v_m, N^m_{MC}) - \tilde{f}(x_m, N^m_{MC}).$$

Stencil failure implies that $\tilde{\varDelta}_m \geq 0$.

We will now show that

$$\varDelta_m - \tilde{\varDelta}_m = O(1/\sqrt{N^m_{MC}}) = o(h_m) \tag{16}$$

infinitely often w.p.1. Fix $\delta \in (0, 1/2)$ in Assumption 1. Then, for each m, Assumption 1 says that

$$Prob\left(|f(x_m) - \tilde{f}(x_m, N^m_{MC})| > \frac{c_F(\delta)}{\sqrt{N^m_{MC}}} \right) < \delta$$

and

$$Prob\left(|f(x_m + h_m v_m) - \tilde{f}(x_m + h_m v_m, N^m_{MC})| > \frac{c_F(\delta)}{\sqrt{N^m_{MC}}} \right) < \delta.$$

Therefore, for each m,

$$Prob\left(|\varDelta_m - \tilde{\varDelta}_m| > \frac{2c_F}{\sqrt{N^m_{MC}}} \right) \leq 2\delta. \tag{17}$$

Since $\delta \in (0, 1/2)$, the probability that (16) fails infinitely often is zero. Hence, we may refine the sequence further if necessary and complete the proof in this case.

In the second case, $x_m + h_m v_m \in \mathcal{D}$ for only finitely many m. We may then refine the subsequence, if necessary, and assume that $x_m + h_m v_m \notin \mathcal{D}$ for all m. Since x_m is a major iterate, $\tilde{f}(x_m, N^m_{MC})$ is finite. Therefore Assumption 1 implies that the probability that $x_m \notin \mathcal{D}$ for any given m is $\leq \phi(N^m_{MC})$. Hence the probability that $x_m \notin \mathcal{D}$ infinitely often is zero.

Since $x_m \in \mathcal{D}$ for infinitely many m w.p.1 and $x_m + h_m v_m \notin \mathcal{D}$, we have

$$f^o(x^*; v) = +\infty \qquad \text{w.p.1.}$$

which completes the proof.

3 Implementation and Examples

In this section we illustrate the theory with two examples. Each example is solved with the **`imfil.m`** implementation of implicit filtering [30]. We compare the optimization histories with varying $N_{MC}(h)$ to those with N_{MC} fixed at the maximum of $N_{MC}(h)$ for each case.

3.1 Implicit Filtering Implementation

Implementation of stencil-based direct search methods generally do more than evaluate the function on the stencil. Often a second algorithm is added to the loop. In this part of the optimization one may, for example, create a surrogate model of f with the function values at the points on the stencil [2, 6–8, 30, 33], and use that to attempt to find a better point. If this second phase finds a better point, then one accepts that new point and proceeds with the optimization. There are several ways to do this, and we focus on the implicit filtering method in this example. Implicit filtering was invented to solve a problem [23, 43] which had both an embedded Monte Carlo algorithm in the objective function and hidden constraints.

It is important to note that our surrogate model is for f, not for the embedded Monte Carlo simulation. The use of a surrogate model in this way does not change the the theory, because decisions to terminate the iteration or reduce h are based on stencil failure.

As we said in § 1, the theoretical asymptotic results on direct search methods provide guidance into the design of algorithms and software. In practice, one cannot continue the iteration long enough to observe asymptotic behavior. Examples like this one are intended to illustrate how the theory can advise the solvers and how one can use existing software to incorporate the ideas from the theoretical results. In particular, we can now significantly increase the sample size for the example problem in this section and obtain results in a matter of hours, as opposed to days that it would take if we use only the larger sample size.

We used the `imfil.m` code from [30]. This program uses a quasi-Newton method with approximate gradients based on the feasible stencil points for the search phase. `imfil.m` allows for stencils with random directions. For this paper we used the centered difference stencil and additional random directions. When a search point is infeasible for either the bound constraints or the hidden constraints, the value of the objective function is set to NaN and not used in the gradient approximation.

We also "cached" the function evaluations. This means that we do not reevaluate $\tilde{f}$ at a point after a change of scale. Doing this helps avoid a break down in the iteration, which we have sometimes observed in practice for this problem.

