Non-monotone direct-search methods for deterministic and stochastic derivative-free optimization

A. Ding* Trang H. Tran † L. N. Vicente‡

September 10, 2026

Abstract

In derivative-free optimization (DFO), one minimizes functions for which the gradient is unavailable or expensive to compute. In many applications, objective function values and gradients are noisy due to simulations or system randomness. A class of standard direct-search methods for DFO accept a trial point when it decreases the objective function by an amount proportional to the squared stepsize. However, when applied to complex landscapes, such a requirement may trap the algorithm in a neighborhood of sub-optimal solutions. We study a non-monotone direct-search alternative where the trial function value is compared with the largest objective function obtained through the M most recent distinct iterates. This max-M non-monotone condition permits temporary increases in the objective function and can help navigate narrow curved valleys; however, its theoretical analysis is significantly more challenging due to the lack of monotonic decrease. In this paper, we develop a comprehensive complexity theory for the max-M non-monotone direct-search in both deterministic and stochastic DFO problems. For deterministic objectives, we establish a worst-case iteration bound for a complete poll based on a positive spanning set and an expected iteration bound for a probabilistic-descent poll. We then analyze a stochastic variant using independent function estimates and show the expected iteration complexity under tail-bound assumptions of the stochastic errors. All three results have the standard complexity of $\mathcal{O}(\epsilon^{-2})$, which matches the iteration complexity of monotone direct-search methods. Our theory is enabled by a new family of merit functions that correct the stored objective values by ordered multiples of the squared stepsize, together with a renewal-reward stopping-time argument for the probabilistic methods. Numerical experiments on CUTEst problems indicate that when comparing with monotone direct search, the max-M non-monotone method is particularly helpful on problems exhibiting negative curvature.

1 Introduction

In this paper, we consider unconstrained optimization problems of the form

$$\min_{x\in\mathbb{R}^n} f(x),$$

*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA (and523@lehigh.edu).

†Department of Mathematics, Emory University, Atlanta, GA 30322-1005, USA (htran31@emory.edu). This work was done when the author was a postdoctoral researcher at Lehigh University.

‡Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA (lnv@lehigh.edu).

whose derivatives are assumed to be unavailable and costly to approximate. Besides, the objective function f is assumed to be L-smooth and bounded below by f^*. We will first consider the case in which optimization algorithms have access to a zeroth-order oracle $f(x)$. Then, we will consider the case in which optimization algorithms have access to a stochastic zeroth-order oracle $F(x, \xi)$, which provides noisy observations of its deterministic counterpart $f(x)$ (and where ξ denotes a random estimator). The absence of derivative information is a common feature in many simulation-based optimization problems and constitutes the primary focus of derivative-free optimization (DFO). In this setting, access to the objective function f is limited to evaluations via a zeroth-order oracle, which are often computationally expensive. Consequently, the central objective of DFO is to obtain high-quality solutions with few such oracle queries.

Trust-region, direct-search, and line-search methods constitute three major classes of algorithms in DFO. Trust-region methods construct local surrogate models of f, which are then used to generate trial steps and determine their acceptance. In contrast, direct-search methods probe the search space using a set of directions or polling points without relying on explicit models. Bridging these two paradigms, line-search methods approximate descent directions, typically via finite differences, and perform searches along these directions. For a comprehensive overview of these classes of DFO methods, we refer the reader to, e.g., [1, 9, 24]. A representative example within direct-search methods is the probabilistic descent approach proposed in [20]. Unlike other classical direct-search methods that rely on a positive spanning set (see, e.g., [9]) to guarantee a descent direction (requiring at least $n + 1$ function evaluations), this method incorporates randomness to obtain a descent direction with a sufficiently large probability using only one or two function evaluations. It is shown in [20] that, for deterministic objective functions, the method achieves a zeroth-order oracle complexity of $\mathcal{O}(n/\epsilon^2)$ to find an ϵ-approximate first-order stationary point x defined by $\|\nabla f(x)\| \leq \epsilon$.

It is recognized in [8, 11, 21] that enforcing the monotonicity of the function values may considerably slow the convergence of optimization algorithms, particularly in the presence of narrow curved valleys. To alleviate this phenomenon, a non-monotone technique was proposed in [21] for line-search algorithms, with reported favorable numerical results. Roughly speaking, such a non-monotone line search relaxes the Armijo condition (see, e.g., [27]) and allows some non-monotonicity by using the maximum of objective function values at $M+1$ past and current iterate points, as follows

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq M} f(x_{k-j}) + \frac{1}{2}\alpha_k \nabla f(x_k)^\top d_k,$$

where d_k is a descent direction, $\nabla f(x_k)^\top d_k < 0$, and $\alpha_k > 0$ is a stepsize parameter. From this design, a non-monotone line search [29] may avoid creeping along the bottom of a narrow curved valley and can generally obtain better numerical results.

1.1 Literature review

We first review some theoretical results that have been obtained for non-monotone line search. In the original paper [21], non-monotone line search was proven to converge globally, meaning that it converges for an arbitrary starting iterate point x_0, for smooth non-convex functions. Later, in [10], it was shown to exhibit a linear rate for uniformly convex functions. One variant using a moving average of past objective function values instead of their maximum was proposed and investigated in [29], where it was shown to converge globally for smooth non-convex functions

and to exhibit linear rates for strongly convex functions. For smooth non-convex functions, non-monotone line search was proven in [7] to find an ϵ-approximate first-order stationary point in at most $\mathcal{O}(\epsilon^{-2})$ function and gradient evaluations.

Another interesting application setting for non-monotone algorithms is when deterministic noise occurs in the objective function. The deterministic noise in the objective function may destroy the benign geometry of the objective function and cause many spurious local minima, even when the algorithm is still far from stationarity. To handle this problem, especially when noise is bounded, the authors in [3] add a constant non-monotonicity term $\epsilon_f > 0$ to the Armijo condition as follows:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \frac{1}{2}\alpha_k \nabla f(x_k)^\top d_k + \epsilon_f.$$

Similarly, to handle bounded deterministic noises, the authors in [6] consider the following acceptance condition in trust-region methods

$$\frac{f(x_k) - f(x_k + s_k) + r}{m_k(x_k) - m_k(x_k + s_k)} \geq \eta_1,$$

where $\eta_1 > 0$ is a parameter, s_k is a candidate point, m_k is a local model, and $r > 0$ is also a constant non-monotonicity parameter term. Different variants and analyzes following this lead can be found, for example, in [4, 23].

Now let us also review what has been proposed so far in non-monotone (derterministic) methods for DFO. The first non-monotone DFO algorithm known to us is a line-search algorithm that was proposed and shown in [13] to converge globally. At the beginning of one iteration in their linesearch algorithm, a set of search directions D_k is computed, and a stepsize α is set to 1. Then the algorithm repeatedly shrinks the stepsize until the following sufficient decrease condition is satisfied for one of the directions d in D_k

$$f(x_k + \alpha d) \leq \max_{0 \leq j \leq M} f(x_{k-j}) + \eta_k - \alpha^2 \beta_k,$$

where $\{\eta_k > 0\}$ satisfies $\sum_{k=0}^{\infty} \eta_k < \infty$ and $\{\beta_k > 0\}$ is selected in a way that for all infinite subsets of indices $K \subset \mathbb{N}$, one has $\lim_{k \in K} \beta_k = 0 \Rightarrow \lim_{k \in K} \nabla f(x_k) = 0$. Therefore, at the end of each iteration, under appropriate assumptions, a distinct x_{k+1} is always found. Direct search and line search are very similar algorithms, but the stepsize in direct search is maintained adaptively across different iterations. This distinction then requires a slightly different analysis for these two algorithms. Other non-monotone derivative-free algorithms include a parallel and sequential algorithm [15] using a space decomposition scheme for box constrained optimization problems, an inexact line-search algorithm [22] constructing search directions by combining coordinate rotations with simplex gradients, and a direct-search algorithm [14] for linearly constrained minimization problems. Although these derivative-free non-monotone methods have generally been shown to converge globally, their iteration complexity results have yet to be established. While finishing this paper we became aware of a new paper [16] establishing a worst-case complexity result for non-monotone direct search, where $f(x_k)$ in the sufficient decrease condition is replaced by $f(x_k) + \eta_k$, with $\eta_k \geq 0$ chosen to satisfy $\sum_{k=0}^{\infty} \eta_k < \infty$. This condition is different from the max-M acceptance rule that it requires a user-specified sequence η_k to control the non-monotonicity at each iteration k. In practice, parameter tuning is often needed to find the best values for η_k as these values usually depends on the scale of the objective function and the problem.

1.2 Our contributions

In this paper, we develop and analyze max-M non-monotone direct-search methods for deterministic and stochastic DFO. We replace the usual sufficient decrease condition

$$f(x_k + \delta_k d_k) \leq f(x_k) - c\delta_k^2,$$

with the following acceptance rule

$$f(x_k + \delta_k d_k) \leq \max\left(f(x_{succ,k}^{M-1}), \ldots, f(x_{succ,k}^{1}), f(x_k)\right) - c\delta_k^2,$$

where $c > 0$, $\delta_k > 0$ is the adaptive stepsize, d_k is a search direction, $x_{succ,k}^{i}$ is the i-th latest successful iterate from x_k, and $M > 1$ is a chosen memory length. The method may therefore accept a trial point whose objective value exceeds $f(x_k)$, but it still requires a decrease of order δ_k^2 compared to the largest function value in a window of M past iterates.

For deterministic objectives, we show theoretical analysis for direct-search methods with this acceptance rule: one based on a positive spanning set and one based on probabilistic descent directions. For the former, we establish a worst-case bound on the number of iterations required to produce an iterate x_k satisfying $\|\nabla f(x_k)\| \leq \epsilon$, while for the latter, we establish an expected iteration bound. Both analysis recover the standard $\mathcal{O}(\epsilon^{-2})$ dependence on the stationary tolerance ϵ for monotone direct-search methods. In addition, we propose a stochastic max-M direct-search algorithm where acceptance test uses independent stochastic estimates of the objective values at $x_k + \delta_k d_k$, x_k, and $x_{succ,k}^{i}$ for $i = 1, \ldots, M-1$. Under conditional probabilistic assumptions on the sampling error and on a probabilistic descent direction, we show an $\mathcal{O}(\epsilon^{-2})$ bound on the expected number of iterations for the proposed stochastic method. Finally, we show a competitive, sometimes better performance of deterministic and stochastic max-M algorithms compared to their monotone counterparts on two sets of CUTEst problems.

To the best of our knowledge, this is the first work that analyzes the non-monotone method in stochastic DFO, and also the first work that analyzes the max-M acceptance rule for direct search. The main technical challenge in our analysis is the construction of a merit function adapted to the max-M non-monotone decrease condition. The most natural, naive merit function consists of the maximum objective value among M prior iterates and a multiple of the squared stepsize, however, it may not decrease after a successful iteration because the stepsize is increased. To overcome this difficulty, we subtract ordered terms of the squared stepsize from the objective values before taking their maximum. These corrections are carefully chosen to account both for the shift of merit function per iteration and for the change in the stepsize. Most importantly, this merit function allows for a decrease which only depends on the acceptance condition at the current iteration. This local dependence makes such merit function feasible for the analysis of the subsequent stochastic algorithm.

2 Merit function design for max-M non-monotone direct search

Our convergence theory is based on the development of an appropriate merit (Lyapunov) function, and we would like to discuss the main properties of such functions that guided our efforts. Merit functions are commonly seen in constrained optimization [27] and are used in the analyses of many algorithms for continuous optimization of non-linear functions. They are designed to evaluate the progress of an iterative algorithm. They are usually bounded from below and

expected to decrease by a non-negligible amount at each iteration of an algorithm before a stationary point is found. Such properties of merit functions allow us to use them for the derivation of asymptotic global convergence or iteration complexity bounds. As optimization algorithms become more advanced and complicated, merit functions provide an intuitive and simplified approach to analyze these methods (e.g., [5] provides an framework for analyzing stochastic objective functions).

