VRDFON – line search in noisy unconstrained derivative-free optimization

Morteza Kimiaei
Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
email: kimiaeim83@univie.ac.at
WWW: http://www.mat.univie.ac.at/~kimiaei

Abstract. In this paper, a new randomized solver (called VRDFON) for noisy unconstrained derivative-free optimization problems is discussed. Complexity bounds in the presence of noise for nonconvex, convex, and strongly convex functions are studied. Two effective ingredients of VRDFON are an improved derivative-free line search algorithm with many heuristic enhancements and quadratic models in adaptively determined subspaces. VRDFON is more robust and efficient than other state-of-the-art solvers, especially for medium and high dimensions.

Keywords. Noisy derivative-free optimization, heuristic optimization, randomized line search method, complexity bounds, sufficient decrease

2000 AMS Subject Classification: 90C15; 90C30; 90C56.

Acknowledgment. The author acknowledges the financial support of the Doctoral Program *Vienna Graduate School on Computational Optimization* (*VGSCO*) funded by the Austrian Science Foundation under Project No W1260-N35. Thanks also to Giampaolo Liuzzi for preparing a MEX file for calling UOBYQA from Matlab and to Arnold Neumaier for useful discussions.

October 31, 2022

1 Introduction

We consider the problem of finding a minimizer of the unconstrained derivative-free optimization (DFO) problem

$$\min_{x\in\mathbb{R}^n} f(x). \tag{1}$$

Here the smooth real-valued function $f:\mathbb{R}^n\to\mathbb{R}$ is known only by a noisy oracle which, for a given $x\in\mathbb{R}^n$, gives an approximation $\tilde{f}(x)$ to the exact function value $f(x)$, contaminated by noise. This problem is called the **noisy DFO problem**. We denote by $g(x)$ the unknown exact gradient vector of f at x and by $\tilde{g}(x)$ its approximation. The algorithm does not use knowledge of g, the Lipschitz constants of f, the structure of f, or the statistical properties of noise. Noise may be deterministic (caused by modelling, truncation, and/or discretization errors) or stochastic (caused by inaccurate measurements or rounding errors).

To analyze the limit accuracy and the complexity of our algorithm for solving the noisy DFO problem (1), we assume, like other DFO methods, see, e.g., Bergou et al. [9], Gratton et al. [24], and Kimiaei & Neumaier [35], that
(A1) the function f is continuously differentiable on $\mathbb{R}^n$, and its gradient is Lipschitz continuous with Lipschitz constant L,

(A2) the level set $\mathcal{L}(x^0) := \{x \in \mathbb{R}^n \mid f(x) \le f(x^0)\}$ of f at x^0 is compact, and
(A3) the approximation $\tilde{f}(x)$ of f at $x \in \mathbb{R}^n$ satisfies

$$|\tilde{f}(x) - f(x)| \le \omega. \tag{2}$$

In the noiseless case $\omega = 0$, (A3) implies $\tilde{f} = f$. (A2) implies that

$$\widehat{f} := \inf\{f(x) \mid x \in \mathbb{R}^n\} = f(\widehat{x}) > -\infty \tag{3}$$

for any global minimizer $\widehat{x}$ of (1).

Given a positive scaling vector $s \in \mathbb{R}^n$ (fixed in throughout the paper), we define the scaled 2-norm $\|p\|$ of $p \in \mathbb{R}^n$ and the dual norm $\|g\|_*$ of $g \in \mathbb{R}^n$ by

$$\|p\| := \sqrt{\sum p_i^2/s_i^2} \quad \text{and} \quad \|g\|_* := \sqrt{\sum s_i^2 g_i^2}.$$

A **complexity bound** of an algorithm for the noisy DFO problem (1) is an upper bound on the number of function evaluations to find an approximate point x (unknown to us since L and $g(x)$ are unknown) near a local optimizer whose unknown exact gradient norm is below a given fixed threshold $\omega > 0$ (unknown to us but appearing in our complexity bound) and whose function value $f(x)$ is sufficiently small compared to the initial function value $f(x^0)$, i.e.,

$$f(x) \le \sup\{f(y) \mid y \in \mathbb{R}^n, \quad f(y) \le f(x^0), \quad \text{and} \quad \|g(y)\|_* = \mathcal{O}(\sqrt{n\omega L})\}. \tag{4}$$

1.1 Related work

There are many efficient and robust methods for the noisy DFO problem (1), e.g., see the books Audet & Hare [5] and Conn et al. [15]. Larson et al. [36] have discussed these methods and their complexity bounds (if any). These methods are based on line search, direct search, model-based, etc., and are either deterministic or randomized or both. We focus here on

- line search methods, e.g., see [36, Section 2.3.4] (Grippo & Rinaldi [27], Grippo & Sciandrone [28], Lucidi & Sciandrone [38], and Neumaier et al. [41]),
- model-based methods, e.g., see [36, Section 2.2] (Bandeira et al. [6], Buhmann [13], Conn & Toint [16], Gratton et al. [24, 25], Gratton et al. [26], Powell [43, 44], Huyer & Neumaier [31], and Wild et al. [48]),
- randomized methods, e.g., see [36, Section 3.2] (Bandeira et al. [6], Diniz-Ehrhardt et al. [19], Gratton et al. [24, 25], and Van Dyke & Asaki [47]).

A solver with a high number of solved problems is called **robust** and a solver with a low relative cost of function evaluations is called **efficient**. A solver is competitive if it is both efficient and robust. A number of state-of-the-art DFO solvers is listed in Table 1 to determine which are competitive on the unconstrained `CUTEst` test problems from the collection of Gould et al. [23], two main tools are used: **robustness** (highest number of problems solved) and **efficiency** (lowest relative cost of function evaluations). The data profile of Moré & Wild [40] and the performance profile of Dolan & Moré [20] are used to compare these solvers in terms of robustness and efficiency, respectively. Two good references for studying the efficiency and robustness of these solvers are Rios & Sahinidis [45] and Kimiaei & Neumaier [35].

solver	model-based	line search	direct search	evolution strategy	deterministic	randomized	Reference
BCDFO	+	−	−	−	+	−	Gratton et al. [26]
UOBYQA	+	−	−	−	+	−	Powell [43]
NEWUOA	+	−	−	−	+	−	Powell [44]
SnobFit	+	−	−	−	+	−	Huyer & Neumaier [31]
GRID	+	−	−	−	+	−	Elster & Neumaier [22]
subUOBYQA	+	−	−	−	+	−	present paper
subNEWUOA	+	−	−	−	+	−	present paper
VRBBO	−	+	−	−	+	+	Kimiaei & Neumaier [35]
SDBOX	−	+	−	−	+	−	Lucidi & Sciandrone [38]
FMINUNC	−	+	−	−	+	−	Matlab Optimization Toolbox
DSPFD	−	−	+	−	−	+	Gratton et al. [24]
BFO	−	−	+	−	+	+	Porcelli & Toint [42]
NMSMAX	−	−	+	−	+	−	Higham [29]
subNMSMAX	−	−	+	−	+	−	present paper
MCS	+	−	+	−	+	−	Huyer & Neumaier [30]
NOMAD	+	−	+	−	+	−	Abramson et al. [1]
CMAES	−	−	−	+	−	+	Auger & Hansen [4]
LMMAES	−	−	−	+	−	+	Loshchilov et al. [37]
fMAES	−	−	−	+	−	+	Beyer [10]
BiPopMAES	−	−	−	+	−	+	Beyer & Sendhoff [11]

Table 1: DFO solvers needed in this paper. `subUOBYQA`, `subNEWUOA`, and `subNMSMAX` are, respectively, `UOBYQA`, `NEWUOA`, and `NMSMAX` in a random subspace to handle problems in medium and high dimensions. We denote by `NOMAD1` the model-based version of `NOMAD` and by `NOMAD2` the model-free version of it.

1.2 Known limit accuracy and complexity bounds

In this section, we discuss the achievable limit accuracy and complexity bounds of several well-known noisy DFO methods under standard assumptions (A1)–(A3).