We implement Algorithm `general_mc` using features of `imfil.m` which allow for two-way communication with the objective function. This allows N_{MC} to depend on h as it does in the formal description of the algorithm in § 2.

The `imfil.m` code is designed for bound constrained problems and uses the bounds to scale the variables to be in $[0, 1]$. Implicit filtering uses a projected BFGS (Broyden, Fletcher, Goldfarb, Shanno) [9, 22, 25, 41] iteration for the search step. The theory for the poll step in § 2 applies with little change to the bound constrained case. This is exactly the same as for the deterministic case [30].

After the scaling is done, the scales in the search are $h = 1/2, 1/4, \dots, 2^{-8}$. In each example we compare two scenarios. The first holds N_{MC} constant throughout the iteration at $N_{MC} = N_{Base}+196608$. The second varies N_{MC} using the formula

$$N_{MC}(h) = N_{Base} + \text{floor}(h^{-2} \max(.1, \log_2(\log_2(1/h)))) \tag{18}$$

which satisfies (13) because

$$(h\sqrt{N_{MC}(h)})^{-1} = O((\log_2(\log_2(1/h)))^{-1/2}).$$

Here N_{Base} is problem dependent and selected to be small, but still large enough to capture the important coarse-grained properties of f.

The formula (18) is *ad hoc*. For $h = 1/2, 1/4, \dots, 2^{-8}$ we have

$$N_{MC} = N_{Base} + (0, 16, 64, 512, 2048, 8192, 31868, 196608).$$

We would expect that a formula similar to (18) would work for other problems, especially given the limitations of any asymptotic theory that we mentioned in the introduction.

We experimented with other ways to change N_{MC} as $h \to 0$. We considered formulae of the form

$$N_{MC}(h) = N_{Base} + \text{floor}(C_H \theta(h) h^{-2}) \text{ for } h < 1/2.$$

Values of $C_H \in [25, 50, 100, 200]$ and functions

$$\theta(h) = \max(1, \log_2(\log_2(1/h))) \text{ and } \theta(h) = \log_2(1/h),$$

produced essentially the same results as those we report in this section. One could increase the initial value of N_{Base}, but that would make the cost of computing $\tilde{f}$ in the early phase of the optimization far too large.

3.2 A Simple Example

We illustrate the theory first with an artificial example which is a modification from one in [30]. The problem is to minimize

$$f_{exact}(x) = (1/2 - x_1)^2 + (1 - x_1)^2(1 - x_2)^2/4$$
$$+(1/2 - x_1)^2(1 + x_2 - 2x_2^2)/10, \tag{19}$$

subject to the bound constraints

$$0 \le x_1 \le 1, 0 \le x_2 \le 1,$$

and the linear constraint

$$x_1 + x_2 \ge 1. \tag{20}$$

We use the extreme barrier approach to handle the linear constraints. The minimizer is $x^* = (.5, 1)^T$. The initial iterate in this example is $x_0 = (.75, .75)$.

In the example here we perturb both f and the constraints by a normal random variable with mean 0 and variance $1/\sqrt{N_{MC}}$. The objective function is

$$f(x) = \begin{cases} \text{NaN}, & \text{if } x_1 + x_1 < 1 + \xi \\ f_{exact}(x)(1 + \xi), & \text{otherwiser} \end{cases}$$

where ξ is sampled from $\mathcal{N}(0, 1/\sqrt{N_{MC}})$ every time $f(x)$ is evaluated.

In this example we limited each run to a budget of 100 function evaluations and terminated the iteration when either the budget was exceeded, the scales had been exhausted, or the objective function value fell below 10^{-6} at the end of an iteration. $N_{MC}(h)$ is given by (18) with $N_{Base} = 100$. We use one random direction in addition to the central difference stencil in this example.

We perform 40 runs of the optimization for each scenario. We tabulate cost on the horizontal axis in terms of

$$S = \sum_{h=2^{-p}, 1 \le p \le 8} \text{function evaluations } \times N_{MC}(h),$$

as the iteration progresses. Figure 1 shows that the performance in terms of reduction in the function value as the iteration progresses is roughly the same for each case, but the variable N_{MC} optimizations cost substantially less.