For monotone direct search, the objective value itself is a natural progress measure because every successful iteration satisfies

$$f(x_{k+1}) \leq f(x_k) - c\delta_k^2.$$

This argument is unavailable for max-M direct search. The accepted point is compared with the largest objective value among the current and $M-1$ most recent distinct iterates, and hence $f(x_{k+1})$ may exceed $f(x_k)$.

The most immediate and natural candidate for a merit function is

$$\widehat{\Phi}_k = \max\Big(f(x_{succ,k}^{M-1}), \ldots, f(x_{succ,k}^1), f(x_k)\Big) - f^* + \eta\delta_k^2,$$

where $\eta > 0$ is a constant parameter. This candidate is bounded below, but it does not necessarily decrease at each iteration. In fact, suppose that iteration k is successful and that

$$\max\Big(f(x_{succ,k}^{M-2}), \ldots, f(x_{succ,k}^1), f(x_k)\Big) \geq f(x_{succ,k}^{M-1}),$$

then we have

$$\begin{aligned}
\widehat{\Phi}_{k+1} &= \max\Big(f(x_{succ,k+1}^{M-1}), \ldots, f(x_{succ,k+1}^1), f(x_{k+1})\Big) + \eta\delta_{k+1}^2 - f^* \\
&= \max\Big(f(x_{succ,k}^{M-2}), \ldots, f(x_k), f(x_{k+1})\Big) + \eta\gamma^2\delta_k^2 - f^* \\
&\geq \max\Big(f(x_{succ,k}^{M-2}), \ldots, f(x_k)\Big) + \eta\gamma^2\delta_k^2 - f^* \\
&\geq \max\Big(f(x_{succ,k}^{M-1}), \ldots, f(x_{succ,k}^1), f(x_k)\Big) + \eta\gamma^2\delta_k^2 - f^*.
\end{aligned}$$

Since $\gamma > 1$, $\widehat{\Phi}_k$ increases by at least $\eta(\gamma^2 - 1)\delta_k^2$ in this case. Therefore, $\widehat{\Phi}_k$ does not offer a decrease, and we need a new merit function design for the max-M non-monotone condition.

Let $0 < c_1 < \cdots < c_{M-1} < \eta$ be an increasing sequence of constants. Our main design assigns different stepsize correction $c_i\delta_k^2$ to each prior objective value $f(x_{succ,k}^i)$ as follows:

$$\Phi_k = \max\Big(f(x_{succ,k}^{M-1}) - c_{M-1}\delta_k^2, \ldots, f(x_{succ,k}^1) - c_1\delta_k^2, f(x_k)\Big) - f^* + \eta\delta_k^2, \tag{2.1}$$

This merit function is non-negative, and the oldest stored value receives the largest correction. We note that Φ_k reduces to $\widehat{\Phi}_k$ when $c_1 = \cdots = c_{M-1} = 0$. On a successful iteration, the memory window shifts, the current iterate becomes a past iterate, and the stepsize is enlarged. The ordered corrections $c_i > 0$ are carefully chosen so that the quantity $c_i\gamma^2 - c_{i-1}$ is positive and the change in the merit function generates a decrease. On an unsuccessful iteration, the past points remain unchanged and the stepsize contracts; hence choosing η sufficiently large relative to c_{M-1} makes the decrease in $\eta\delta_k^2$ dominate the change in the corrected maximum. Balancing these two cases yields constants for which

$$\Phi_k - \Phi_{k+1} \geq \nu\delta_k^2,$$

for some constant $\nu > 0$.

For the direct-search method based on a positive spanning set, we utilize a simpler merit function, as this algorithm allows for a lower bound on the stepsize. As a result, the dynamic stepsize δ_k was replaced by a constant threshold δ_ϵ. In contrast, for the latter cases where the search direction is random and the function potentially contains noise, the stepsize does not admit a lower bound, which makes the analyis of the merit function much more challenging. For stochastic DFO, our analysis obtains a bound on the expected decrease of the merit function, instead of a deterministic bound.

3 Non-monotone direct search for deterministic functions

3.1 Max-M non-monotone direct search with positive spanning set

The first method in our consideration (presented in Algorithm 1) replaces the sufficient decrease condition in standard direct search with positive spanning set [9] by the max-M non-monotone condition. Without loss of generality, let $\mathcal{D}$ be a positive spanning set with cosine measure $1/\sqrt{n}$, e.g., $\mathcal{D} = \{\pm e_1, \ldots, \pm e_n\}$. The algorithm searches for a direction $d_k \in \mathcal{D}$ such that $f(x_k + \delta_k d_k)$ satisfies the decrease condition (3.1). If the algorithm finds a successful direction, the polling point becomes the next iterate and the stepsize is enlarged by γ; if no direction in $\mathcal{D}$ is successful, the iterate is retained and the stepsize is reduced by θ.

Algorithm 1 Max-M Non-monotone Direct Search Based on Positive Spanning Set

Initialization. Choose integer $M > 1$, $c > 0$, x_0, δ_0, $\theta \in (0, 1)$, $\gamma \in (1, \infty)$.
for $k = 0, 1, \ldots$ **do**
  For $i \in \mathbb{N}^+$, if there is an i-th latest successful iterate point from x_k, then record it with $x^i_{succ,k}$. Otherwise let $x^i_{succ,k}$ be x_0.
  if there exists $d_k \in \mathcal{D}$ such that

$$\max\Big(f(x^{M-1}_{succ,k}), \ldots, f(x^1_{succ,k}), f(x_k)\Big) - f(x_k + \delta_k d_k) \geq c\delta_k^2, \tag{3.1}$$

  then
    Set $x_{k+1} = x_k + \delta_k d_k$ and $\delta_{k+1} = \gamma\delta_k$. This iteration is successful.
  else
    Set $x_{k+1} = x_k$ and $\delta_{k+1} = \theta\delta_k$. This iteration is not successful.
  end if
end for

The goal of our analysis is to bound the number of iterations T_ϵ defined as follows

$$T_\epsilon = \inf\{k \geq 0 : \|\nabla f(x_k)\| \leq \epsilon\}.$$

We first assume that the objective function has a lower bound and its gradient is Lipschitz smooth, which are standard assumptions in first-order complexity analyses of smooth, non-convex direct-search methods.

Assumption 3.1 *The objective function $f(x)$ is bounded below by f^*.*

Assumption 3.2 *The gradient of the objective function $f(x)$ is Lipschitz continuous (with constant $L > 0$).*

As the objective is Lipschitz smooth, we show (using well known arguments) that before the gradient norm approaches ϵ, there exists a threshold δ_ϵ that any stepsize below δ_ϵ produces a successful iteration.

Lemma 3.1 *Let Assumption 3.2 hold and let $\delta_\epsilon = \frac{2\epsilon}{\sqrt{n}(L+2c)}$. For Algorithm 1, if $\delta_k \leq \delta_\epsilon$ and $k < T_\epsilon$, then iteration k is successful.*

Proof. It is well known (see, for example, [26, Lemma 1.2.3]) that Assumption 3.2 implies for any x, y from $\mathbb{R}^n$ that

$$f(x) - f(y) \geq \nabla f(x)^\top (x - y) - \frac{L}{2}\|x - y\|^2.$$

After substituting $x = x_k$ and $y = x_k + \delta_k d_k$, since $\|d_k\| = 1$, we obtain

$$f(x_k) - f(x_k + \delta_k d_k) \geq -\nabla f(x_k)^\top (\delta_k d_k) - \frac{L}{2}\delta_k^2. \tag{3.2}$$

Since the cosine measure of $\mathcal{D}$ is $1/\sqrt{n}$, by definition of cosine measure, we have

$$\max_{d_k \in \mathcal{D}} \frac{-\nabla f(x_k)^\top d_k}{\|\nabla f(x_k)\|\|d_k\|} \geq \frac{1}{\sqrt{n}}. \tag{3.3}$$

Note that $k < T_\epsilon$ implies $\|\nabla f(x_k)\| > \epsilon$. Together with (3.3) and $\|d_k\| = 1$, there exists $d_k \in \mathcal{D}$ such that

$$-\nabla f(x_k)^\top (\delta_k d_k) \geq \frac{1}{\sqrt{n}}\|\nabla f(x_k)\|\|d_k\|\delta_k > \frac{\epsilon\delta_k}{\sqrt{n}}. \tag{3.4}$$

After combining (3.2) and (3.4), we have: when $\delta_k \leq \delta_\epsilon = \frac{2\epsilon}{\sqrt{n}(L+2c)}$ and $k < T_\epsilon$, there exists $d_k \in \mathcal{D}$ such that

$$\begin{aligned}\max\Big(f(x^{M-1}_{succ,k}), \ldots, f(x^1_{succ,k}), f(x_k)\Big) - f(x_k + \delta_k d_k) &\geq f(x_k) - f(x_k + \delta_k d_k) \\ &\geq \frac{\epsilon\delta_k}{\sqrt{n}} - \frac{L}{2}\delta_k^2 \geq c\delta_k^2,\end{aligned}$$

which means iteration k is successful. □

From the result of Lemma 3.1, we show that all the stepsizes in Algorithm 1 admit a lower bound as follows.

Lemma 3.2 *Let Assumption 3.2 hold. In Algorithm 1, suppose that $\delta_\epsilon \leq \delta_0$ and $k \leq T_\epsilon$. We have*

$$\delta_k \geq \theta\delta_\epsilon.$$

Proof. We prove this by induction. For $k = 0$, we have $\delta_0 \geq \delta_\epsilon \geq \theta\delta_\epsilon$, and the conclusion holds. We assume for induction that $t \leq T_\epsilon$ and $\delta_t \geq \theta\delta_\epsilon$ for some arbitrary $t \geq 0$. Then it suffices to prove that $t + 1 \leq T_\epsilon$ implies $\delta_{t+1} \geq \theta\delta_\epsilon$. We prove this by cases. If $\delta_t \geq \delta_\epsilon$, then by the algorithm mechanism, we have $\delta_{t+1} \geq \theta\delta_\epsilon$. Otherwise, we have $\theta\delta_\epsilon \leq \delta_t < \delta_\epsilon$. Since $t + 1 \leq T_\epsilon$ implies $t < T_\epsilon$, using Lemma 3.1 shows that iteration t is successful. Therefore, the algorithm mechanism indicates that $\delta_{t+1} = \gamma\delta_t > \delta_t \geq \theta\delta_\epsilon$, which concludes the proof. □

We are interested in the number of successes before iteration k, denoted by $N_{succ,k}$. The stepsize mechanism of Algorithm 1 implies

$$\delta_k = \delta_0 \gamma^{N_{succ,k}} \theta^{k - N_{succ,k}}.$$

The following lemma shows a relation between $N_{succ,k}$ and the total number of iteration k.

Lemma 3.3 *Let Assumption 3.2 hold. In Algorithm 1, suppose that $\delta_\epsilon \leq \delta_0$ and $k \leq T_\epsilon$. We have*

$$k \leq \frac{\ln \frac{\delta_0}{\theta\delta_\epsilon} + N_{succ,k} \ln \frac{\gamma}{\theta}}{\ln \frac{1}{\theta}}.$$

Proof. From Lemma 3.2, we have $\delta_k \geq \theta\delta_\epsilon$. The result is obtained after using the fact that $\delta_k = \delta_0 \gamma^{N_{succ,k}} \theta^{k - N_{succ,k}}$ and rearranging. □

Let the merit function for Algorithm 1 be defined as

$$\phi_k = \max\left(f(x^{M-1}_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x^1_{succ,k}) - c_1\theta^2\delta_\epsilon^2, f(x_k)\right) - f^*,$$

where $0 < c_1 < \cdots < c_{M-1} < \eta$. Note that this function is slightly different from Φ_k in (2.1) which uses the squared stepsize δ_k^2. Now we are ready to present the decrease lemma for ϕ_k after a successful iteration.