For the noiseless case, see LARSON et al. [36, Table 8.1] and KIMIAEI & NEUMAIER [35, Tables 1–3] for a summary of known results on worst case complexity and corresponding references. To obtain $\|g(x)\|_* \le \varepsilon$ (under the assumptions (A1) and (A2)), one needs
- $\mathcal{O}(\varepsilon^{-2})$ function evaluations for the general case,
- $\mathcal{O}(\varepsilon^{-1})$ function evaluations for the convex case,
- $\mathcal{O}(\log \varepsilon^{-1})$ function evaluations for the strongly convex case. In all cases, the factors are ignored. Randomized algorithms typically have complexity bounds that are a factor n better than those of deterministic algorithms, see [6].

In the presence of noise, the limit accuracy and complexity results of some algorithms have been investigated. We summarized only the results of line search based in Table 2. Other useful references for complexity results of stochastic DFO methods are CHEN [14], DZAHINI [21], and BLANCHET et al. [12].

Table 2: Known limit accuracy and complexity noisy DFO methods regardless of L and n. As stated in the introduction, $\tilde{g}(x)$ stands for the estimated gradient at x and $\widehat{f}$ is the function value at any global minimizer $\widehat{x}$.

type of noise	theoretical result
deterministic	nonconvex: $\|g\|_* = \mathcal{O}(\sqrt{\omega})$
assumptions:	(A1)–(A3)
reference	LUCIDI & SCIANDRONE [38] and ELSTER & NEUMAIER [22]
deterministic	strongly convex: $f - \widehat{f} = \mathcal{O}(\omega)$
assumptions:	(A1)–(A3)
reference:	BERAHAS et al. [7]
stochastic	nonconvex: $\mathcal{O}(\varepsilon^{-2})$ with $\mathbf{E}(\|g\|_*) \le \varepsilon$
	convex: $\mathcal{O}(\varepsilon^{-1})$ with $\mathbf{E}(\|g\|_*) \le \varepsilon$, $\mathbf{E}(f - \widehat{f}) \le \varepsilon$
	strongly convex: $\mathcal{O}(\log \varepsilon^{-1})$ with $\mathbf{E}(\|g\|_*) \le \varepsilon$, $\mathbf{E}(f - \widehat{f}) \le \varepsilon$
assumptions:	(A1)–(A3) and norm condition:
	$\|\tilde{g}(x) - g(x)\| \le \theta \|g(x)\|$ for some $0 < \theta < 1$
reference:	BERAHAS et al. [8]

2 An overview of our method

We propose a new randomized solver for noisy unconstrained DFO problems – called **Vienna noisy randomized derivative-free optimization** (`VRDFON`). Following the classifications of LARSON et al. [36] and RIOS & SAHINIDIS [45], our new solver `VRDFON` is a local model-based randomized solver.

2.1 Difference between VRDFON and other solvers

2.1.1 Algorithmic with new practical enhancements

VRDFON is an adaptation of our recent solver VRBBO (Kimiaei & Neumaier [35]) to the noisy case, while using an improved version of the multi-line search algorithm of VRBBO. VRDFON uses only one of the practical enhancements of VRBBO, namely random subspace directions.

VRDFON repeatedly calls a decrease search called DS, which uses a multi-line search algorithm called MLS that is likely to reduce inexact function values.

Unlike VRBBO, SDBOX, and FMINUNC, an implemented version of VRDFON uses many new practical enhancements, which are discussed in Section 6.1:

• An implemented version of MLS uses heuristics to find and update step sizes in a new way.

• After a few calls to MLS by DS, without decreasing the inexact function value, step sizes may become too small if noise is high. All derivative-free line search methods have this drawback when noise is high. To remedy this, an implemented version of DS changes step sizes heuristically.

• Surrogate quadratic models are constructed in adaptively determined subspaces that can handle medium and large scale problems.

• An implemented version of MLS is performed along several new directions, such as random approximate coordinate, perturbed random, and improved trust region directions.

2.1.2 Complexity results for VRDFON

We prove for VRDFON complexity results with probability arbitrarily close to one for non-convex, convex, and strongly convex functions in the presence of noise. In all cases, the order of ω in our bounds is the same as in Berahas et al. [8]. In contrast to the method of Berahas et al. [8], which uses the norm condition defined in Table 2, our line search does not use the approximate directional derivative $\tilde{g}^T p$ in the line search condition, but $\gamma\alpha^2$ with $0 < \gamma < 1$ because the estimation of the gradient may be inaccurate in the presence of high noise, leading to failure of the line search algorithm. However, we estimate the gradient to generate different heuristic directions in Section 6.1. Therefore, we obtain our complexity bound regardless of the norm condition since the nature of the line search algorithms is different. Our bounds are obtained with high probability unlike the results of Berahas et al. [8], are valid only in expectation.

2.2 Organization

In Section 3 we discuss how to generate scaled random directions. Then, in Section 4, we describe VRDFON using scaled random directions. Complexity results of VRDFON for all cases with a given probability arbitrarily close to one in the presence of noise are proved in Section 5. Section 6 provides a comparison between VRDFON and other state-of-the-art DFO solvers listed in Table 1 on the 549 unconstrained CUTEst test problems from the collection of Gould et al. [23] and makes a recommendation as to which solvers are robust and efficient based on dimension and noise level. It turns out that VRDFON is more robust and efficient, especially for medium and high dimensions.

3 Search direction

In this section, we describe how to generate random directions. Then it is shown that these directions satisfy the two-sided angle condition (defined below) with probability at least half.

We define a standard random direction as a random direction p drawn uniformly **i.i.d.** (independent and identically distributed) in $[-\frac{1}{2}, \frac{1}{2}]^n$. A **scaled random direction** is a standard random direction p scaled by $\gamma_{\rm rd}/\|p\|$, where $0 < \gamma_{\rm rd} < 1$ is a tiny tuning parameter, resulting in $\|p\| = \gamma_{\rm rd}$.

The scaling of the direction p by $\gamma_{\rm rd}$ is the same as the scaling of the direction p by δ in [35, (17)]. Therefore, our scaled random direction is the scaled random direction obtained by [35, (17)]. Thus, the next result holds for any scaling vector $s \in \mathbb{R}^n$ (defined in Section 1.2).

Essential for our complexity bounds is the following result (Proposition 2 in [35]) for the unknown gradient $g(x)$ of $f(x)$ at $x \in \mathbb{R}^n$.

3.1 Proposition. *Any scaled random direction p satisfies the inequality*

$$\Pr(\|g(x)\|_*\|p\| \le 2\sqrt{cn}|g(x)^Tp|) \ge \frac{1}{2} \tag{5}$$

with a positive constant $c \le 12.5$.

The condition (5) is called **two-sided angle condition** because we cannot check whether any scaled random direction is a descent direction or not. Hence, instead of searching along one ray $\alpha > 0$ only, our line search allows to search the line $x + \alpha p$ in both directions ($\alpha \in \mathbb{R}$).

4 `VRDFON`

In this section, `VRDFON` is described with its ingredients and how to work them.

`VRDFON` counts the number $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ of calls to `DS` and the number $t \in \{1, 2, \cdots, T_0\}$ of calls to `MLS`. Here $1 \le T_0 < \infty$ is a tuning parameter. Moreover, `MLS` counts the number $r \in \{1, 2, \cdots, R_m\}$ of scaled random directions. The upper bound ($2 \le R_m < \infty$) of this number is a tuning parameter. To simplify our algorithms, once all tuning parameters are given in line 1 of `VRDFON`, not mentioned as input.

An iteration of `MLS` is called **successful** if at least one reduction of the inexact function value is found and **unsuccessful** otherwise. An iteration of `DS` is called **successful** if `MLS` has at least one successful iteration. An iteration of `VRDFON` is called **successful** if `DS` has at least one successful iteration.

We denote by $x_{\rm best}$ the **overall best point** and by $\tilde{f}_{\rm best} := \tilde{f}(x_{\rm best})$ the **overall best inexact function value** of `VRDFON`, i.e., the final best point and its inexact function value found by `DS`. Indeed, the overall best point is an ε-approximate stationary point of the sequence x^k ($k = 1, 2, 3, \cdots$) after `VRDFON` terminates at a finite number of iterations.