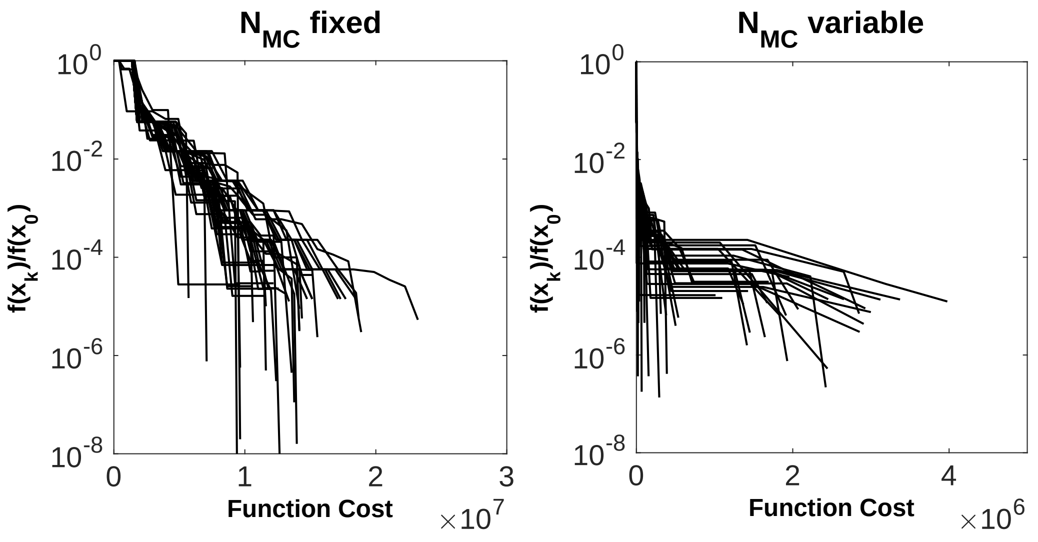

Fig. 1 Value of $\tilde{f}$ vs cumulative objective function cost: 40 runs.

3.3 Water Resources Example

The example here is far noisier than the one in § 3.2. The function is more expensive to evaluate and there are multiple local minima. These complications make it more difficult to evaluate the final results, since the converged results are not the same for all of the optimizations. However, as we will see below, the performance with varying N_{MC} was no worse that that with N_{MC} set to the maximum of $N_{MC}(h)$.

We consider an example from [11, 20, 30, 32] on water resources policy. We refer the reader to those references for the details of the application and focus here only on the properties of the problem which are relevant to this paper. The objective function depends on six continuous variables which determine policy decisions to by or sell water rights or options to water rights. Within the objective function is a water supply model which uses rainfall information to determine the cost of the water supply policy and whether the policy fails to deliver enough water. The MC simulation provides the rainfall information to the water supply model by sampling from historical data. Hence if there are N_{MC} samples, then the call to the objective function will run the water supply model N_{MC} times corresponding to the samples from historical data. The water supply model results are also averaged to determine the probability of failure of the policy and the conditional value at risk (CVaR). We put limits on the failure probability and CVaR which are the hidden constraints. The hidden constraints are tested after the N_{MC} water supply model runs are complete.

Unlike an MC simulation based on high-dimensional integration, for example, we cannot prove that (7) holds for this example, but we do believe that (7) is a reasonable assumption. We point out that we used a sample size of at most 500 in our previous work. In this paper we increase that to $197,708$. Here $N_{MC}(h)$ is given by (18) with $N_{Base} = 500$. We used twelve random directions in addition to the central difference stencil in this example.

Figure 2 illustrates the effects of increasing N_{MC}. The optimization landscapes are typical of one with hidden constraints. Here we plot $\tilde{f}$ as a function of two of the variables with the others held fixed. MATLAB simply plots NaNs as missing values, hence the gaps in the landscape. Note that the plot for $N_{MC} = 200$ has a rough hidden constraint boundary and that $\mathcal{D}$ is not simply connected. One can also see that the landscape is rougher than that for the larger values of N_{MC}. As one increases N_{MC}, the hidden constraint boundary becomes smoother and $\mathcal{D}$ becomes simply connected.