Lemma 3.4 *Let Assumption 3.2 hold. Let $c_i = \frac{i}{M}c$, $1 \leq i \leq M - 1$. In Algorithm 1, suppose that $\delta_\epsilon \leq \delta_0$ and $k \leq T_\epsilon$. From the satisfaction of the sufficient decrease condition (3.1) at iteration k, we have*

$$\begin{aligned}
&\max\left(f(x^{M-1}_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x^1_{succ,k}) - c_1\theta^2\delta_\epsilon^2, f(x_k)\right) \\
&- \max\left(f(x^{M-2}_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x^1_{succ,k}) - c_2\theta^2\delta_\epsilon^2, f(x_k) - c_1\theta^2\delta_\epsilon^2, f(x_k + \delta_k d_k)\right) \\
&\geq \frac{c}{M}\theta^2\delta_\epsilon^2.
\end{aligned}$$

Proof. For ease of notation, we denote that

$$B_k = \max\left(f(x^{M-1}_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x^1_{succ,k}) - c_1\theta^2\delta_\epsilon^2, f(x_k)\right).$$

Since c_i, $1 \leq i \leq M - 1$, are monotonically increasing, one has

$$\begin{aligned}
B_k &= \max\left(f(x^{M-1}_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x^1_{succ,k}) - c_1\theta^2\delta_\epsilon^2, f(x_k)\right) \\
&\geq \max\left(f(x^{M-1}_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x^1_{succ,k}) - c_{M-1}\theta^2\delta_\epsilon^2, f(x_k) - c_{M-1}\theta^2\delta_\epsilon^2\right) \\
&= \max\left(f(x^{M-1}_{succ,k}), \ldots, f(x^1_{succ,k}), f(x_k)\right) - c_{M-1}\theta^2\delta_\epsilon^2,
\end{aligned}$$

which, together with (3.1) and Lemma 3.2, implies

$$\begin{aligned} B_k - f(x_k + \delta_k d_k) &\geq \max\left(f(x_{succ,k}^{M-1}), \ldots, f(x_{succ,k}^{1}), f(x_k)\right) - c_{M-1}\theta^2\delta_\epsilon^2 - f(x_k + \delta_k d_k) \\ &\geq c\delta_k^2 - c_{M-1}\theta^2\delta_\epsilon^2 \\ &\geq \frac{c}{M}\theta^2\delta_\epsilon^2. \end{aligned} \tag{3.5}$$

Note that

$$\begin{aligned} &B_k - \max\left(f(x_{succ,k}^{M-2}) - c_{M-1}\theta^2\delta_\epsilon^2, \ldots, f(x_{succ,k}^{1}) - c_2\theta^2\delta_\epsilon^2, f(x_k) - c_1\theta^2\delta_\epsilon^2, f(x_k + \delta_k d_k)\right) \\ &= B_k + \min\left(c_{M-1}\theta^2\delta_\epsilon^2 - f(x_{succ,k}^{M-2}), \ldots, c_1\theta^2\delta_\epsilon^2 - f(x_k), -f(x_k + \delta_k d_k)\right) \\ &= \min\left(B_k + c_{M-1}\theta^2\delta_\epsilon^2 - f(x_{succ,k}^{M-2}), \ldots, B_k + c_1\theta^2\delta_\epsilon^2 - f(x_k), B_k - f(x_k + \delta_k d_k)\right). \end{aligned}$$

Using the definition of B_k and (3.5) gives

$$\begin{aligned} &\min\left(B_k + c_{M-1}\theta^2\delta_\epsilon^2 - f(x_{succ,k}^{M-2}), \ldots, B_k + c_1\theta^2\delta_\epsilon^2 - f(x_k), B_k - f(x_k + \delta_k d_k)\right) \\ &\geq \min\left(c_{M-1}\theta^2\delta_\epsilon^2 - c_{M-2}\theta^2\delta_\epsilon^2, \ldots, c_2\theta^2\delta_\epsilon^2 - c_1\theta^2\delta_\epsilon^2, c_1\theta^2\delta_\epsilon^2, \frac{c}{M}\theta^2\delta_\epsilon^2\right) \\ &= \frac{c}{M}\theta^2\delta_\epsilon^2, \end{aligned}$$

which concludes the proof. □

As the merit function decrease a fixed amount after each successful iteration, the following theorem bounds the number of successful iterations before the function value is reduced to optimality. As a result, we obtain a bound on the total number of iterations for Algorithm 1.

Theorem 3.5 *Let Assumption 3.1 and 3.2 hold. Let $c_i = \frac{i}{M}c$, $1 \leq i \leq M-1$. In Algorithm 1, suppose that $\delta_\epsilon \leq \delta_0$. We have*

$$N_{succ,T_\epsilon} \leq \frac{f(x_0) - f^*}{\frac{c}{M}\theta^2\delta_\epsilon^2}.$$

Furthermore,

$$T_\epsilon \leq \frac{\ln\frac{\delta_0\sqrt{n}(L+2c)}{2\theta\epsilon}}{\ln\frac{1}{\theta}} + \frac{\ln\frac{\gamma}{\theta}}{\ln\frac{1}{\theta}}\frac{nM(L+2c)^2}{4c\theta^2\epsilon^2}\left(f\left(x_0\right) - f^*\right).$$

Proof. Let s_k denote the index of the k-th successful iteration in Algorithm 1. Lemma 3.4 and the mechanism of Algorithm 1 give

$$\begin{aligned} \sum_{0\leq k\leq T_\epsilon}(\phi_k - \phi_{k+1}) &= \sum_{\{j:0\leq s_j\leq T_\epsilon\}}\left(\phi_{s_j} - \phi_{s_j+1}\right) \\ &\geq \sum_{\{k:0\leq s_k\leq T_\epsilon\}}\frac{c}{M}\theta^2\delta_\epsilon^2 \\ &\geq \sum_{\{k:0\leq s_k\leq T_\epsilon-1\}}\frac{c}{M}\theta^2\delta_\epsilon^2 \\ &= N_{succ,T_\epsilon}\frac{c}{M}\theta^2\delta_\epsilon^2. \end{aligned}$$

We note that $\sum_{0\leq k\leq T_\epsilon}(\phi_k-\phi_{k+1})=\phi_0-\phi_{T_\epsilon+1}\leq\phi_0$. Therefore, we have

$$N_{succ,T_\epsilon}\frac{c}{M}\theta^2\delta_\epsilon^2\leq\phi_0.$$

Note that, by definition, $x^0_{succ,0}=\cdots=x^{M-1}_{succ,0}=x_0$, which implies $\phi_0=f(x_0)-f^*$. Then, applying Lemma 3.3 with $k=T_\epsilon$ delivers the claimed result after substituting $\delta_\epsilon=\frac{2\epsilon}{\sqrt{n}(L+2c)}$. □

The complexity of Algorithm 1 is $\mathcal{O}(n\epsilon^{-2})$, which matches the complexity of standard direct-search methods in number of iterations.

3.2 Max-M non-monotone direct search with probabilistic descent

In this section, we present Algorithm 2, which replaces the complete positive-spanning-set poll by a single random direction. This practice was first used in the probabilistic-descent assumption [20], reducing the search cost per iteration at the expense of guaranteeing a useful direction only with a sufficiently large probability. In contrast to the monotone method in [20], Algorithm 2 uses the maximum of the last M distinct iterates in the decrease condition and chooses one random direction D_k in each step. We note that a practical version of Algorithm 2 searches for a direction from $\{D_k,-D_k\}$ where D_k is sampled uniformly at random from the unit sphere.

Algorithm 2 Max-M Non-monotone Direct Search Based on Probabilistic Descent

Initialization. Choose integer $M>1$, $c>0$, X_0, Δ_0, $\theta\in(0,1)$, $\gamma\in(1,\infty)$.
for $k=0,1,\ldots$ **do**
  Uniformly select a random direction D_k from the unit sphere.
  For $i\in\mathbb{N}^+$, if there is an i-th latest successful iterate point from X_k, then record it with $X^i_{succ,k}$. Otherwise let $X^i_{succ,k}$ be X_0.
  if

$$\max\Big(f(X^{M-1}_{succ,k}),\ldots,f(X^1_{succ,k}),f(X_k)\Big)-f(X_k+\Delta_kD_k)\geq c\Delta_k^2, \tag{3.6}$$

  then
    Set $X_{k+1}=X_k+\Delta_kD_k$ and $\Delta_{k+1}=\gamma\Delta_k$. This iteration is successful.
  else
    Set $X_{k+1}=X_k$ and $\Delta_{k+1}=\theta\Delta_k$. This iteration is not successful.
  end if
end for

For the purpose of the merit-function analysis, we record the successful stepsizes $\Delta^i_{succ,k}$ associated with the i-th latest successful iterate $X^i_{succ,k}$ for $i\in\mathbb{N}^+$. Hence $\Delta^i_{succ,k}=\Delta_0$ for $X^i_{succ,k}=X_0$. We also use the notations $X^0_{succ,k}=X_k$ and $\Delta^0_{succ,k}=\Delta_k$ for convenience.

To formalize conditioning on the past, let $\mathcal{F}_k$ denote the σ-algebra generated by randomness up to the end of iteration $k-1$. For Algorithm 2, the randomness is from the previous polling directions $D_0,\ldots,D_{k-1}$. Similar to [20], we assume the following probabilistic-descent condition on the random directions.

Assumption 3.3 *The sequence D_k is p-probabilistically κ-descent, i.e., there exist $\kappa \in (0,1]$ and $p \in (0,1]$ such that, for every $k < T_\epsilon$,*

$$P\left(cm(D_k, -\nabla f(X_k)) \geq \kappa \,|\, \mathcal{F}_k\right) \geq p,$$

where $cm(a,b)$ denotes the cosine measure for the vectors $a, b \in \mathbb{R}^n$.

Assumption 3.3 states that the cosine measure of the random direction and the descent direction is favorable for some probability p. Hence, it is verifiable and independent of the algorithmic construction. For example, if D_k is uniformly generated from a unit sphere, the direction is descent with $\kappa \geq 1/(7\sqrt{n})$ for a sufficiently large probability [20]. This assumption enables the acceptance of the non-monotone decrease condition, which we demonstrate below.

Lemma 3.6 *Let Assumptions 3.2 and 3.3 hold and define*

$$\delta_\epsilon = \frac{2\kappa\epsilon}{L + 2c}.$$

Then, for every $k < T_\epsilon$,

$$P\left(\max\left(f(X_{succ,k}^{M-1}), \ldots, f(X_{succ,k}^{1}), f(X_k)\right) - f(X_k + \Delta_k D_k) \geq c\Delta_k^2 \mid \mathcal{F}_k\right) \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}} \geq p \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}}.$$

Proof. Using the argument in Lemma 3.1 we have

$$f(X_k) - f(X_k + \Delta_k D_k) \geq -\nabla f(X_k)^\top (\Delta_k D_k) - \frac{L}{2}\Delta_k^2. \tag{3.7}$$

Using the cosine measure in Assumption 3.3, $\|\nabla f(X_k)\| \geq \epsilon$ and $\|D_k\| = 1$, we have

$$f(X_k) - f(X_k + \Delta_k D_k) \geq \kappa\epsilon\Delta_k - \frac{L}{2}\Delta_k^2 \geq c\Delta_k^2,$$

where the last line follows from the fact that $\Delta_k \leq \delta_\epsilon$. Hence, the max-$M$ acceptance condition holds for every $k \leq T_\epsilon$. Taking conditional probabilities proves the result. □

Recall from (2.1) that the non-negative merit function sequence is defined by

$$\Phi_k = \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^{1}) - c_1\Delta_k^2, f(X_k)\right) - f^* + \eta\Delta_k^2, \tag{3.8}$$

where $0 < c_1 < \cdots < c_{M-1} < \eta$. The following lemma specifies the constants $\{c_i\}$ and bounds the change in the first term of this merit function when the iteration is successful.