VRDFON initializes the initial step size $\delta_0 := \delta_{\max} > 0$, which is a tuning parameter, and computes the inexact function value $\tilde{f}(x^0)$ at the initial point x^0. In each iteration, VRDFON calls DS to hopefully find at least one reduction of the inexact function value. Once the step size δ_k is below a minimum threshold $0 \le \delta_{\min} < 1$, which is a tuning parameter, VRDFON terminates with the overall best point $x_{\text{best}} := x^{k+1}$ and its inexact function value $\tilde{f}(x_{\text{best}}) := \tilde{f}(x^{k+1})$ in the $(k+1)$th iteration. Otherwise, if DS cannot find any reduction of the inexact function value the step size δ_{k+1} is reduced by a factor of $Q > 1$, which is a tuning parameter; otherwise, δ_{k+1} is δ_k. VRDFON tries to find an ε-approximate stationary point x_{best} that satisfies $\|g(x_{\text{best}})\|_* \le \varepsilon$ for a threshold $\varepsilon = \sqrt{\omega}$ before the condition $\delta_k \le \delta_{\min}$ is satisfied.

DS generates the sequences y^t and $\tilde{f}(y^t)$ $(t = 1, \cdots, T_0)$. First, it initializes two sequences y^t and $\tilde{f}(y^t)$ $(t = 1, \cdots, T_0)$. Subsequently, DS has T_0 calls to MLS to find reductions of the inexact function values. If no reduction is found, then

$$x^{k+1} = y^{T_0} = x^k \quad \text{and} \quad \tilde{f}(x^{k+1}) = \tilde{f}(y^{T_0}) = \tilde{f}(x^k).$$

Otherwise, at least one reduction of the inexact function value is found and so

$$x^{k+1} = y^{T_0} \neq x^k \quad \text{and} \quad \tilde{f}(x^{k+1}) = \tilde{f}(y^{T_0}) < \tilde{f}(x^k).$$

MLS generates the sequences z^r and $\tilde{f}(z^r)$ $(r = 1, \cdots, R_m)$. It initializes the extrapolation step size α_r and the point z_{best} (found in the $(t-1)$th iteration of DS) and its inexact function value $\tilde{f}(z_{\text{best}})$. MLS includes a for loop. In the rth iteration, the scaled random direction p^r is computed. Then, extrapolation along one of $\pm p^r$ may be performed by applying `extrapolate`. If extrapolation cannot be performed, α_r is reduced.

The goal of `extrapolate` is to find at least one reasonable reduction of the inexact function value. It speeds up reaching an approximate minimizer. Indeed, as long as reductions of the inexact function values are found, extrapolation step sizes are expanded and new trial points and their inexact function values are computed. Once, no decrease in the inexact function value is found, `extrapolate` ends.

In line 16 and 19 of Algorithm 1, if the condition $\tilde{f}_{\text{best}} - \tilde{f}_r > \gamma\alpha_r^2$ holds, $\gamma\alpha_r^2$-sufficient gain $(r \in \{1, 2, \cdots, R_m\})$ is found along the search directions p^r. Since there is no reduction of the inexact function value at the last point evaluated by `extrapolate` (line 27), the previous point of this point is accepted by `extrapolate` as the new best point (line 28) whose inexact function value already has been saved in $\tilde{f}_e$. If $\delta_{\min} = 0$ is chosen, for a finite termination of VRDFON, three other stopping tests are needed, see Section 6.3.

Algorithm 1 `VRDFON`**, a randomized method for noisy DFO problems**

Tuning parameters: $Q > 1$ (factor for reducing δ_k), $0 < \gamma_{\rm rd} < 1$ (parameter for scaling random directions), $0 < \gamma < 1$ (parameter for line search), $\gamma_e > 1$ (factor for updating step size inside `MLS`), $\delta_{\max} > 0$ (initial value for δ_k), $0 \le \delta_{\min} \le 1$ (minimum threshold for δ_k), $0 < \eta < \frac{1}{2}$ (parameter for R_m), $R_m := \lceil \log_2 \eta^{-1} \rceil \ge 2$ (number of random direction in each `MLS`), $T_0 \ge 1$ (number of calls to `MLS` by `DS`).

Set $\delta_0 := \delta_{\max}$ and compute $\tilde f(x^0)$;
for $k = 0, 1, 2, \cdots$ **do**
    run $[x^{k+1},\ \tilde f(x^{k+1})] =$`DS`$(\delta_k,\ x^k,\ \tilde f(x^k))$;
    if $\delta_k \le \delta_{\min}$, set $x_{\rm best} := x^{k+1}$ and $\tilde f_{\rm best} := \tilde f(x^{k+1})$; **stop**; **end if**
    if $\tilde f(x^{k+1}) \ge \tilde f(x^k)$ **then**, reduce $\delta_{k+1} := \delta_k / Q$;
    else, set $\delta_{k+1} := \delta_k$;
    end if
end for

---

function $[x^{k+1},\ \tilde f(x^{k+1})] =$ `DS`$(\delta_k,\ x^k,\ \tilde f(x^k))$

---

set $y^0 := x^k$ and $\tilde f(y^0) := \tilde f(x^k)$;
for $t = 1, \cdots, T_0$ **do**, run $[y^t,\ \tilde f(y^t)] =$ `MLS`$(\delta_k,\ y^{t-1},\ \tilde f(y^{t-1})$; **end for**;
set $x^{k+1} := y^{T_0}$ and $\tilde f(x^{k+1}) := \tilde f(y^{T_0})$.

---

function $[y^t,\ \tilde f(y^t)] =$ `MLS`$(\delta_k,\ y^{t-1},\ \tilde f(y^{t-1}))$

Initialize the extrapolation step size $\alpha_1 := \delta_k$, $z_{\rm best} := y^{t-1}$, and $\tilde f_{\rm best} := \tilde f(y^{t-1})$;
for $r = 1, \cdots, R_m$ **do**
    compute the scaled random direction p^r, $z^r := z_{\rm best} + \alpha_r p^r$, and $\tilde f_r = \tilde f(z^r)$;
    if $\tilde f_{\rm best} - \tilde f_r > \gamma \alpha_r^2$, **then** run $[z_{\rm best}, \tilde f_{\rm best}] =$ `extrapolat`$(z_{\rm best}, \tilde f_{\rm best}, \tilde f_r, \alpha_r, p^r)$;
    else
        compute $p^r := -p^r$, $z^r := z_{\rm best} + \alpha_r p^r$, and $\tilde f_r = \tilde f(z^r)$;
        if $\tilde f_{\rm best} - \tilde f_r > \gamma \alpha_r^2$ **then**, run $[z_{\rm best}, \tilde f_{\rm best}] =$ `extrapolat`$(z_{\rm best}, \tilde f_{\rm best}, \tilde f_r, \alpha_r, p^r)$;
        else, reduce the step size to $\alpha_{r+1} := \alpha_r / \gamma_e$;
        end if
    end if
end for
set $y^t := z_{\rm best}$ and $\tilde f(y^t) := \tilde f_{\rm best}$.

---

function $[z_{\rm best}, \tilde f_{\rm best}] =$ `extrapolat`$(z_{\rm best}, \tilde f_{\rm best}, \tilde f, \alpha, p)$

---

while true **do**
    set $\tilde f_e = \tilde f$, expand $\alpha = \gamma_e \alpha$ and compute $z := z_{\rm best} + \alpha p$ and $\tilde f = \tilde f(z)$;
    if $\tilde f_{\rm best} - \tilde f \le \gamma \alpha^2$
            update $\alpha := \alpha / \gamma_e$, $z_{\rm best} := z_{\rm best} + \alpha p$, and $\tilde f_{\rm best} := \tilde f_e$; **stop**.
    end if
end while

5 Complexity bounds for `VRDFON`

In addition to (A1)–(A3) to obtain our complexity results we assume the following assumption:
(A4) Given a minimum threshold $0 < \alpha_{\min} < \infty$ for step sizes in `MLS` and the tuning parameter $\gamma_e > 1$ to update step sizes, the condition $\gamma_e^{1-R_m}\delta_k \ge \alpha_{\min}$ holds.

Under the assumptions (A1)–(A4), this section proves that, with a given probability arbitrarily close to 1, `VRDFON` terminates after at most
- $\mathcal{O}(R_m T_0 L\omega^{-1})$ function evaluations in the general case,
- $\mathcal{O}(\sqrt{n} R_m T_0 L^{3/2}\omega^{-1/2})$ function evaluations in the convex case,
- $\mathcal{O}(nR_m T_0 L\log(\omega^{-1}))$ function evaluations in the strongly convex case

with a point x (unknown to us because gradients and Lipschitz constants are unknown), satisfying (4).