In Figure 3 we plot iteration histories from 12 optimization runs. The initial iterate for all the optimizations was

$$x_0 = (40000, 10000, 1.1, 1.3, .85, 1.10)^T.$$

As in the previous section, we plot the value of $\tilde{f}$ against the cost N_{MC} of the objective function evaluation. In this particular example, a failed call to the objective function costs the same as one that returns a value. The changes in scale h and sample size N_{MC} can be seen by the kinks in the plots. The changes in scale were triggered by stencil failure. The quasi-Newton method, BFGS [9, 22, 25, 41] in this case, was responsible for most of the decrease in the early part of the iteration. The poll took over in the latter part of the iteration, with the BFGS step failing to provide any decrease.

One can see that the histories follow the same general track, albeit with some variance. The solutions are within the range reported in [30]. As we mentioned in the introduction, the asymptotic theory does not precisely predict the history of the iteration, but it does provide guidance and help interpret the results. As was the case in the previous example, the quality of the results for the two cases is roughly the same, but the cost was much less if one varied N_{MC}.

The computations in this section were done on an IBM Blade Server at North Carolina State University High Performance Computing Services.

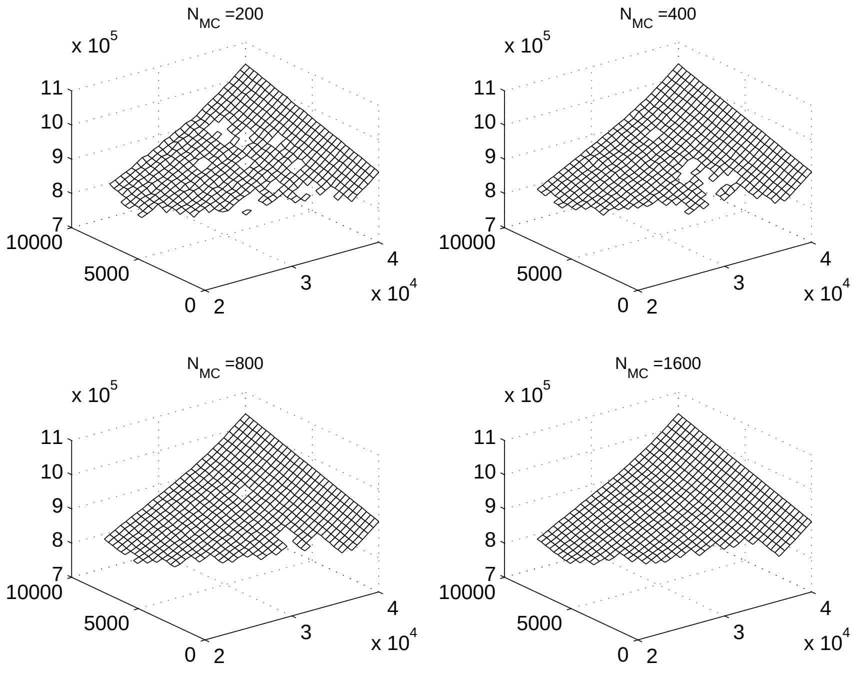

Fig. 2 $\tilde{f}$ becomes smoother as N_{MC} increases.

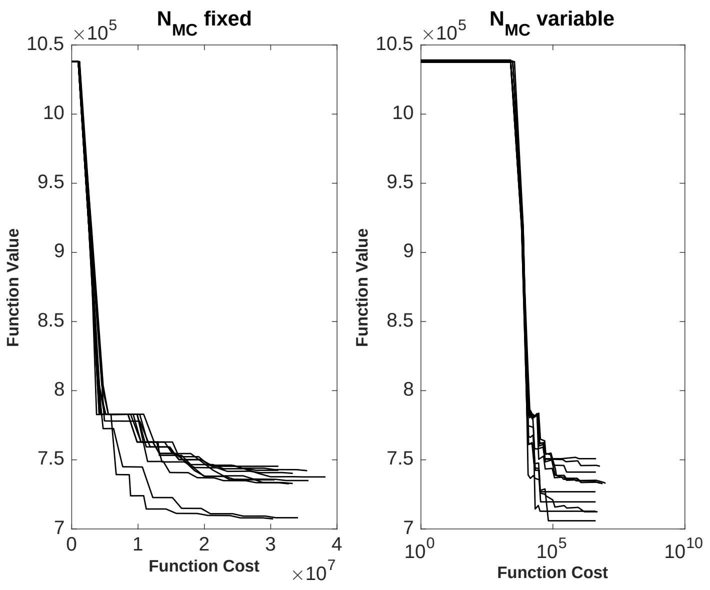

Fig. 3 Value of $\tilde{f}$ vs objective function cost: 12 runs.