Lemma 3.7 *Let $c_i = \frac{c}{\gamma^2}\frac{1-\gamma^{-2i}}{1-\gamma^{-2M}}$, $1 \leq i \leq M-1$. In Algorithm 2, from the satisfaction of the sufficient decrease condition (3.6) at iteration k, we have*

$$\begin{aligned}
&\max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^{1}) - c_1\Delta_k^2, f(X_k)\right) \\
&- \max\left(f(X_{succ,k}^{M-2}) - c_{M-1}\gamma^2\Delta_k^2, \ldots, f(X_{succ,k}^{1}) - c_2\gamma^2\Delta_k^2, f(X_k) - c_1\gamma^2\Delta_k^2, f(X_k + \Delta_k D_k)\right) \\
&\geq c\frac{1-\gamma^{-2}}{1-\gamma^{-2M}}\Delta_k^2.
\end{aligned}$$

Proof. Denote $B_k = \max\Big(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2, f(X_k)\Big)$. Since c_i, $1 \leq i \leq M-1$, are monotonically increasing, one has

$$\begin{aligned} B_k &= \max\Big(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2, f(X_k)\Big) \\ &\geq \max\Big(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_{M-1}\Delta_k^2, f(X_k) - c_{M-1}\Delta_k^2\Big) \\ &= \max\Big(f(X_{succ,k}^{M-1}), \ldots, f(X_{succ,k}^1), f(X_k)\Big) - c_{M-1}\Delta_k^2, \end{aligned}$$

which, together with (3.6), implies

$$\begin{aligned} B_k - f(X_k + \Delta_k D_k) &\geq \max\Big(f(X_{succ,k}^{M-1}), \ldots, f(X_{succ,k}^1), f(X_k)\Big) - c_{M-1}\Delta_k^2 - f(X_k + \Delta_k D_k) \\ &\geq (c - c_{M-1})\Delta_k^2. \end{aligned} \tag{3.9}$$

Note that

$$\begin{aligned} &B_k - \max\Big(f(X_{succ,k}^{M-2}) - c_{M-1}\gamma^2\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_2\gamma^2\Delta_k^2, f(X_k) - c_1\gamma^2\Delta_k^2, f(X_k + \Delta_k D_k)\Big) \\ &= B_k + \min\Big(c_{M-1}\gamma^2\Delta_k^2 - f(X_{succ,k}^{M-2}), \ldots, c_1\gamma^2\Delta_k^2 - f(X_k), -f(X_k + \Delta_k D_k)\Big) \\ &= \min\Big(B_k + c_{M-1}\gamma^2\Delta_k^2 - f(X_{succ,k}^{M-2}), \ldots, B_k + c_1\gamma^2\Delta_k^2 - f(X_k), B_k - f(X_k + \Delta_k D_k)\Big). \end{aligned}$$

Using the definition of B_k and (3.9) gives

$$\begin{aligned} &\min\Big(B_k + c_{M-1}\gamma^2\Delta_k^2 - f(X_{succ,k}^{M-2}), \ldots, B_k + c_1\gamma^2\Delta_k^2 - f(X_k), B_k - f(X_k + \Delta_k D_k)\Big) \\ &\geq \min\left(c_{M-1}\gamma^2\Delta_k^2 - c_{M-2}\Delta_k^2, \ldots, c_2\gamma^2\Delta_k^2 - c_1\Delta_k^2, c_1\gamma^2\Delta_k^2, (c - c_{M-1})\Delta_k^2\right), \end{aligned}$$

which concludes the proof after some basic calculations. □

We present the sufficient decrease lemma for the merit function of Algorithm 2 below.

Lemma 3.8 *There exist constants* $c_i = \frac{c}{\gamma^2}\frac{1-\gamma^{-2i}}{1-\gamma^{-2M}}$, $1 \leq i \leq M-1$, $\eta = \frac{c\left(\gamma^2 - \theta^2 - (1-\theta^2)\gamma^{2-2M}\right)}{\gamma^2(1-\gamma^{-2M})(\gamma^2-\theta^2)}$, *and* $\nu = \frac{c(\gamma^2-1)(1-\theta^2)\gamma^{-2M}}{(1-\gamma^{-2M})(\gamma^2-\theta^2)} > 0$ *such that*

$$\Phi_k - \Phi_{k+1} \geq \nu\Delta_k^2.$$

Proof. We consider separately whether iteration k is successful or not. Using the mechanism of the algorithm and Lemma 3.7, if iteration k is successful, then (3.6) holds and

$$\begin{aligned} &\Phi_k - \Phi_{k+1} \\ &= \max\Big(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2, f(X_k)\Big) + \eta(1-\gamma^2)\Delta_k^2 \\ &- \max\Big(f(X_{succ,k}^{M-2}) - c_{M-1}\gamma^2\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_2\gamma^2\Delta_k^2, f(X_k) - c_1\gamma^2\Delta_k^2, f(X_k + \Delta_k D_k)\Big) \\ &\geq \left(c\frac{1-\gamma^{-2}}{1-\gamma^{-2M}} + \eta(1-\gamma^2)\right)\Delta_k^2 \\ &= \nu\Delta_k^2. \end{aligned}$$

We will use the facts that $\max(a,b,c) = \max(c, \max(a,b))$ and that if $x \le y$, then $\max(a,x) - \max(a,y) \ge x - y$. Also note that $a - \max(b,c) = \min(a-b, a-c)$. These observations, together with the mechanism of the algorithm, imply that, if iteration k is not successful, then

$$\begin{aligned}
\Phi_k - \Phi_{k+1} =& \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2, f(X_k)\right) + \eta(1-\theta^2)\Delta_k^2 \\
&- \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\theta^2\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\theta^2\Delta_k^2, f(X_k)\right) \\
\ge& \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2)\right) + \eta(1-\theta^2)\Delta_k^2 \\
&- \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\theta^2\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\theta^2\Delta_k^2\right) \\
\ge& \min\left((\theta^2-1)c_{M-1}, \ldots, (\theta^2-1)c_1\right)\Delta_k^2 + \eta(1-\theta^2)\Delta_k^2 \\
=& (\theta^2-1)c_{M-1}\Delta_k^2 + \eta(1-\theta^2)\Delta_k^2 \\
=& \nu\Delta_k^2.
\end{aligned}$$

Therefore, from the algorithm, one has

$$\begin{aligned}
\Phi_k - \Phi_{k+1} &= (\Phi_k - \Phi_{k+1})(\mathbf{1}\{\text{iteration } k \text{ is successful}\} + \mathbf{1}\{\text{iteration } k \text{ is not successful}\}) \\
&\ge \nu\Delta_k^2(\mathbf{1}\{\text{iteration } k \text{ is successful}\} + \mathbf{1}\{\text{iteration } k \text{ is not successful}\}) \\
&= \nu\Delta_k^2.
\end{aligned}$$

□

We now explain how the stopping-time framework of Blanchet et al. [5] applies. This framework relies on the two following conditions.

1. There exists a fixed threshold δ_ϵ such that for $\Delta_k \le \delta_\epsilon$ and $k < T_\epsilon$, the stepsize is multiplied by γ with conditional probability at least p and otherwise is multiplied by θ. This is proved in Lemma 3.6. In addition, we assume that $p\log\gamma + (1-p)\log\theta > 0$, which prevents the method from spending too many iterations at stepsizes much smaller than δ_ϵ.

2. For $k < T_\epsilon$, the merit function satisfies $E[\Phi_k - \Phi_{k+1}|\mathcal{F}_k] \ge \nu\Delta_k^2$ for some values $\nu > 0$. This is proved in Lemma 3.8.

With these two conditions, the framework in [5, 12] studies a renewal–reward martingale process generated by the stepsize Δ_k and shows an upper bound for the expected stopping time $E[T_\epsilon]$ of the algorithm. We note that Blanchet et al. [5] analyzed the standard case with $\gamma\theta = 1$ and $p > \frac{1}{2}$, while Ding et al. [12] subsequently stated the corresponding bound for a general success probability and multiplicative factors that satisfy $p\log\gamma + (1-p)\log\theta > 0$.

Theorem 3.9 (expected number of iterations) *Let Assumptions 3.1, 3.2, and 3.3 hold, let $\delta_\epsilon = 2\kappa\epsilon/(L+2c)$, and suppose that $p\ln\gamma + (1-p)\ln\theta > 0$. Then*

$$E[T_\epsilon - 1] \le \frac{\ln\gamma}{p\ln\gamma + (1-p)\ln\theta} \cdot \frac{\Phi_0}{\nu\theta^2\delta_\epsilon^2} = \frac{\ln\gamma}{p\ln\gamma + (1-p)\ln\theta} \cdot \frac{\Phi_0(L+2c)^2}{4\nu\theta^2\kappa^2\epsilon^2},$$

where the first bound follows from the stopping-time framework [5]. Consequently, the expected iteration complexity of Algorithm 2 is $\mathcal{O}(\epsilon^{-2})$.

The explicit constant ν in Lemma 3.8 decreases geometrically with M, so the resulting bound contains a factor of order γ^{2M}. In contrast, the complete-poll analysis gives linear dependence on M. The geometric factor is a consequence of the particular feasible correction coefficients used here, not a lower bound for every max-M method; hence, the possibility of sharper coefficients is open.

We note that another merit function Ψ_k for Algorithm 2 can be constructed differently from Φ_k, which we present in detail in Appendix A. When extending to the stochastic objective functions, Ψ_k has a disadvantage. Specifically, the decrease of Ψ_k relies on the exactness of M sufficient decrease condition evaluations; therefore, error control in the algorithm for the stochastic objective function becomes relatively difficult when the stepsize is adaptive. The benefit of Φ_k is that its decrease depends merely on the current sufficient decrease condition evaluation and extends more naturally to the stochastic case, as shown in the next section.

4 Non-monotone direct search for stochastic functions

A challenge in the stochastic setting arises when the past and the current stochastic function estimates in the decrease condition are not independent. A common technique to overcome this challenge is resampling the random estimates in new iterations. We propose a stochastic max-M non-monotone direct-search algorithm, following this principle. Let $f(x)$ be the objective function of our interest and let $F(x, \xi)$ be its stochastic oracle. Our method extends Algorithm 2 by replacing the exact objective value in the non-monotone condition with an independent stochastic estimate, which we presented in Algorithm 3 as follows.

Algorithm 3 Stochastic Max-M Non-monotone Direct Search

Initialization. Choose integer $M > 1$, $c > 0$, X_0, Δ_0, $\theta \in (0, 1)$, $\gamma \in (1, \infty)$.
for $k = 0, 1, \ldots$ **do**
  Uniformly select a random direction D_k from the unit sphere.
  For $i \in \mathbb{N}^+$, if there is an i-th latest successful iterate point from X_k, then record it with $X^i_{succ,k}$. Otherwise let $X^i_{succ,k}$ be X_0. Let $X^d_k = X_k + \Delta_k D_k$ be the candidate point.
  Query the stochastic oracle at $X^i_{succ,k}$, $1 \le i \le M-1$, X_k, and X^d_k. Denote the query results by $F(X^i_{succ,k}, \xi^i_{k,1})$, $1 \le i \le M-1$, $F(X_k, \xi_{k,2})$, and $F(X^d_k, \xi_{k,3})$. Here the random variables $\xi^i_{k,1}$, $1 \le i \le M-1$, $\xi_{k,2}$, and $\xi_{k,3}$ are independent.
  if $\max\left(F(X^{M-1}_{succ,k}, \xi^{M-1}_{k,1}), \ldots, F(X^1_{succ,k}, \xi^1_{k,1}), F(X_k, \xi_{k,2})\right) - F(X^d_k, \xi_{k,3}) \ge c\Delta_k^2$, **then**
    Set $X_{k+1} = X^d_k$ and $\Delta_{k+1} = \gamma\Delta_k$. This iteration is successful.
  else
    Set $X_{k+1} = X_k$ and $\Delta_{k+1} = \theta\Delta_k$. This iteration is not successful.
  end if
end for

We denote the following quantities for the analysis of Algorithm 3.

$$G_k = \max\left(F(X^{M-1}_{succ,k}, \xi^{M-1}_{k,1}), \ldots, F(X^1_{succ,k}, \xi^1_{k,1}), F(X_k, \xi_{k,2})\right) - F(X^d_k, \xi_{k,3}),$$
$$H_k = \max\left(f(X^{M-1}_{succ,k}), \ldots, f(X^1_{succ,k}), f(X_k)\right) - f(X^d_k),$$

$$\varepsilon_k = G_k - H_k.$$

Here, ε_k is the sampling noise of G_k when estimating H_k. Hence, the random sources in the algorithm are $\{D_k\}$ and $\{\varepsilon_k\}$. Similar to the previous section, we let $\mathcal{F}_k$ be the σ-algebra generated by randomness up to the end of iteration $k-1$. In addition, let $\mathcal{F}_{k+1/2}$ denote the σ-algebra generated by randomness up to D_k. Therefore, X_k^d and H_k are $\mathcal{F}_{k+1/2}$-measurable; G_k, X_{k+1}, and Δ_{k+1} are $\mathcal{F}_{k+1}$-measurable. We denote the event $A_k := \mathbf{1}(\text{iteration } k \text{ is successful})$. Similar to the deterministic setting, we aim to bound T_ϵ for a given threshold $\epsilon > 0$. We impose the following assumption on the sampling error ϵ_k.