For the choice $R_m = 2$ and $T_0 = 1$, our bounds are better by a factor n than in the deterministic case (see Bandeira et al. [6] and Kimiaei & Neumaier [35]), but numerically these factors should be large to increase the efficiency and robustness of our algorithm. In practice, our algorithm has better numerical performance if R_m and T_0 are chosen such that $R_m T_0 \sim n$, with the same complexity bound as in the deterministic case.

In contrast to the method of Berahas et al. [8], which uses the norm condition, our line search does not use an approximate directional derivative $\tilde{g}^T p$ in the line search condition, since the estimate of the gradient may be inaccurate in the presence of high noise, leading to the failure of the line search algorithm, but $\gamma\alpha^2$. Therefore, we are not interested in obtaining our complexity bound under the norm condition defined in Table 2. We will only use the estimated gradient to generate some heuristic directions in Section 6.1. In all cases, the order of ω in our bounds is the same as in [8], although the nature of the line search algorithms is different and our bounds are obtained with high probability, unlike the results of Berahas et al. [8], are valid only in expectation.

The following result generalizes Proposition 1 in [35]. It is shown that if the rth iteration of `MLS` is unsuccessful, a useful bound for the directional derivative can be found. In this case, no $\gamma\alpha_r^2$-sufficient gain ($r \in \{1, 2, \cdots, R_m\}$) is found along the search directions $\pm p^r$.

5.1 Proposition. *Let $\{z^r\}$ ($r = 1, 2, \cdots, R_m$) be the sequence generated by `MLS` in the $(k+1)$th iteration of `VRDFON`. Moreover, suppose that (A1)–(A3) hold and $0 < \gamma < 1$. Then, for all $z^r, p^r \in \mathbb{R}^n$, at least one of the following holds:*
(i) $\tilde{f}(z_{\text{best}} + \alpha_r p^r) < \tilde{f}(z^r) - \gamma\alpha_r^2$,
(ii) $\tilde{f}(z_{\text{best}} + \alpha_r p^r) > \tilde{f}(z^r) + \gamma\alpha_r^2$ *and* $\tilde{f}(z_{\text{best}} - \alpha_r p^r) < \tilde{f}(z^r) - \gamma\alpha_r^2$,
(iii) $|g(z^r)^T p^r| \le \gamma\alpha_r + 2\omega/\alpha_r + \dfrac{1}{2}L\alpha_r\|p^r\|^2$.
Here, in the kth iteration of `VRDFON`, z_{best} is the best point found by `MLS`.

Proof. (A1) results in

$$\alpha_r g(z^r)^T p^r - \frac{1}{2}L\alpha_r^2\|p^r\|^2 \le f(z_{\text{best}} + \alpha_r p^r) - f(z^r) \le \alpha_r g(z^r)^T p^r + \frac{1}{2}L\alpha_r^2\|p^r\|^2. \tag{6}$$

We assume that (iii) is violated, so that

$$|g(z^r)^T p^r| > \gamma\alpha_r + 2\omega/\alpha_r + \frac{1}{2}L\alpha_r\|p^r\|^2. \tag{7}$$

We consider the proof in the two cases:
CASE 1. If $g(z^r)^T p^r \le 0$, then from (2) and (7) we get

$$\begin{aligned}
\tilde{f}(z_{\text{best}} + \alpha_r p^r) - \tilde{f}(z^r) &\le f(z_{\text{best}} + \alpha_r p^r) - f(z^r) + 2\omega \\
&\le \alpha_r g(z^r)^T p^r + \frac{1}{2}L\alpha_r^2\|p^r\|^2 + 2\omega \\
&= -\alpha_r|g(z^r)^T p^r| + \frac{1}{2}L\alpha_r^2\|p^r\|^2 + 2\omega < -\gamma\alpha_r^2,
\end{aligned} \tag{8}$$

meaning that (i) must hold.
CASE 2. If $g(z^r)^T p^r \ge 0$, then from (2) and (7) we get

$$\begin{aligned}
\tilde{f}(z_{\text{best}} - \alpha_r p^r) - \tilde{f}(z^r) &\le f(z_{\text{best}} - \alpha_r p^r) - f(z^r) + 2\omega \\
&\le g(z^r)^T(-\alpha_r p^r) + \frac{1}{2}L\alpha_r^2\|p^r\|^2 + 2\omega \\
&= -\alpha_r|g(z^r)^T p^r| + \frac{1}{2}L\alpha_r^2\|p^r\|^2 + 2\omega < -\gamma\alpha_r^2,
\end{aligned} \tag{9}$$

Therefore, the second inequality in (ii) holds. By (2), (6), and (7) the first half

$$\begin{aligned}
\tilde{f}(z_{\text{best}} + \alpha_r p^r) - \tilde{f}(z^r) &\ge f(z_{\text{best}} + \alpha_r p^r) - f(z^r) - 2\omega \\
&\ge \alpha_r g(z^r)^T p^r - \frac{1}{2}L\alpha_r^2\|p^r\|^2 - 2\omega > \gamma\alpha_r^2
\end{aligned}$$

must be satisfied. Hence the first inequality in (ii) holds. □

As discussed earlier, `VRDFON` has $1 \le K < \infty$ calls to `DS` and `DS` has $1 \le T_0 < \infty$ calls to `MLS`. Hence, `VRDFON` has KT_0 calls to `MLS`. As defined in `VRDFON`, $R_m = \lceil \log_2 \eta^{-1} \rceil$ for a given $0 < \eta < \frac{1}{2}$ is the number of random directions used by `MLS`. For given $1 \le T_0 < \infty$ and $1 \le K < \infty$, defined by

$$R_d := T_0 R_m = T_0\lceil \log_2 \eta^{-1} \rceil \ge \log_2 \eta^{-T_0} \tag{10}$$

is the number of random directions used by `DS` and defined by

$$R_v := KR_d = KT_0R_m = KT_0\lceil \log_2 \eta^{-1} \rceil \ge \log_2 \eta^{-KT_0} \tag{11}$$

is the number of random directions used by `VRDFON`.

Assuming (A1)–(A4), we find
• for `MLS` at least one point whose unknown gradient norm is below a constant bound (see (13)) with probability $\ge 1 - \eta > \frac{1}{2}$ (see Theorem 5.2);
• for `DS` at least one point whose unknown gradient norm is below a constant bound (see (17)) with probability $\ge 1 - 2^{-R_d} = 1 - 2^{-T_0 R_m} \ge 1 - \eta^{T_0} \ge 1 - \eta > \frac{1}{2}$, (see Theorem 5.3);
• for `VRDFON` at least one point whose unknown gradient norm is below a constant bound

(see (22)) with probability $\geq 1-2^{-R_v} = 1-2^{-KT_0R_m} \geq 1-\eta^{KT_0} \geq 1-\eta > \frac{1}{2}$ (see Theorem 5.4).

The following result is a generalization of Theorem 1 in [35] for `MLS` to the noisy case in the $(k+1)$th iteration of `VRDFON`. As discussed earlier, `MLS` uses R_m scaled random directions p^r $(r = 1, 2, \cdots, R_m)$. If $\alpha_{\min} > 0$, in the worst case, it is proved that one of the following holds:
(i) If at least on iteration $(r' \in \{1, 2, \cdots, R_m\})$ of `MLS` is successful, a minimum reduction in the inexact function value is found when $-p^{r'}$ is tried.
(ii) An upper bound on the unknown gradient norm of at least one $z^{r''}$ $(r'' \in \{1, 2, \cdots, R_m\})$ of the points generated by the unsuccessful iterations of `MLS` is found with a given probability arbitrarily close to one. In fact, it is not clear to us which point, since the gradients and Lipschitz constants are not available.