Acknowledgements The work of C. T. Kelley was partially supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725, and National Science Foundation Grants DMS-1406349 and CDI-0941253. The work of Xiaojun Chen was partially supported by Hong Kong Research Grant Council grant PolyU5003/11p.

References

1. E. J. Anderson and M. C. Ferris. A direct search algorithm for optimization with noisy function evaluations. *SIAM J. Optim.*, 11(3):837–857, 2001.
2. C. Audet, A. J. Booker, J. E. Dennis, D. W. Moore, and P. D. Frank. A surrogate-model-based method for constrained optimization, AIAA-2000-4891. In *Proceedings of the Symposium on Multidisciplinary Analysis and Optimization*, 2000.
3. C. Audet and J. E. Dennis. Analysis of generalized pattern searches. *SIAM J. Optim.*, 13:889–903, 2003.
4. C. Audet and J. E. Dennis. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17:188–217, 2006.
5. C. Audet and J. E. Dennis. A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.*, 20:445–472, 2009.
6. A. J. Booker. DOE for computer output. Technical Report BCSTECH-94-052, Boeing Computer Services, Seattle, WA, 1994.
7. A. J. Booker. Well–conditioned kriging models for optimization of computer models. Technical Report M&CT-TECH-002, Boeing Phantom Works, Mathematics and Computing Technology, 2000.
8. A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17:1–13, 1999.
9. C. G. Broyden. A new double-rank minimization algorithm. *AMS Notices*, 16:670, 1969.
10. R. G. Carter, J. M. Gablonsky, A. Patrick, C. T. Kelley, and O. J. Eslinger. Algorithms for noisy problems in gas transmission pipeline optimization. *Optim. Engin.*, 2:139–157, 2001.
11. G. W. Characklis, B. R. Kirsch, J. Ramsey, K. E. M. Dillard, and C. T. Kelley. Developing portfolios of water supply transfers. *Water Resources Research*, 42:W05403–1 – W05403–14, 2006.
12. T. D. Choi, O. J. Eslinger, C. T. Kelley, J. W. David, and M. Etheridge. Optimization of automotive valve train components with implicit filtering. *Optim. Engin.*, 1:9–27, 2000.
13. F. H. Clarke. *Optimization and Nonsmooth Analysis.* Number 5 in Classics in Applied Mathematics. SIAM, Philadelphia, 1990.
14. A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization.* MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.
15. J. W. David, C. T. Kelley, and C. Y. Cheng. Use of an implicit filtering algorithm for mechanical system parameter identification, 1996. SAE Paper 960358, 1996 SAE International Congress and Exposition Conference Proceedings, Modeling of CI and SI Engines, pp. 189–194, Society of Automotive Engineers, Washington, DC.
16. G. Deng and M. C. Ferris. Adaptation of the UOBYQA algorithm for noisy functions. In L. P. Perrone, B Lawson, J Liu, and F Wieland, editors, *2007 Winter Simulation Conference*, pages 312–219, 2006.
17. G. Deng and M. C. Ferris. Extension of the DIRECT optimization algorithm for noisy functions. In B Biller, S Henderson, M Hsieh, and J Shortle, editors, *2007 Winter Simulation Conference*, pages 497–504. IEEE, 2007.
18. G. Deng and M. C. Ferris. Variable-number sample-path optimization. *Math. Prog.*, 117:81–109, 2009.
19. J. E. Dennis and V. Torczon. Direct search methods on parallel machines. *SIAM J. Optim.*, 1:448 – 474, 1991.
20. K. E. M. Dillard. *An application of implicit filtering to water resources management.* PhD thesis, North Carolina State University, Raleigh, North Carolina, 2007.
21. D. E. Finkel and C. T. Kelley. Convergence analysis of sampling methods for perturbed Lipschitz functions. *Pacific J. Opt.*, 5:339–350, 2009.
22. R. Fletcher. A new approach to variable metric methods. *Comput. J.*, 13:317–322, 1970.