Assumption 4.1 *The stochastic oracle is accurate such that the distribution of ε_k satisfies for each k*

$$P(\varepsilon_k \geq 0 | \mathcal{F}_{k+1/2}) \geq p_0$$
$$P(\varepsilon_k \geq b | \mathcal{F}_{k+1/2}) \leq \frac{\beta \Delta_k^2}{b}, \quad \forall b > 0.$$

Assumption 4.1 controls the two ways in which sampling error can affect the acceptance decision. The first condition ensures with probability at least p_0 that the estimated decrease does not understate the true decrease, while the second condition bounds the upper tail of the error. We note that this assumption concerns the error bound between H_k and its stochastic estimate G_k, not for each function observation taken individually. The first condition can be attained for $p_0 = 1/2$ if the individual errors are symmetric. In the case of non-symmetric errors, the value of p_0 depends on the distribution of those errors. In general, our analysis does not require a lower bound on p_0 and thus the first condition is very mild. The second condition is a one-sided tail bound, which is common in the stochastic direct search literature [28]. In Appendix B, we show that this condition can be satisfied when the number of samples are $\mathcal{O}(\Delta_k^{-4})$. We note that this order of samples is also standard in the literature [28] to ensure the same (deterministic) iteration complexity for stochastic direct search.

We also need Assumption 3.3, which assumes good behavior of the random direction and therefore is independent of the stochastic oracles. By Lemma 3.6, for every $k < T_\epsilon$, we have

$$P\left(H_k \geq c\Delta_k^2 \mid \mathcal{F}_k\right) \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}} \geq p \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}}. \tag{4.1}$$

Independence between the direction and the subsequent oracle samples allows this probability and the probability in Assumption 4.1 to combine into pp_0 in the following lemma.

Lemma 4.1 *Let Assumptions 3.2, 3.3, and 4.1 hold and $\delta_\epsilon = \frac{2\kappa\epsilon}{L+2c}$. For any $\epsilon > 0$ and $k < T_\epsilon$, we obtain*

$$P(A_k = 1 | \mathcal{F}_k) \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}} \geq pp_0 \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}}.$$

Proof. Note that $\{H_k \geq c\Delta_k^2\} \cap \{G_k - H_k \geq 0\} \subseteq \{A_k = 1\}$. Therefore, we have

$$P(A_k = 1 | \mathcal{F}_k) \geq P\left(\{H_k \geq c\Delta_k^2\} \cap \{G_k - H_k \geq 0\} | \mathcal{F}_k\right)$$

Using the tower property of conditional expectation and the definition of ε, from Assumption 4.1, we have

$$P(A_k = 1 | \mathcal{F}_k) \geq E[\mathbf{1}_{\{H_k \geq c\Delta_k^2\}} P\left(G_k - H_k \geq 0 | \mathcal{F}_{k+1/2}\right) | \mathcal{F}_k]$$

$$
\begin{aligned}
&= E[\mathbf{1}_{\{H_k \geq c\Delta_k^2\}} P\left(\varepsilon_k \geq 0 | \mathcal{F}_{k+1/2}\right) | \mathcal{F}_k] \\
&\geq E[\mathbf{1}_{\{H_k \geq c\Delta_k^2\}} p_0 | \mathcal{F}_k] \\
&= p_0 P\left(H_k \geq c\Delta_k^2 | \mathcal{F}_k\right).
\end{aligned}
$$

Together with the bound (4.1), we have

$$
\begin{aligned}
P(A_k = 1 | \mathcal{F}_k) \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}} &\geq p_0 P\left(H_k \geq c\Delta_k^2 | \mathcal{F}_k\right) \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}} \\
&\geq p p_0 \mathbf{1}_{\{\Delta_k \leq \delta_\epsilon\}}.
\end{aligned}
$$

□

The merit function Φ_k is defined similarly as in (3.8). The next Lemma establishes the coefficients and the expected decrease of Φ_k as follows.

Lemma 4.2 *Let Assumption 4.1 hold for $\beta = \frac{1}{2}(\eta - c_{M-1})(1-\theta^2) > 0$. There exist $c_{M-1} > 0$, $c_i = \frac{1-\gamma^{-2i}}{1-\gamma^{2-2M}} c_{M-1}$, $1 \leq i \leq M-2$, $\eta > c_{M-1}$, and $\nu > 0$ such that*

$$
E[\Phi_k - \Phi_{k+1} | \mathcal{F}_{k+1/2}] \geq \nu \Delta_k^2.
$$

Proof. Selecting $c_{M-1} > 0$, $\eta > c_{M-1}$, and $\nu > 0$ such that

$$
\nu = \frac{\gamma^2 - 1}{1 - \gamma^{2-2M}} c_{M-1} + \eta(1 - \gamma^2) = \frac{1}{2}(\eta - c_{M-1})(1 - \theta^2)
$$

and

$$
c = c_{M-1}\theta^2 + \eta\gamma^2 - \eta\theta^2. \tag{4.2}
$$

This gives

$$
\begin{aligned}
c_{M-1} &= c \frac{(2\gamma^2 - \theta^2 - 1)(1 - \gamma^{2-2M})}{\text{aux}} \\
\eta &= c \frac{2\gamma^2 - 2 + (1-\theta^2)(1 - \gamma^{2-2M})}{\text{aux}} \\
\nu &= c \frac{(1-\theta^2)(\gamma^2 - 1)\gamma^{2-2M}}{\text{aux}},
\end{aligned}
$$

where $\text{aux} = \theta^2(2\gamma^2 - \theta^2 - 1)(1 - \gamma^{2-2M}) + (\gamma^2 - \theta^2)\left(2\gamma^2 - 2 + (1-\theta^2)(1-\gamma^{2-2M})\right)$ is positive.

We will use similar arguments as in Lemma 3.8. In particular, we note again that if $x \leq y$, then $\max(a, x) - \max(a, y) \geq x - y$ and that $a - \max(b, c) = \min(a - b, a - c)$. Therefore, if iteration k is not successful, then, from the mechanism of the algorithm,

$$
\begin{aligned}
\Phi_k - \Phi_{k+1} = &\max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2, f(X_k)\right) + \eta\Delta_k^2 \\
&- \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\theta^2\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\theta^2\Delta_k^2, f(X_k)\right) - \eta\theta^2\Delta_k^2 \\
\geq &\max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\Delta_k^2\right) + \eta\Delta_k^2 \\
&- \max\left(f(X_{succ,k}^{M-1}) - c_{M-1}\theta^2\Delta_k^2, \ldots, f(X_{succ,k}^1) - c_1\theta^2\Delta_k^2\right) - \eta\theta^2\Delta_k^2
\end{aligned}
$$

$$
\begin{aligned}
&\geq \min\left((\theta^2-1)c_{M-1},\ldots,(\theta^2-1)c_1\right)\Delta_k^2+\eta\Delta_k^2-\eta\theta^2\Delta_k^2\\
&=(\eta-c_{M-1})(1-\theta^2)\Delta_k^2.
\end{aligned}
$$

Also using the definition of H_k, the mechanism of the algorithm and $a-\max(b,c)=\min(a-b,a-c)$, if iteration k is successful, then (note also that $c_i<c_{M-1}$)

$$
\begin{aligned}
&\Phi_k-\Phi_{k+1}\\
=&\max\left(f(X_{succ,k}^{M-1})-c_{M-1}\Delta_k^2,\ldots,f(X_{succ,k}^1)-c_1\Delta_k^2,f(X_k)\right)+\eta\Delta_k^2\\
&-\max\left(f(X_{succ,k}^{M-2})-c_{M-1}\gamma^2\Delta_k^2,\ldots,f(X_k)-c_1\gamma^2\Delta_k^2,f(X_k^d)\right)-\eta\gamma^2\Delta_k^2\\
\geq&\min\left(c_{M-1}\gamma^2\Delta_k^2-c_{M-2}\Delta_k^2,\ldots,c_2\gamma^2\Delta_k^2-c_1\Delta_k^2,c_1\gamma^2\Delta_k^2,H_k-c_{M-1}\Delta_k^2\right)+\eta(1-\gamma^2)\Delta_k^2\\
=&\min\left(\frac{\gamma^2-1}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2,H_k-c_{M-1}\Delta_k^2\right)+\eta(1-\gamma^2)\Delta_k^2.
\end{aligned}
$$

Based on the arguments above, one is able to bound below $\Phi_k-\Phi_{k+1}$, which is $\mathcal{F}_{k+1}$-measurable, by Δ_k and H_k, which are $\mathcal{F}_{k+1/2}$-measurable, when A_k equals 0 and 1 separately. Therefore, one has

$$
\begin{aligned}
E[\Phi_k-\Phi_{k+1}|\mathcal{F}_{k+1/2}]&=E[(\Phi_k-\Phi_{k+1})\mathbf{1}(A_k=1)|\mathcal{F}_{k+1/2}]+E[(\Phi_k-\Phi_{k+1})\mathbf{1}(A_k=0)|\mathcal{F}_{k+1/2}]\\
&\geq\left(\min\left(\frac{\gamma^2-1}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2,H_k-c_{M-1}\Delta_k^2\right)+\eta(1-\gamma^2)\Delta_k^2\right)P(A_k=1|\mathcal{F}_{k+1/2})+\\
&\quad(\eta-c_{M-1})(1-\theta^2)\Delta_k^2P(A_k=0|\mathcal{F}_{k+1/2})\\
&=\left(\min\left(\frac{\gamma^2-1}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2,H_k-c_{M-1}\Delta_k^2\right)+\eta(1-\gamma^2)\Delta_k^2\right)P(A_k=1|\mathcal{F}_{k+1/2})+\\
&\quad(\eta-c_{M-1})(1-\theta^2)\Delta_k^2(1-P(A_k=1|\mathcal{F}_{k+1/2}))\\
&=\left(\min\left(\frac{\gamma^2-1}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2+c_{M-1}\Delta_k^2,H_k\right)+\eta(1-\gamma^2)\Delta_k^2-c_{M-1}\Delta_k^2\right)P(A_k=1|\mathcal{F}_{k+1/2})+\\
&\quad(\eta-c_{M-1})(1-\theta^2)\Delta_k^2(1-P(A_k=1|\mathcal{F}_{k+1/2}))\\
&=\left(\min\left(\frac{\gamma^2-\gamma^{2-2M}}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2,H_k\right)+(\eta\theta^2-\eta\gamma^2-c_{M-1}\theta^2)\Delta_k^2\right)P(A_k=1|\mathcal{F}_{k+1/2})+\\
&\quad(\eta-c_{M-1})(1-\theta^2)\Delta_k^2.
\end{aligned}
$$

Note that $S_k^1=\{\omega: H_k\geq\frac{\gamma^2-\gamma^{2-2M}}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2\}$ and $S_k^2=\{\omega: H_k<\frac{\gamma^2-\gamma^{2-2M}}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2\}$ are two disjoint $\mathcal{F}_{k+1/2}$-measurable events such that $S_k^1\cup S_k^2=\Omega$. It suffices to prove $E[\Phi_k-\Phi_{k+1}|\mathcal{F}_{k+1/2}]\geq\nu\Delta_k^2$ for these two scenarios separately.