5.2 Theorem. *Assume that (A1)–(A4) hold,* $0 < \eta < \frac{1}{2}$, $0 < \gamma_{\text{rd}} < 1$, $\gamma_e > 1$, *and* $0 < \gamma < 1$. *Moreover, define*

$$\overline{L} := 2\gamma/\gamma_{\text{rd}} + L\gamma_{\text{rd}} \tag{12}$$

and let $\{z^r\}$ $(r = 1, 2, \cdots, R_m)$ *be the sequence generated by* `MLS` *in the* $(k+1)$*th iteration of* `VRDFON`*. Then one of the following happens:*
(i) If at least `MLS` *has a successful iteration,* $r' \in \{1, 2, \cdots, R_m\}$, *then it decreases the inexact function value by at least* $\gamma\alpha_{r'}^2$.
(ii) If `MLS` *has no successful iteration, then at least one* $z^{r''}$ $(1 \leq r'' \leq R_m)$ *of the points evaluated by the unsuccessful iterations of* `MLS`*, with the probability at least* $1-\eta$*, has an unknown gradient* $g(z^{r''})$ *satisfying*

$$\|g(z^{r''})\|_* \leq \sqrt{cn}\,\Gamma(\delta_k) \quad \text{with } \Gamma(\delta_k) := \overline{L}\delta_k + \gamma_e^{R_m-1}\frac{4\omega}{\gamma_{\text{rd}}\delta_k} \tag{13}$$

for a given $0 < \eta < \frac{1}{2}$. *Here c comes from Proposition 3.1,* δ_k *is fixed in* `MLS`*, independent of* r*, and is updated outside* `MLS`*.*

Proof. Let $\mathcal{R} := \{1, \ldots, R_m\}$. We denote by p^r the rth scaled random search direction, by z^r the rth point, and by $\alpha_r = \gamma_e^{1-r}\delta_k \geq \alpha_{\min}$ the rth step size from (A4).

(i) Let $r' \in \{1, 2, \cdots, R_m\}$. The worst case requires $2R_m + 1$ function evaluations and assumes that the r'th iteration of `MLS` is successful and the other iterations are unsuccessful. In the unsuccessful iterations, two function values are computed along the directions $\pm p^r$ $(r \in \mathcal{R} \setminus \{r'\})$, but in the r'th iteration which is successful, no reduction of the inexact function value is found when $p^{r'}$ is attempted, while reduction of the inexact function value is found when $-p^{r'}$ is attempted. Therefore, an extrapolation step along $-p^{r'}$ is performed with at most two additional function evaluations and the $\gamma\alpha_{r'}^2$-sufficient gain. Consequently, (i) is verified.

(ii) Suppose that $\tilde{f}(z^r)$ does not decrease by more than $\gamma\alpha_r^2$ for all $r \in \mathcal{R}$; all iterations are unsuccessful. Then we define $\Gamma_0(\alpha_r) := \overline{L}\alpha_r + \dfrac{4\omega}{\gamma_{\text{rd}}\alpha_r}$. Since $\Gamma_0(\alpha_r)$ for $\alpha_r > 0$ is a convex function, we obtain for $r \in \mathcal{R}_m$

$$\Gamma_0(\alpha_r) \quad \leq \quad \max\{\Gamma_0(\alpha_1), \Gamma_0(\alpha_R)\} < \overline{L}\alpha_1 + \frac{4\omega}{\gamma_{\text{rd}}\alpha_R}$$

$$= \quad \Gamma(\delta_k) = \overline{L}\delta_k + \gamma_e^{R_m-1}\frac{4\omega}{\gamma_{\rm rd}\delta_k}, \tag{14}$$

where $\alpha_1 := \max\limits_{r\in\mathcal{R}}\{\alpha_r\} = \delta_k > \alpha_R := \min\limits_{r\in\mathcal{R}}\{\alpha_r\} = \gamma_e^{1-R_m}\delta_k$ since $\gamma_e > 1$ and $R_m \geq 2$. Then we obtain from Proposition 5.1 and since $\|p^r\| = \gamma_{\rm rd}$ (due to the definition of the scaled random direction in Section 3), for all $r \in \mathcal{R}$,

$$|g(z^r)^T p^r| \leq \gamma\alpha_r + 2\omega/\alpha_r + \frac{L}{2}\alpha_r\|p^r\|^2 = \gamma\alpha_r + 2\omega/\alpha_r + \frac{L}{2}\gamma_{\rm rd}^2\alpha_r,$$

so that for all $r \in \mathcal{R}$ and from (14), the inequality

$$\begin{aligned} \|g(z^r)\|_* &= \|g(z^r)\|_*\|p^r\|/\gamma_{\rm rd} \leq 2\sqrt{cn}|g(z^r)^T p^r|/\gamma_{\rm rd} \\ &\leq \sqrt{cn}\left(\Big(\frac{2\gamma}{\gamma_{\rm rd}} + L\gamma_{\rm rd}\Big)\alpha_r + \frac{4\omega}{\gamma_{\rm rd}\alpha_r}\right) = \sqrt{cn}\Gamma_0(\alpha_r) < \sqrt{cn}\Gamma(\delta_k) \end{aligned}$$

holds with probability $\frac{1}{2}$ or more according to Proposition 3.1. In other words,

$$\Pr\Big(\|g(z^r)\|_* > \sqrt{cn}\Gamma(\delta_k)\Big) < \frac{1}{2}, \quad \text{for any fixed } r \in \mathcal{R}.$$

Therefore, we find at least one of the gradients $g = g(z^{r''})$ ($r'' \in \mathcal{R}$) such that (13) holds, that is, for a given $0 < \eta < \frac{1}{2}$

$$\begin{aligned} \Pr\Big(\|g\|_* \leq \sqrt{cn}\Gamma(\delta_k)\Big) &= 1 - \prod_{r=1}^{R_m}\Pr\Big(\|g(z^r)\|_* > \sqrt{cn}\Gamma(\delta_k)\Big) \\ &\geq 1 - 2^{-R_m} \geq 1 - \eta. \end{aligned}$$

□

The following result discusses the complexity bound for `DS` in the $(k+1)$th iteration of `VRDFON`. It is proved that either an upper bound on the number of function evaluations is found or an upper bound on the unknown gradient norm of at least one of the points generated by the unsuccessful iterations of `DS` is found with a given probability arbitrarily close to one in the presence of noise; in fact, it is not clear to us which point, since the gradients and Lipschitz constants are not available.

5.3 Theorem. *Suppose that (A1)–(A4) hold and let $f(x^0)$ be the initial value of f. Moreover, let $\{y^t\}$ ($t = 1, 2, \cdots, T_0$) be the sequence generated by* `DS` *in the $(k+1)$th iteration of* `VRDFON` *and let $0 < \eta < \frac{1}{2}$, $0 < \gamma_{\rm rd} < 1$, $\gamma_e > 1$ and $0 < \gamma < 1$. Then:*
(i) The number of successful iterations of `DS` *is bounded by*

$$\overline{\gamma}^{-1}\delta_k^{-2}\Big(f(x^0) - \widehat{f} + 2\omega\Big), \tag{15}$$

where $\overline{\gamma} := \gamma_e^{2(2-R_m)}\gamma > 0$, $\widehat{f}$ is finite by (A1) and (A2), and the step size δ_k is fixed, independent of t, and updated outside `DS`*. Moreover, the number of function evaluations of* `DS` *is bounded by*

$$2R_mT_0 + (2R_m+1)T_0\overline{\gamma}^{-1}\delta_k^{-2}\Big(f(x^0) - \widehat{f} + 2\omega\Big). \tag{16}$$

(ii) Unsuccessful iterations of `DS` *have at least one point $y^{t'}$ $(1 \le t' \le T_0)$, with probability at least $\ge 1 - 2^{-R_d} \ge 1 - \eta > \frac{1}{2}$ for a given $0 < \eta < \frac{1}{2}$, satisfying*

$$\|g(y^{t'})\|_* \le \sqrt{cn}\,\Gamma(\delta_k), \tag{17}$$

where c and $\Gamma(\delta_k)$ come from Proposition 3.1 and Theorem 5.2. Here R_d comes from (10).