23. P. Gilmore and C. T. Kelley. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM J. Optim.*, 5:269–285, 1995.
24. P. A. Gilmore, S. S. Berger, R. F. Burr, and J. A. Burns. Automated optimization techniques for phase change piezoelectric ink jet performance enhancement. In *1997 International Conference on Digital Printing Technologies*, pages 716–721. Society for Imaging Science and Technology, IS&T's NIP 13, November, 1997.
25. D. Goldfarb. A family of variable metric methods derived by variational means. *Math. Comp.*, 24: 23–26, 1970.
26. R. Hooke and T. A. Jeeves. 'Direct search' solution of numerical and statistical problems. *JACM*, 8:212–229, 1961.
27. J. Jhan. *Introduction to the theory of nonlinear optimization.* Springer Verlag, Berlin, 1996.
28. D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *J. Opt. Th. Appl.*, 79:157–181, 1993.
29. C. T. Kelley. *Iterative Methods for Optimization.* Number 18 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 1999.
30. C. T. Kelley. *Implicit Filtering.* Number 23 in Software Environments and Tools. SIAM, Philadelphia, 2011.
31. S. Kim and D. Zhang. Convergence properties of direct search methods for stochastic optimization. *Proceedings of the 2009 Winter Simulation Conference*, pages 1003–1011, December 2010. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5679089`.
32. B. R. Kirsch, G. W. Characklis, K. E. M. Dillard, and C. T. Kelley. More efficient optimization of long-term water supply portfolios. *Water Resources Research*, 45:W03414–1 – W03414–12, 2009.
33. T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45:385–482, 2003.
34. T. G. Kolda, R. M. Lewis, and V. Torczon. Stationarity results for generating set search for linearly constrained optimization. *SIAM J. Optim.*, 17:943–968, 2006.
35. R. M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. Technical Report 96-71, Institute for Computer Applications in Science and Engineering, December 1996.
36. R. M. Lewis and V. Torczon. Pattern search algorithms for linearly constrained minimization. *SIAM J. Optim.*, 10:917–941, 2000.
37. J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, 7:308–313, 1965.
38. M. J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Math. Prog. ser B*, 92:555–582, 2002.
39. R. T. Rockafellar and J. B. Wets. *Variational Analysis.* Springer, Berlin, 1998.
40. S. M. Ross. *Introduction to Probability Models.* Academic Press, New York, ninth edition, 2007.
41. D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24: 647–657, 1970.
42. A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming.* MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.
43. D. E. Stoneking, G. L. Bilbro, P. Gilmore, R. J. Trew, and C. T. Kelley. Yield optimization using a GaAs process simulator coupled to a physical device model. *IEEE Trans Microwave Th and Tech*, 40:1353–1363, 1992.
44. M. W. Trosset. On the use of direct search methods for stochastic optimization. Technical report, Rice University, Department of Computational and Applied Mathematics, 2000.
45. L. N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. *Math. Prog.*, 133(1-2):299–325, 2012.
46. J. Willert, X. Chen, and C. T. Kelley. Newton's method for Monte Carlo-based residuals, 2015. to appear in SIAM J. Numer. Anal.
47. J. Willert, C. T. Kelley, D. A. Knoll, and H. K. Park. A hybrid approach to the neutron transport k-eigenvalue problem using NDA-based algorithms. In *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering*, pages 1934–1941, 2013.
48. J. Willert, C. T. Kelley, D. A. Knoll, and H. K. Park. Hybrid deterministic/Monte Carlo neutronics. *SIAM J. Sci. Comput.*, 35:S62–S83, 2013.
49. D. Williams. *Probability with Martingales.* Cambridge University Press, Cambridge, 1991.

50. W. Yu. Positive basis and a class of direct search techniques. *Scientia Sinica, Special Issue of Mathematics*, 1:53–67, 1979.