Case 1. First, we consider the case $S_k^1=\{\omega: H_k\geq\frac{\gamma^2-\gamma^{2-2M}}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2\}$. Then one has

$$
\begin{aligned}
E[\Phi_k-\Phi_{k+1}|\mathcal{F}_{k+1/2}]\mathbf{1}_{S_k^1}&\geq\left(\frac{\gamma^2-1}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2+\eta(1-\gamma^2)\Delta_k^2\right)P(A_k=1|\mathcal{F}_{k+1/2})\mathbf{1}_{S_k^1}+\\
&\quad(\eta-c_{M-1})(1-\theta^2)\Delta_k^2P(A_k=0|\mathcal{F}_{k+1/2})\mathbf{1}_{S_k^1}\\
&\geq\min\left(\frac{\gamma^2-1}{1-\gamma^{2-2M}}c_{M-1}\Delta_k^2+\eta(1-\gamma^2)\Delta_k^2,(\eta-c_{M-1})(1-\theta^2)\Delta_k^2\right)\mathbf{1}_{S_k^1}
\end{aligned}
$$

$$=\nu\Delta_k^2 \mathbf{1}_{S_k^1}.$$

Case 2. Otherwise, we consider $S_k^2 = \{\omega : H_k < \frac{\gamma^2-\gamma^{2-2M}}{1-\gamma^{2-2M}} c_{M-1}\Delta_k^2\}$.Then one has

$$E[\Phi_k - \Phi_{k+1}|\mathcal{F}_{k+1/2}]\mathbf{1}_{S_k^2} \geq \left(H_k + (\eta\theta^2 - \eta\gamma^2 - c_{M-1}\theta^2)\Delta_k^2\right) P(A_k = 1|\mathcal{F}_{k+1/2})\mathbf{1}_{S_k^2} + (\eta - c_{M-1})(1-\theta^2)\Delta_k^2 \mathbf{1}_{S_k^2}. \tag{4.3}$$

Since $0 < \theta < 1 < \gamma$, from the definition of c in (4.2), one can check that

$$H_k < \frac{\gamma^2 - \gamma^{2-2M}}{1-\gamma^{2-2M}} c_{M-1}\Delta_k^2 < c\Delta_k^2.$$

Then, from Assumption 4.1, the choice of $\beta = \frac{1}{2}(\eta - c_{M-1})(1-\theta^2)$, and (4.2),

$$\begin{aligned} P(A_k = 1|\mathcal{F}_{k+1/2})\mathbf{1}_{S_k^2} &= P(G_k \geq c\Delta_k^2|\mathcal{F}_{k+1/2})\mathbf{1}_{S_k^2} \\ &= P(\varepsilon_k \geq c\Delta_k^2 - H_k|\mathcal{F}_{k+1/2})\mathbf{1}_{S_k^2} \\ &\leq \frac{1}{2}\frac{(\eta - c_{M-1})(1-\theta^2)\Delta_k^2}{c\Delta_k^2 - H_k}\mathbf{1}_{S_k^2} \\ &= -\frac{1}{2}\frac{(\eta - c_{M-1})(1-\theta^2)\Delta_k^2}{H_k + (\eta\theta^2 - \eta\gamma^2 - c_{M-1}\theta^2)\Delta_k^2}\mathbf{1}_{S_k^2}, \end{aligned}$$

which gives, from (4.3),

$$E[\Phi_k - \Phi_{k+1}|\mathcal{F}_{k+1/2}]\mathbf{1}_{S_k^2} \;\geq\; \frac{1}{2}(\eta - c_{M-1})(1-\theta^2)\Delta_k^2\mathbf{1}_{S_k^2} \;=\; \nu\Delta_k^2\mathbf{1}_{S_k^2}.$$

□

Similar to Theorem 3.9, we apply the stopping time framework [5, 12]. In this case, Lemma 4.1 shows the success probability pp_0 instead of p, and Lemma 4.2 bounds the expected decrease of the merit function. We have the following Theorem.

Theorem 4.3 (expected number of iterations) *Let Let Assumptions 3.1, 3.2, and 3.3 hold and $\delta_\epsilon = 2\kappa\epsilon/(L+2c)$. In addition, let Assumption 4.1 hold for $\beta = \frac{1}{2}(\eta - c_{M-1})(1-\theta^2) > 0$ and suppose that $pp_0 \ln\gamma + (1-pp_0)\ln\theta > 0$. Then*

$$E[T_\epsilon - 1] \leq \frac{\ln\gamma}{pp_0\ln\gamma + (1-pp_0)\ln\theta} \cdot \frac{\Phi_0}{\nu\theta^2\delta_\epsilon^2} = \frac{\ln\gamma}{pp_0\ln\gamma + (1-pp_0)\ln\theta} \cdot \frac{\Phi_0(L+2c)^2}{4\nu\theta^2\kappa^2\epsilon^2},$$

where the first bound follows from the stopping-time framework [5]. Consequently, the expected iteration complexity of Algorithm 3 is $\mathcal{O}(\epsilon^{-2})$.

5 Experiment results

5.1 Experimental setup

In this section, we assess the numerical performance of max-M non-monotone direct search on two sets of problems from the CUTEst collection [18]. The first set consists of 38 problems

suggested by [17], with dimensions in $\{5, 10, 100\}$. These problems exhibit a range of structural properties, including non-linearity, non-convexity, and partial separability. The second set consists of 53 problems selected from [17, 19], with dimensions ranging from 2 to 50. For each problem in this set, negative curvature was detected by running one of the second-order methods reported in [2]. The names and corresponding dimensions of such problems are reported in Appendix C.

We summarize the performance of all methods across the problems using performance profiles [25], which are briefly reviewed below. Let $\mathcal{S}$ be the set of solvers, $\mathcal{P}$ be the set of test problems, and $t_{p,s} > 0$ the performance measure obtained by solver $s \in \mathcal{S}$ on problem $p \in \mathcal{P}$ (smaller $t_{p,s}$ is better). We define the performance ratio $r_{p,s}$ and the performance profile $\rho_s(\alpha)$ as follows

$$r_{p,s} \;=\; \frac{t_{p,s}}{\min\{t_{p,s} \mid s \in \mathcal{S}\}}, \quad \rho_s(\alpha) \;=\; \frac{1}{|\mathcal{P}|}\text{size}\{p \in \mathcal{P} \mid r_{p,s} \leq \alpha\}.$$

If a solver s fails to satisfy the convergence test on a problem p, we set $r_{p,s} = 2\max_{p,s} r_{p,s}$. We plot $\rho_s(\alpha)$ as a curve over α. The solver associated with the highest curve is the one with the best performance. In particular, $\rho_s(1)$ is the fraction of problems for which solver s performs the best, and the value $\rho_s(\alpha)$ for large α represents the fraction of problems solved by solver s. These quantities provide measures of relative efficiency and robustness, respectively.

For all methods, we set $\gamma = 2$, $\theta = 0.5$, and $c = 1$. We compare different choices of M in max-M non-monotone direct search where M takes values in $\{2, 5, 10, 20\}$ and impose a budget of 1000 function evaluations per run. For every solver-problem pair, we perform 10 runs using random seeds $0, 1, \ldots, 9$ and report the average objective-value. In all the experiments, we sample a direction d uniformly from the unit sphere. If this direction does not yield a successful point, we proceed with the the opposite direction $-d$ before generating a new random direction. Since $M = 2$ provides the strongest overall performance among the tested values, we use this choice when comparing the max-M non-monotone to monotone methods.

5.2 Deterministic results

In this section, we present the results for the case where function values are deterministic. The left panel of Figure 5.1 displays the performance profiles comparing the max-M non-monotone direct search (Algorithm 2) for different values of $M \in \{2, 5, 10, 20\}$. Across both problem sets, the max-2 variant clearly dominates the others in terms of both efficiency and robustness. While our empirical and theoretical results confirm that the max-M method converges for different values of M, its performance deteriorates as the memory size M increases. This observation is not surprising as historical function values are generally high, which leads to more acceptances and more steps to find good function decrease.

The right panel of Figure 5.1 compares the max-2 non-monotone method against standard monotone direct search. For Set 1, the monotone method demonstrates better efficiency, while the max-2 non-monotone algorithm is more robust. For Set 2, the max-2 non-monotone method shows competitive efficiency and achieves a better robustness compared to the monotone approach. Hence, non-monotone method yields good performance overall, especially when the problem has negative curvature as in Set 2.

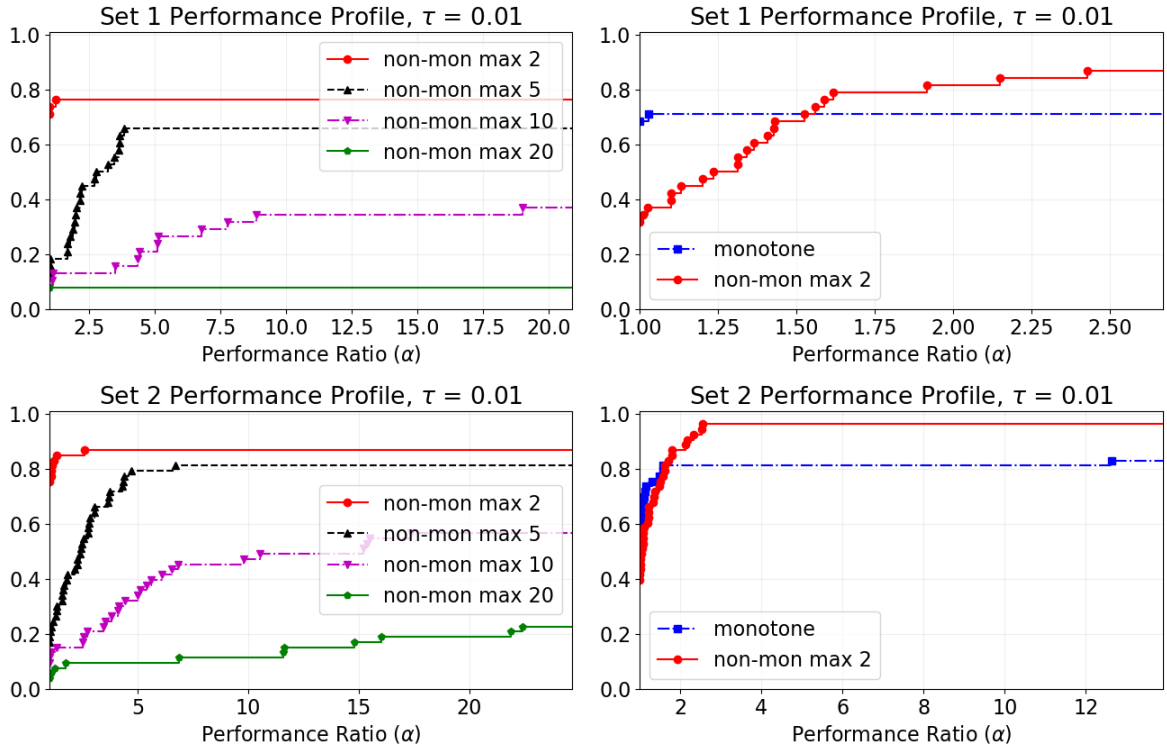

Figure 5.1: Performance profiles for two problem sets in the deterministic case.

5.3 Stochastic results

For the stochastic experiments, we perturb the objective values with Gaussian noise and choose the number of samples of order $\mathcal{O}(\delta_k^4)$. We implement Algorithm 3 where the stochastic function values at prior successful iterates are resampled independently. As a result, in this theoretical scenario, the max-M method requires $M+1$ stochastic function estimates per iteration, which is higher than the two estimates for the monotone method. Alternatively, we implement a practical scenario, where only the stochastic function value at the candidate point X_k^d is resampled in each iteration. Hence, the per-iteration costs of max-M non-monotone and monotone methods are the same (one stochastic evaluation), while the random estimates used in the acceptance test are no longer independent.

In the theoretical scenario (Figure 5.2), where all values are resampled, the monotone direct search outperforms the max-2 non-monotone one on Set 1, while on Set 2, max-2 non-monotone is better. This is consistent with the observation in the deterministic case that max-2 non-monotone offers significant improvement when the problems have negative curvature. In the practical scenario (Figure 5.3), where only the new point is sampled and the per-iteration cost is the same for both methods, the performance of max-2 non-monotone algorithm is better than the monotone one in both initial efficiency and eventual robustness.

A natural question arises whether the non-monotone condition should use the most recent function values from all iterations or only those associated with successful iterates. Our experiments indicate that the two choices yield broadly comparable performance for the two problem sets in both deterministic and stochastic settings, therefore, we omit the results.