Proof. (i) S denotes the index set of successful iterations of `DS`, where each successful iteration is a result of at least one successful iteration of `MLS`. Let $r' \in \{1, 2, \cdots, R_m\}$. As discussed in the proof of Theorem 5.2(i), in the worst case at least the r'th iteration of `MLS` is successful that a result of an extrapolation along $-p^{r'}$. We do not know how many times we can extrapolate $\alpha_{r'}$ to γ_e along the fixed direction $-p^{r'}$, but at least once $\alpha_{r'}$ is expanded by γ_e in an extrapolation and therefore at most $R_m - 1$ times $\alpha_1 = \delta_k$ is reduced by γ_e if we cannot extrapolate along the other scaled random directions and their opposite directions. Therefore, for each $t \in S$, according to the role of updating α_r in lines 13, 20, and 28 of Algorithm 1,

$$\alpha_{r'} \ge \gamma_e \delta_k / \gamma_e^{R_m - 1} = \gamma_e^{2 - R_m} \delta_k$$

in the $(k+1)$th iteration of `VRDFON`. Put $\overline{\gamma} := \gamma_e^{2(2-R_m)} \gamma > 0$. We now find an upper bound on the number of successful iterations and the corresponding function evaluations of `DS`. For all $t \in S$ in the $(k+1)$th iteration of `VRDFON`, we have

$$\tilde{f}(y^{t+1}) - \tilde{f}(y^t) = \tilde{f}(z^{\hat{r}}) - \tilde{f}(z_{\text{best}}) \le -\gamma \alpha_{r'}^2 \le -\overline{\gamma} \delta_k^2,$$

recursively resulting in $\tilde{f}(y^{t+1}) \le \tilde{f}(x^0) - \overline{\gamma}\delta_k^2 \sum_{t \in S} 1 = \tilde{f}(x^0) - \overline{\gamma}\delta_k^2 |S|$. From (2) we conclude that

$$|S| \le \overline{\gamma}^{-1} \delta_k^{-2} \Big(\tilde{f}(x^0) - \tilde{f}(y^{t+1})\Big) \le \overline{\gamma}^{-1} \delta_k^{-2} \Big(f(x^0) - \widehat{f} + 2\omega\Big).$$

Therefore, (15) is valid. The step size δ_k is fixed, independent of t, and updated outside `DS`. As mentioned earlier, `MLS` requires at most $2R_m + 1$ function evaluations in the worst case (using R_m scaled random directions and R_m corresponding opposite directions, all iterations of `MLS` are unsuccessful; however, if the r'th iteration is successful, then no reduction of the inexact function value is found when $p^{r'}$ is attempted, while reduction of the inexact function value is found when $-p^{r'}$ is attempted; a sufficient gain along the last opposite direction $-p^{r'}$ is found. Therefore, an extrapolation with at most two function evaluations is attempted. Therefore, the successful iterations of `DS` use at most

$$(2R_m + 1)\overline{\gamma}^{-1} \delta_k^{-2} \Big(f(x^0) - \widehat{f} + 2\omega\Big)$$

function evaluations.

U denotes the index set of unsuccessful iterations of `DS`. Since $T_0 = |U| + |S|$ and `MLS` uses $2R_m$ function evaluations for each unsuccessful iteration, we conclude that the unsuccessful iterations of `DS` use at most $2R_m|U| \le 2R_m T_0$ function evaluations. Consequently, the number of function evaluations of `DS` is bounded by (16).
(ii) In this case, the unsuccessful iterations of `DS` generate the sequence y^t $(t = 1, \cdots, T_0)$, resulting in, that for at least one $y^{t'}$ $(1 \le t' \le T_0)$ of the evaluated points, with probability $\ge 1 - 2^{-R_d} = 1 - 2^{-T_0 R_m} \ge 1 - \eta^{T_0} \ge 1 - \eta$, $\|g(y^{t'})\|_* \le \sqrt{cn}\,\Gamma(\delta_k)$ holds for a given

$0 < \eta < \frac{1}{2}$. As mentioned in (i), the step size δ_k is fixed, independent of t; hence the bound $\sqrt{cn}\Gamma(\delta_k)$ is fixed for DS. $\square$

The objective function f is convex ($\sigma = 0$) if the condition

$$f(y) \geq f(x) + g(x)^T(y-x) + \frac{1}{2}\sigma\|y-x\| \quad \text{for } x, y \in \mathbb{R}^n \tag{18}$$

holds and is strongly convex ($\sigma > 0$) if (18) holds.

It is proved that an upper bound for the unknown gradient norm of at least one of points generated by the unsuccessful iterations of VRDFON is found for all cases with a given probability arbitrarily close to one in the presence of noise.

5.4 Theorem. *Assume that (A1)–(A4) hold and $\delta_{\max} > 0$, $Q > 1$, $0 < \gamma_{\rm rd} < 1$, $\gamma_e > 1$, $0 < \gamma < 1$, $\overline{\tau} > \underline{\tau} > 0$, and*

$$\underline{\tau}\sqrt{\omega/L} \leq \delta_{\min} \leq \overline{\tau}\sqrt{\omega/L}. \tag{19}$$

Let $\{x^k\}$ $(k = 1, 2, \cdots)$ be the sequence generated by VRDFON. *Then*

$$\delta_\ell = Q^{1-\ell}\delta_{\max} \quad \text{for } \ell \geq 1 \tag{20}$$

and VRDFON *terminates after at most*

$$K := 1 + \Big\lfloor \frac{\log(\delta_{\max}/\delta_{\min})}{\log Q} \Big\rfloor \tag{21}$$

unsuccessful iterations. Then, for a given $0 < \eta < \frac{1}{2}$, VRDFON *finds at least one point $x^{\ell'}$ with probability at least $1 - 2^{-R_v} \geq 1 - \eta$ satisfying*
(i) in the nonconvex case the condition

$$\|g(x^{\ell'})\|_* = \mathcal{O}(\sqrt{nL\omega}); \tag{22}$$

(ii) in the convex case the condition (22) *and*

$$f(x^{\ell'}) - \widehat{f} = \mathcal{O}(r_0\sqrt{nL\omega}), \tag{23}$$

where r_0 is given by

$$r_0 := \sup\Big\{\|x - \widehat{x}\| \mid x \in \mathbb{R}^n, \quad f(x) \leq f(x^0)\Big\} < \infty; \tag{24}$$

(iii) in the strongly convex case the condition (22)*,*

$$f(x^{\ell'}) - \widehat{f} = \frac{\mathcal{O}(nL\omega)}{2\sigma}, \quad \text{and} \quad \|x^{\ell'} - \widehat{x}\| = \frac{\mathcal{O}(\sqrt{nL\omega})}{\sigma}. \tag{25}$$

Here $\widehat{f}$ is finite by (A1) and (A2) and R_v comes from (11)*.*

Proof. (i) Since VRDFON has K unsuccessful iterations, from calls to DS and Theorem 5.3(ii), the condition

$$\|g(x^{\ell'})\|_* \leq \sqrt{cn}\min_{\ell=0:K}\Gamma(\delta_\ell), \tag{26}$$

holds for at least one $x^{\ell'}$ of the evaluated points with probability

$$\geq 1 - 2^{-R_v} = 1 - 2^{-T_0 K R_m} \geq 1 - \eta^{T_0 K} \geq 1 - \eta > \frac{1}{2}$$

for a given $0 < \eta < \frac{1}{2}$. By (21), we have $\delta_K = Q^{1-K}\delta_{\max} \leq \delta_{\min}$, resulting in

$$\begin{aligned}
\Gamma(\delta_K) &= \overline{L}\delta_K + \gamma_e^{R_m-1}\frac{4\omega}{\gamma_{\rm rd}\delta_K} = \overline{L}Q^{1-K}\delta_{\max} + \gamma_e^{R_m-1}Q^{K-1}\frac{4\omega}{\gamma_{\rm rd}\delta_{\max}} \\
&\overset{(21)}{=} \overline{L}\frac{\delta_{\min}}{\delta_{\max}}\delta_{\max} + \gamma_e^{R_m-1}\frac{\delta_{\max}}{\delta_{\min}}\frac{4\omega}{\gamma_{\rm rd}\delta_{\max}} \\
&\overset{(19)}{\leq} (2\gamma/\gamma_{\rm rd} + L\gamma_{\rm rd})\overline{\tau}(\sqrt{\omega/L}) + \gamma_e^{R_m-1}\frac{4\omega}{\gamma_{\rm rd}\underline{\tau}\sqrt{\omega/L}} = \mathcal{O}(\sqrt{L\omega}),
\end{aligned}$$

whose application in (26) leads to (22). Here $\overline{L}$ comes from (12).
(ii) The convexity of f leads to

$$\widehat{f} \geq f_\ell + g(x^\ell)^T(\widehat{x} - x^\ell) \quad \text{for all } \ell \geq 0.$$