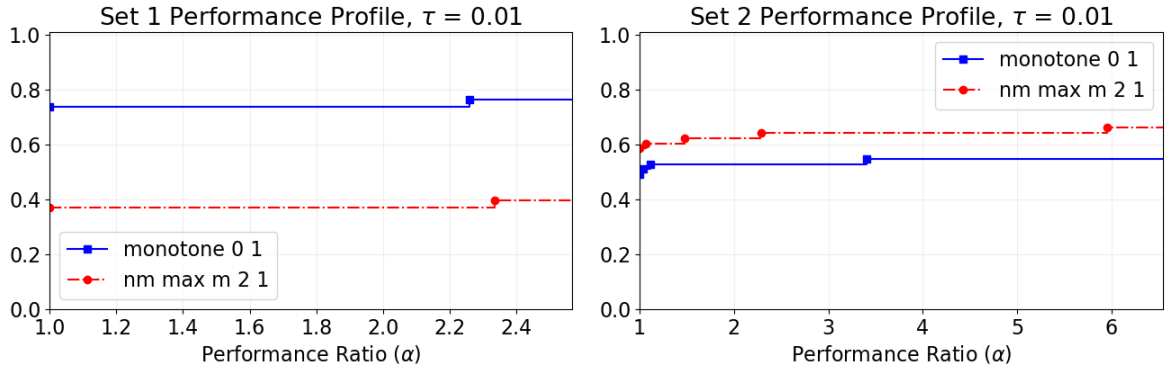

Figure 5.2: Performance profiles for two problem sets in the stochastic case, theoretical scenario: all stochastic function values are resampled.

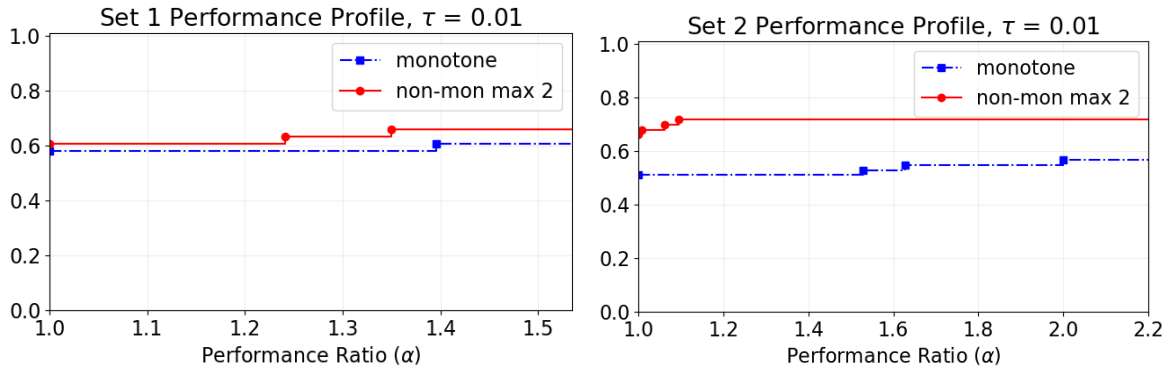

Figure 5.3: Performance profiles for two problem sets in stochastic case, practical scenario: only sample the value of stochastic function at X_k^d.

Overall, the stochastic results indicate that the max-2 non-monotone method performs consistently well on problems with negative curvature, but resampling all historical function values can offset its benefits when stochastic evaluations are expensive.

6 Conclusions and future work

We studied direct-search methods in which a trial point is compared with the largest objective value among the M most recent distinct iterates, rather than with the current value alone. This acceptance rule permits temporary objective increases while retaining a quantified decrease relative to the recent function values. We developed complexity analyses for three settings: deterministic complete polling, deterministic probabilistic descent, and probabilistic descent with stochastic function estimates. Together, these results provide a comprehensive complexity theory for non-monotone derivative-free optimization. We also conducted thorough experiments to compare max-M non-monotone direct search with the monotone method. Our experiment offers insights into the advantages of the practical choice $M = 2$ and the option of using historical values instead of resampling.

While both max-M non-monotone and monotone direct search converge and have competitive empirical performance, future work could identify the problem geometries (such as suitable classes of curved valleys) for which non-monotone acceptance has a provable advantage. A non-monotone technique using the averages of objective values at recent iterates could also be an interesting topic.

On the other hand, we expect that merit function design developed in this paper can be adapted to establish complexity bounds for line-search and trust-region methods. In a line-search framework, the usual monotone sufficient-decrease condition could be replaced by one based on the maximum objective value over a finite window of recent iterates. The analysis would then combine probabilistically accurate function and gradient estimates with a Lyapunov function that accounts for all the recent objective values. For trust-region methods, the actual reduction in the acceptance ratio could analogously be measured from the largest recent objective value. Developing such line-search and trust-region extensions, and determining whether non-monotonicity yields similar practical gains in those settings, are promising directions for future work.

References

[1] C. Audet and W. Hare, *Derivative-Free and Blackbox Optimization*, Springer, New York, 2017.

[2] C. P. Avelino, J. M. Moguerza, A. Olivares, and F. J. Prieto, *Combining and scaling descent and negative curvature directions*, Math. Program., 128 (2011), pp. 285–319.

[3] A. S. Berahas, R. H. Byrd, and J. Nocedal, *Derivative-free optimization of noisy functions via quasi-Newton methods*, SIAM J. Optim., 29 (2019), pp. 965–993.

[4] A. S. Berahas, L. Cao, and K. Scheinberg, *Global convergence rate analysis of a generic line search algorithm with noise*, SIAM J. Optim., 31 (2021), pp. 1489–1518.

[5] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg, *Convergence rate analysis of a stochastic trust-region method via supermartingales*, INFORMS J. Optim., 1 (2019), pp. 92–119.

[6] L. Cao, A. S. Berahas, and K. Scheinberg, *First-and second-order high probability complexity bounds for trust-region methods with noisy oracles*, Math. Program., 207 (2024), pp. 55–106.

[7] C. Cartis, P. R. Sampaio, and P. L. Toint, *Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization*, Optim., 64 (2015), pp. 1349–1361.

[8] R. Chamberlain, M. Powell, C. Lemarechal, and H. Pedersen, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, in Algorithms for Constrained Minimization of Smooth Nonlinear Functions, Springer, New York, 2009, pp. 1–17.

[9] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia, 2009.

[10] Y.-H. Dai, *On the nonmonotone line search*, J. Optim. Theory Appl., 112 (2002), pp. 315–330.

[11] J. E. Dennis Jr and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[12] A. Ding, F. Rinaldi, and L. N. Vicente, *Sequential test sampling for stochastic derivative-free optimization*, ISE Technical Report 25T-016, Lehigh University, (2025).

[13] M. Diniz-Ehrhardt, J. Martínez, and M. Raydán, *A derivative-free nonmonotone line-search technique for unconstrained optimization*, J. Comput. Appl. Math., 219 (2008), pp. 383–397.

[14] U. M. García-Palomares, *Non-monotone derivative-free algorithm for solving optimization models with linear constraints: Extensions for solving nonlinearly constrained models via exact penalty methods*, Trans. Oper. Res., 28 (2020), pp. 599–625.

[15] U. M. García-Palomares, I. J. García-Urrea, and P. S. Rodríguez-Hernández, *On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization*, Optim. Methods Softw., 28 (2013), pp. 1233–1261.

[16] R. Garmanjani, *NoMoDS: A non-monotone directional direct-search algorithm with worst-case complexity analysis.*

[17] T. Giovannelli, O. Sohab, and L. N. Vicente, *The limitation of neural nets for approximation and optimization*, J. Global Optim., 94 (2024), pp. 1–30.

[18] N. I. Gould, D. Orban, and P. L. Toint, *CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization*, Comput. Optim. Appl., 60 (2015), pp. 545–557.

[19] S. Gratton, C. W. Royer, and L. N. Vicente, *A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds.*, Math. Program., 179 (2020), pp. 195–222.

[20] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, *Direct search based on probabilistic descent*, SIAM J. Optim., 25 (2015), pp. 1515–1541.

[21] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[22] L. Grippo and F. Rinaldi, *A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations*, Comput. Optim. Appl., 60 (2015), pp. 1–33.

[23] J. Larson, M. Menickelly, and J. Shi, *A novel noise-aware classical optimizer for variational quantum algorithms*, INFORMS J. Comput., 37 (2025), pp. 63–85.

[24] J. Larson, M. Menickelly, and S. M. Wild, *Derivative-free optimization methods*, Acta Numer., 28 (2019), pp. 287–404.

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

[26] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Science & Business Media, New York, 2013.

[27] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2006.

[28] F. Rinaldi, L. N. Vicente, and D. Zeffiro, *Stochastic trust-region and direct-search methods: A weak tail bound condition and reduced sample sizing*, SIAM J. Optim., 34 (2024), pp. 2067–2092.

[29] H. Zhang and W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim., 14 (2004), pp. 1043–1056.

Appendix

A An alternative merit function for Algorithm 2

This appendix presents the alternative merit function Ψ_k mentioned in Section 2. The first lemma below shows that the maximum function values of M successful iterates is monotonically increasing when the index is shifted to more recent iterates. This property is helpful later to analyze the merit function.

Lemma A.1 (Monotonicity) *In Algorithm 2, for each $j \geq 0$, we have*

$$\max\left(f(X^{j+M-1}_{succ,k}), \ldots, f(X^{j+1}_{succ,k}), f(X^{j}_{succ,k})\right) \leq \max\left(f(X^{j+M}_{succ,k}), \ldots, f(X^{j+2}_{succ,k}), f(X^{j+1}_{succ,k})\right).$$

Proof. Fix any integer $j \geq 0$. We consider two cases.

First, suppose that there are at least $j+1$ successful iterations from iteration 0 through iteration k. Then the $(j+1)$-st successful iteration in reverse order exists, and the successful-iteration condition gives

$$f(X^{j}_{\mathrm{succ},k}) \leq \max_{1\leq i\leq M} f(X^{j+i}_{\mathrm{succ},k}) - c\left(\Delta^{j+1}_{\mathrm{succ},k}\right)^2 \leq \max_{1\leq i\leq M} f(X^{j+i}_{\mathrm{succ},k}). \tag{A.1}$$

Moreover,

$$\max_{1\leq i\leq M-1} f(X^{j+i}_{\mathrm{succ},k}) \leq \max_{1\leq i\leq M} f(X^{j+i}_{\mathrm{succ},k}). \tag{A.2}$$

Combining (A.1) and (A.2), we obtain the desired result.

Now suppose that there are at most j successful iterations from iteration 0 through iteration k. By the convention used to define the reverse sequence beyond the available successful iterations,

$$X^{j+i}_{\mathrm{succ},k} = X_0, \qquad i = 1, \ldots, M.$$

Consequently,

$$\max_{0\leq i\leq M-1} f(X^{j+i}_{\mathrm{succ},k}) = f(X_0) = \max_{1\leq i\leq M} f(X^{j+i}_{\mathrm{succ},k}).$$

□

In the second lemma below, we show a lower bound for the smallest stepsize among the previous M successful iterations in term of the current stepsize.

Lemma A.2 *In Algorithm 2, we have*

$$\min_{0\leq i\leq M-1}\left\{\left(\Delta^{i}_{succ,k}\right)^2\right\} \geq \frac{\Delta_k^2}{\gamma^{2M-2}}.$$

Proof. There are at most $M-1$ successes between the past $M-1$ successful iterations in reverse order at iteration k and the k-th iteration. Recall that if the i-th success in reverse order at iteration k does not exist, then $\Delta^{i}_{succ,k} = \Delta^{i-1}_{succ,k}$. Therefore, from the stepsize mechanism of Algorithm 2, we have for each $0 \leq i \leq M-1$

$$\Delta_k^2 \leq \gamma^{2M-2}\left(\Delta^{i}_{succ,k}\right)^2.$$

Therefore, we have

$$\Delta_k^2 \leq \gamma^{2M-2} \min_{0\leq i\leq M-1} \left\{ \left(\Delta_{succ,k}^i\right)^2 \right\},$$

which proves the result after rearranging.