From (i), we conclude that for at least one $x^{\ell'}$ of the evaluated points with probability $\geq 1 - 2^{-R_v} > \frac{1}{2}$ the condition

$$f_{\ell'} - \widehat{f} \leq g(x^{\ell'})^T(x^{\ell'} - \widehat{x}) \leq \|g(x^{\ell'})\|_* \|x^{\ell'} - \widehat{x}\| = \mathcal{O}(r_0\sqrt{nL\omega})$$

holds for a given $0 < \eta < \frac{1}{2}$.
(iii) If x is assumed to be fixed, the right-hand side of (18) is a convex quadratic function with respect to y whose gradient in the components vanishes at $y_i = x_i - s_i\sigma^{-1}g_i(x)$ for $i = 1, \cdots, n$, leading to $f(y) \geq f(x) - \dfrac{1}{2\sigma}\|g(x)\|_*^2$. As mentioned earlier, $s \in \mathbb{R}^n$ is a scaling vector here. By applying (22) in this inequality, we obtain at least for one $x^{\ell'}$ of the evaluated points with probability $\geq 1 - 2^{-R_v} > \frac{1}{2}$

$$f_{\ell'} - \widehat{f} \leq \frac{1}{2\sigma}\|g(x^{\ell'})\|_*^2 = \frac{\mathcal{O}(nL\omega)}{2\sigma} \quad \text{for } \ell' \geq 0 \text{ and a given } 0 < \eta < \tfrac{1}{2}.$$

Substituting x for $\widehat{x}$ and y for $x^{\ell'}$ into (18), we get $f_{\ell'} \geq f(\widehat{x}) + \dfrac{\sigma}{2}\|x^{\ell'} - \widehat{x}\|^2$ such that (i) leads to the fact that for at least one $x^{\ell'}$ of the evaluated points with probability $\geq 1 - 2^{-R_v} > \frac{1}{2}$

$$\|x^{\ell'} - \widehat{x}\|^2 \leq \frac{2}{\sigma}(f_{\ell'} - \widehat{f}) \leq \frac{1}{\sigma^2}\|g(x^{\ell'})\|_*^2 = \frac{\mathcal{O}(nL\omega)}{\sigma^2}$$

holds for a given $0 < \eta < \frac{1}{2}$. □

Compared to the results discussed in Section 1.2, the order of ω in the bound (22) is the same as that in the conditions defined in Table 2. The conditions (23) and (25) are the same as those of Berahas et al. [8], except that they are satisfied with high probability.

For ill conditioned problems the Lipschitz constant L is too large and therefore $\delta_{\min} = \sqrt{\omega/L}$ becomes zero. Hence, Algorithm 1 can choose $\delta_{\min} = 0$.

The following result discusses the complexity bound for `VRDFON` for all cases. It is proved that an upper bound on the number of function evaluations used by `VRDFON` is found with a given probability arbitrarily close to one in the presence of noise.

5.5 Theorem. *Let $\{x^k\}$ $(k = 1, 2, \cdots)$ be the sequence generated by* `VRDFON`. *Under the assumptions of Theorem 5.4,* `VRDFON` *terminates after at most*
(i) $\mathcal{O}(R_m T_0 L \omega^{-1})$ function evaluations in the nonconvex case,
(ii) $\mathcal{O}(\sqrt{n} R_m T_0 L^{3/2} \omega^{-1/2})$ function evaluations in the convex case,
(iii) $\mathcal{O}(n R_m T_0 L \log \omega^{-1})$ function evaluations in the strongly convex case.

Proof. Let $N_0 := 1$, $f_\ell = f(x^\ell)$ and denote by N_K the number of function evaluations for the termination of `VRDFON`. Here K comes from (21). In worst case, we terminate `VRDFON` after at most K unsuccessful iterations from calls to `DS`, with K points satisfying (17), and at least one point satisfying (22). Since the gradient and Lipschitz constants are unknown, these points are unknown. As a consequence of this termination, we have $\delta_\ell = Q^{1-\ell}\delta_{\max} \le \delta_{\min}$ for $\ell \ge K$ and

$$\delta_\ell \ge \delta_{\min} \quad \text{for } \ell \in \mathcal{B} := \{1, \cdots, K\}. \tag{27}$$

The condition (27) is used in the proof of (ii) and (iii).
(i) We conclude from (16) and (20)–(21) that

$$\begin{aligned}
N_K &\le 1 + \sum_{\ell=1}^{K} \Big(2T_0 + (2R_m + 1)T_0\overline{\gamma}^{-1}\delta_\ell^{-2}(f(x^0) - \widehat{f} + 2\omega)\Big) \\
&= 1 + 2R_m T_0 K + (2R_m + 1)T_0\overline{\gamma}^{-1}\Big(f(x^0) - \widehat{f} + 2\omega\Big)\sum_{\ell=1}^{K}\delta_\ell^{-2} \\
&= 1 + 2R_m T_0 K + (2R_m + 1)T_0\overline{\gamma}^{-1}\delta_{\max}^{-2}\Big(f(x^0) - \widehat{f} + 2\omega\Big)\sum_{\ell=1}^{K} Q^{2\ell-2} \\
&= 1 + 2R_m T_0 K + (2R_m + 1)T_0\overline{\gamma}^{-1}\delta_{\max}^{-2}\Big(f(x^0) - \widehat{f} + 2\omega\Big)\frac{Q^{2K} - 1}{Q^2 - 1}.
\end{aligned}$$

Here $\overline{\gamma}$ comes from Theorem 5.3 and for a given $0 < \eta < \frac{1}{2}$ $R_m = \lceil \log_2 \eta^{-1} \rceil \ge 2$ comes from Algorithm 1. In this case, using (19) and (21), $R_m T_0 Q^{2K}$ dominates the other terms, resulting in

$$N_K = \mathcal{O}(R_m T_0 Q^{2K}) = \mathcal{O}(R_m T_0 \delta_{\min}^{-2}) = \mathcal{O}(R_m T_0 L \omega^{-1}).$$

(ii) From (A1) and (A2), r_0 is finite. The convexity of f results in

$$\widehat{f} \ge f_\ell + g(x^\ell)^T(\widehat{x} - x^\ell) \quad \text{for all } \ell \ge 0.$$

By Theorem 5.4, for a given $0 < \eta < \frac{1}{2}$, we get with probability $\ge 1 - 2^{-R_v} \ge 1 - \eta > \frac{1}{2}$

$$\begin{aligned}
f_\ell - f_{\ell+1} \le f_\ell - \widehat{f} &\le g(x^\ell)^T(x^\ell - \widehat{x}) \le \|g(x^\ell)\|_* \|x^\ell - \widehat{x}\| && (28) \\
&\le r_0\sqrt{cn}\Big(\overline{L}\delta_\ell + \gamma_e^{R_m - 1}\frac{4\omega}{\gamma_{\mathrm{rd}}\delta_\ell}\Big) \quad \text{for } \ell \in \mathcal{B}. && (29)
\end{aligned}$$

Here $\overline{L}$ comes from (12). We consider the following two cases:
Case 1. The first term $\overline{L}\delta_\ell$ in (29) dominates the second term. Then we have

$$f_\ell - f_{\ell+1} = \mathcal{O}(\overline{L}\sqrt{n}\delta_\ell) = \mathcal{O}(\sqrt{n}L\delta_\ell) \quad \text{for } \ell \in \mathcal{B}\,. \tag{30}$$

Then we define $\mathcal{B}_1 := \{\ell \in \mathcal{B} \mid (30) \text{ holds}\}$.
Case 2. The second term $4\gamma_e^{R_m-1}\omega/(\gamma_{\mathrm{rd}}\delta_\ell)$ in (29) dominates the first term. Then we conclude from (27) that

$$f_\ell - f_{\ell+1} = \mathcal{O}(\sqrt{n}(\omega/\delta_\ell)) = \mathcal{O}(\sqrt{n}(\omega/\delta_{\min})) = \mathcal{O}(\sqrt{n\omega L}) \quad \text{for } \ell \in \mathcal{B}. \tag{31}$$