□

Let us denote the following merit function for Algorithm 2:

$$\begin{aligned}\Psi_k &= \sum_{i=0}^{M-1} \max\left(f(X_{succ,k}^{i+M-1}), \ldots, f(X_{succ,k}^{i+1}), f(X_{succ,k}^{i})\right) - Mf^* + \eta\Delta_k^2 \\ &= \psi_k - Mf^* + \eta\Delta_k^2,\end{aligned}$$

where $\eta > 0$ is a constant. While Ψ_k does not need to include the correcting parameters c_i, $i = 1, \ldots, M-1$ as in Φ_k, it instead has the sum of M prior maximum function values terms. As a result, the decrease of Ψ_k depends on M sufficient decrease conditions, as demonstrated in the following lemma.

Lemma A.3 *In Algorithm 2, the $M-1$ last satisfactions of the sufficient decrease condition and the satisfaction of the current sufficient decrease condition (3.6) at iteration k imply that*

$$\psi_k - \psi_{k+1} \geq c \min_{0\leq i\leq M-1} \left\{ \left(\Delta_{succ,k}^i\right)^2 \right\}.$$

Proof. For $0 \leq j \leq M-2$, because of the $j+1$-th satisfied sufficient decrease condition in reverse order at iteration k, it follows that

$$\max\left(f(X_{succ,k}^{j+M}), \ldots, f(X_{succ,k}^{j+2}), f(X_{succ,k}^{j+1})\right) - f(X_{succ,k}^{j}) \geq c\left(\Delta_{succ,k}^{j+1}\right)^2.$$

For ease of notation, let us denote that $B_k = \max\left(f(X_{succ,k}^{2M-2}), \ldots, f(X_{succ,k}^{M-1})\right)$. Together with Lemma A.1, one has that for each $0 \leq j \leq M-2$

$$\begin{aligned}&B_k - f(X_{succ,k}^{j}) \\ \geq &\max\left(f(X_{succ,k}^{j+M}), \ldots, f(X_{succ,k}^{j+2}), f(X_{succ,k}^{j+1})\right) - f(X_{succ,k}^{j}) \geq c\left(\Delta_{succ,k}^{j+1}\right)^2. \end{aligned} \tag{A.3}$$

The satisfaction of the current sufficient decrease condition gives

$$\max\left(f(X_{succ,k}^{M-1}), \ldots, f(X_{succ,k}^{1}), f(X_{succ,k}^{0})\right) - f(X_{succ,k+1}^{0}) \geq c\left(\Delta_{succ,k}^{0}\right)^2.$$

Then Lemma A.1 implies that

$$\begin{aligned}&B_k - f(X_{succ,k+1}^{0}) \\ \geq &\max\left(f(X_{succ,k}^{M-1}), \ldots, f(X_{succ,k}^{1}), f(X_{succ,k}^{0})\right) - f(X_{succ,k+1}^{0}) \geq c\left(\Delta_{succ,k}^{0}\right)^2. \end{aligned} \tag{A.4}$$

From (A.3) and (A.4), we obtain

$$\psi_k - \psi_{k+1} = \sum_{i=0}^{M-1} \max\left(f(X_{succ,k}^{i+M-1}), \ldots, f(X_{succ,k}^{i})\right) - \sum_{i=0}^{M-1} \max\left(f(X_{succ,k+1}^{i+M-1}), \ldots, f(X_{succ,k+1}^{i})\right)$$

$$
\begin{aligned}
&= \sum_{i=0}^{M-1} \max\left(f(X_{succ,k}^{i+M-1}), \ldots, f(X_{succ,k}^{i})\right) - \sum_{i=1}^{M-1} \max\left(f(X_{succ,k}^{i+M-2}), \ldots, f(X_{succ,k}^{i-1})\right) \\
&\quad - \max\left(f(X_{succ,k}^{M-2}), \ldots, f(X_{succ,k}^{0}), f(X_{succ,k+1}^{0})\right) \\
&= B_k - \max\left(f(X_{succ,k}^{M-2}), \ldots, f(X_{succ,k}^{0}), f(X_{succ,k+1}^{0})\right) \\
&= \min\left(B_k - f(X_{succ,k}^{M-2}), \ldots, B_k - f(X_{succ,k}^{0}), B_k - f(X_{succ,k+1}^{0})\right) \\
&\geq c \min_{0\leq i\leq M-1} \left\{\left(\Delta_{succ,k}^{i}\right)^2\right\}.
\end{aligned}
$$

□

The final lemma specifies the parameter η and shows a deterministic decrease property for the merit function as follows.

Lemma A.4 *Let $N_{succ,k}$ denote the number of successes before iteration k. Suppose that $N_{succ,k} \geq M-1$. Let $\eta = \frac{c}{\gamma^{2M-2}(\gamma^2-\theta^2)}$. There exists $\nu = \frac{c(1-\theta^2)}{\gamma^{2M-2}(\gamma^2-\theta^2)}$ such that, in Algorithm 2, we have*

$$\Psi_k - \Psi_{k+1} \geq \nu \Delta_k^2.$$

Proof. Let $A_k = \mathbf{1}\{\text{iteration } k \text{ is successful}\}$. Since $N_{succ,k} \geq M-1$, it follows from Lemma A.3 and then Lemma A.2 that $A_k = 1$ implies

$$
\begin{aligned}
\Psi_k - \Psi_{k+1} &= \psi_k - \psi_{k+1} + \eta\Delta_k^2(1-\gamma^2) \\
&\geq c \min_{0\leq i\leq M-1} \left\{\left(\Delta_{succ,k}^{i}\right)^2\right\} + \eta\Delta_k^2(1-\gamma^2) \\
&\geq \Delta_k^2 \left(\frac{c}{\gamma^{2M-2}} + \eta(1-\gamma^2)\right) \\
&= \nu\Delta_k^2.
\end{aligned}
$$

Also note that $A_k = 0$ implies that

$$
\begin{aligned}
\Psi_k - \Psi_{k+1} &= \psi_k - \psi_{k+1} + \eta\Delta_k^2(1-\theta^2) \\
&= \eta\Delta_k^2(1-\theta^2) \\
&= \nu\Delta_k^2.
\end{aligned}
$$

Therefore, we conclude that

$$
\begin{aligned}
\Psi_k - \Psi_{k+1} &= (\Psi_k - \Psi_{k+1})A_k + (\Psi_k - \Psi_{k+1})(1-A_k) \\
&\geq \nu\Delta_k^2 A_k + \nu\Delta_k^2(1-A_k) \\
&= \nu\Delta_k^2.
\end{aligned}
$$

□

Finally, from this descent lemma, one can apply similar stopping time arguments as in and Theorem 3.9 to obtain a bound on $E(T_\epsilon)$.

B Sampling complexity

In this appendix, we demonstrate how the tail-bound error in Assumption 4.1 can be satisfied for a standard setting where the stochastic function oracle is estimated from a number of samples. In more detail, we suppose that at iteration k, the oracle $\widehat{f}_k(x)$ estimates the function value at point x using N_k independent samples as follows

$$\widehat{f}_k(x) = \frac{1}{N_k} \sum_{j=1}^{N_k} F(x, \zeta_{k,j}),$$

where $E[F(x,\zeta)] = f(x)$ and $F(x,\zeta) - f(x)$ is sub-Gaussian with variance proxy σ^2. This estimation is independent of the filtration $F_{k+1/2}$.

Lemma B.1 (A sufficient batch size) *Let us compute the stochastic oracles in Algorithm 3 as described. The second condition of Assumption 4.1 holds if*

$$N_k \geq \frac{\sigma^2 \left(M+1\right)^2}{\beta^2 \Delta_k^4}. \tag{B.1}$$

Thus $N_k = \mathcal{O}(\Delta_k^{-4})$ is sufficient to guarantee the satisfaction of this condition.

Proof. Let us denote $a_0 = F(X_k), a_i = F(X^i_{succ,k})$ for $i \in \{1, \ldots, M-1\}, a_d = F(X^d_k)$, and e_0, e_i's, e_d be the corresponding errors when estimating these quantities. Hence

$$\varepsilon_k = G_k - H_k = [\max_i(a_i + e_i) - (a_d + e_d)] - [\max_i a_i - a_d] = \max_i(a_i + e_i) - \max_i(a_i) - e_d.$$

And we have

$$|\varepsilon_k| \leq |\max_i(a_i + e_i) - \max_i(a_i)| + |e_d| \leq \max_i |e_i| + |e_d| < \sum_i |e_i| + |e_d|.$$

As the e_0, e_i's, e_d are estimated with N_k independent samples and each sample has Gaussian error with variance proxy σ^2, the errors e_0, e_i's, e_d are sub-Gaussian with variance proxy σ^2/N_k. This implies the bound

$$E\left[|e_i| \mid \mathcal{F}_{k+1/2}\right] \leq \sqrt{E[e_i^2 \mid \mathcal{F}_{k+1/2}]} \leq \frac{\sigma}{\sqrt{N_k}}. \tag{B.2}$$

As a result

$$E[|\varepsilon_k| \mid \mathcal{F}_{k+1/2}] \leq \sum_i E[|e_i| \mid \mathcal{F}_{k+1/2}] + E[|e_d| \mid \mathcal{F}_{k+1/2}] \leq \frac{(M+1)\sigma}{\sqrt{N_k}} \leq \beta \Delta_k^2,$$

where the last inequality follows from the sample size condition. Markov's inequality gives, for every $b > 0$,

$$P\left(|\varepsilon_k| \geq b \mid \mathcal{F}_{k+1/2}\right) \leq \frac{E[|\varepsilon_k| \mid \mathcal{F}_{k+1/2}]}{b} \leq \frac{\beta \Delta_k^2}{b}.$$

Since $\{\varepsilon_k \geq b\} \subseteq \{|\varepsilon_k| \geq b\}$, this proves the second condition of Assumption 4.1. □

C Problem Set Details

In the experiment, we use two problem sets from the CUTEst collection [18]. We describe the problem names and their corresponding dimensions below.

Problems	Dimensions	Problems	Dimensions	Problems	Dimensions
ARGLINA	10	ARGTRIGLS	10	ARWHEAD	100
BDEXP	100	BOXPOWER	10	BROWNAL	10
COSINE	10	CURLY10	100	DIXON3DQ	10
DQRTIC	10	ENGVAL1	100	EXTROSNB	10
FLETBV3M	10	FLETCBV3	10	FLETCHBV	10
FLETCHCR	10	FREUROTH	10	INDEFM	10
MANCINO	10	MOREBV	10	NONCVXU2	10
NONCVXUN	10	NONDIA	10	NONDQUAR	100
PENALTY2	10	POWER	10	QING	5
QUARTC	100	SENSORS	10	SINQUAD	100
SCOSINE	10	SCURLY10	10	SPARSINE	10
SPARSQUR	10	SSBRYBND	10	TRIDIA	10
TRIGON1	10	TOINTGSS	10		

Table C.1: Problem set 1.

Problems	Dimensions	Problems	Dimensions	Problems	Dimensions
ALLINITU	4	BARD	3	BIGGS6	6
BOX3	3	BROYDN7D	10	BRYBND	10
CUBE	2	DENSCHND	3	DENSCHNE	3
DIXMAANA	15	DIXMAANB	15	DIXMAANC	15
DIXMAAND	15	DIXMAANE	15	DIXMAANF	15
DIXMAANG	15	DIXMAANH	15	DIXMAANI	15
DIXMAANJ	15	DIXMAANK	15	DIXMAANL	15
ENGVAL2	3	ERRINROS	25	EXPFIT	2
FMINSURF	16	GROWTHLS	3	GULF	3
HAIRY	2	HATFLDD	3	HATFLDE	3
HEART6LS	6	HEART8LS	8	HELIX	3
HIELOW	3	HIMMELBB	2	HIMMELBG	2
HUMPS	2	KOWOSB	4	LOGHAIRY	2
MARATOSB	2	MEYER3	3	MSQRTALS	4
MSQRTBLS	9	OSBORNEA	5	OSBORNEB	11
PENALTY3	50	SNAIL	2	SPMSRTLS	28
STRATEC	10	VIBRBEAM	8	WATSON	12
WOODS	4	YFITU	3		

Table C.2: Problem set 2.