Then we define $\mathcal{B}_2 = \{\ell \in \mathcal{B} \mid (31) \text{ holds}\}$.
Then we conclude from (19)–(21) that with probability $\geq 1 - 2^{-R_v} > \frac{1}{2}$

$$
\begin{aligned}
\sum_{\ell\in\mathcal{B}} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\delta_\ell^2} &= \sum_{\ell\in\mathcal{B}_1} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\delta_\ell^2} + \sum_{\ell\in\mathcal{B}_2} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\gamma\delta_\ell^2} \\
&\leq \sum_{\ell\in\mathcal{B}_1} \frac{f_\ell - f_{\ell+1} + 2\omega}{\delta_\ell^2} + \sum_{\ell\in\mathcal{B}_2} \frac{f_\ell - f_{\ell+1} + 2\omega}{\delta_\ell^2} \\
&\leq \sum_{\ell\in\mathcal{B}_1} \frac{\mathcal{O}(\sqrt{n}L\delta_\ell) + 2\omega}{\delta_\ell^2} + \sum_{\ell\in\mathcal{B}_2} \frac{\mathcal{O}(\sqrt{n\omega L}) + 2\omega}{\delta_\ell^2} \\
&\leq \sum_{\ell\in\mathcal{B}} \frac{\mathcal{O}(\sqrt{n}L\delta_\ell) + 2\omega}{\delta_\ell^2} + \sum_{\ell\in\mathcal{B}} \frac{\mathcal{O}(\sqrt{n\omega L}) + 2\omega}{\delta_\ell^2} \\
&= \mathcal{O}(\sqrt{n}L)\sum_{\ell\in\mathcal{B}} \delta_\ell^{-1} + \mathcal{O}(\sqrt{n\omega L})\sum_{\ell\in\mathcal{B}} \delta_\ell^{-2} + 4\omega\sum_{\ell\in\mathcal{B}} \delta_\ell^{-2} \\
&= \mathcal{O}(\sqrt{n}L)\sum_{\ell\in\mathcal{B}} Q^{\ell-1} + \mathcal{O}(\sqrt{n\omega L})\sum_{\ell\in\mathcal{B}} Q^{2\ell-2} + 4\omega\sum_{\ell\in\mathcal{B}} Q^{2\ell-2} \\
&= \mathcal{O}(\sqrt{n}LQ^K) + \mathcal{O}(\sqrt{n\omega L}Q^{2K}) + \omega\mathcal{O}(Q^{2K}) \\
&= \mathcal{O}(\sqrt{n}L(\omega/L)^{-1/2}) + \mathcal{O}(\sqrt{n\omega L}(\omega/L)^{-1}) + \omega\mathcal{O}((\omega/L)^{-1}) \\
&= \mathcal{O}(\sqrt{n}L^{3/2}\omega^{-1/2}) + \mathcal{O}(\sqrt{n}L^{3/2}\omega^{-1/2}) + \mathcal{O}(L) = \mathcal{O}(\sqrt{n}L^{3/2}\omega^{-1/2}),
\end{aligned}
$$

holds for a given $0 < \eta < \frac{1}{2}$, so that by (i) and $R_m = \lceil \log_2 \eta^{-1} \rceil \geq 2$

$$N_K \leq 1 + (2R_m + 1)T_0 \sum_{\ell\in\mathcal{B}} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\gamma\delta_\ell^2} = \mathcal{O}\left(\sqrt{n}L^{3/2}R_m T_0 \omega^{-1/2}\right)$$

holds with probability $\geq 1 - 2^{-R_v} \geq 1 - \eta > \frac{1}{2}$ for a given $0 < \eta < \frac{1}{2}$.
(iii) When x is assumed to be fixed, the right hand side of (18) is a convex quadratic function in terms of y whose gradient in the components vanishes at $y_i = x_i - s_i\sigma^{-1}g_i(x)$ for $i = 1, \cdots, n$, resulting in $f(y) \geq f(x) - \frac{1}{2\sigma}\|g(x)\|_*^2$. Here as mentioned earlier $s \in \mathbb{R}^n$ is a scaling vector. By applying (22) in this inequality, for a given $0 < \eta < \frac{1}{2}$, we get with probability $\geq 1 - 2^{-R_v} \geq 1 - \eta > \frac{1}{2}$

$$f_\ell - f_{\ell+1} \leq f_\ell - \widehat{f} \leq \frac{1}{2\sigma}\|g(x^\ell)\|_*^2 \leq \frac{cn}{2\sigma}\left(\overline{L}\delta_\ell + \gamma_e^{R_m-1}\frac{4\omega}{\gamma_{\rm rd}\delta_\ell}\right)^2 \quad \text{for } \ell \in \mathcal{B}. \tag{32}$$

Here $\overline{L}$ comes from (12). We consider the following two cases:
Case 1. The first term $\overline{L}\delta_\ell$ in (32) dominates the second term. Then we have

$$f_\ell - f_{\ell+1} = \mathcal{O}(n\overline{L}\delta_\ell^2) = \mathcal{O}(nL\delta_\ell^2) \quad \text{for } \ell \in \mathcal{B} \tag{33}$$

and denote $\mathcal{B}_1 := \{\ell \in \mathcal{B} \mid (33) \text{ holds}\}$.
Case 2. The second term $4\gamma_e^{R_m-1}\omega/(\gamma_{\rm rd}\delta_\ell)$ in (32) dominates the first term. Then we conclude from (27) that

$$f_\ell - f_{\ell+1} = \mathcal{O}\Big(n(\omega/\delta_\ell)^2\Big) = \mathcal{O}\Big(n(\omega/\delta_{\min})^2\Big) = \mathcal{O}(nL\omega) \quad \text{for } \ell \in \mathcal{B} \tag{34}$$

and denote $\mathcal{B}_2 = \{\ell \in \mathcal{B} \mid (34) \text{ holds}\}$.
Then for a given $0 < \eta < \frac{1}{2}$ we conclude from (19)–(21) that with probability $\geq 1 - 2^{-R_v} \geq 1 - \eta > \frac{1}{2}$

$$\begin{aligned}
\sum_{\ell \in \mathcal{B}} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\delta_\ell^2} &= \sum_{\ell \in \mathcal{B}_1} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\delta_\ell^2} + \sum_{\ell \in \mathcal{B}_2} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\delta_\ell^2} \\
&\leq \sum_{\ell \in \mathcal{B}_1} \frac{f_\ell - f_{\ell+1} + 2\omega}{\delta_\ell^2} + \sum_{\ell \in \mathcal{B}_2} \frac{f_\ell - f_{\ell+1} + 2\omega}{\delta_\ell^2} \\
&\leq \sum_{\ell \in \mathcal{B}_1} \frac{\mathcal{O}(nL\delta_\ell^2) + 2\omega}{\delta_\ell^2} + \sum_{\ell \in \mathcal{B}_2} \frac{\mathcal{O}(nL\omega) + 2\omega}{\delta_\ell^2} \\
&\leq \sum_{\ell \in \mathcal{B}} \frac{\mathcal{O}(nL\delta_\ell^2) + 2\omega}{\delta_\ell^2} + \sum_{\ell \in \mathcal{B}} \frac{\mathcal{O}(nL\omega) + 2\omega}{\delta_\ell^2} \\
&= \mathcal{O}(nL)K + \mathcal{O}(nL\omega) \sum_{\ell \in \mathcal{B}} \delta_\ell^{-2} + 4\omega \sum_{\ell \in \mathcal{B}} \delta_\ell^{-2} \\
&= \mathcal{O}(nL)K + \mathcal{O}(nL\omega) \sum_{\ell \in \mathcal{B}} Q^{2\ell-2} + 4\omega \sum_{\ell \in \mathcal{B}} Q^{2\ell-2} \\
&= \mathcal{O}(nL)K + \mathcal{O}(nL\omega)\mathcal{O}(Q^{2K}) + \omega\mathcal{O}(Q^{2K}) \\
&= \mathcal{O}(nL\log(\omega/L)^{-1}) + \mathcal{O}(nL\omega(\omega/L)^{-1}) + \omega\mathcal{O}(\omega^{-1}) \\
&= \mathcal{O}(nL\log\omega^{-1}),
\end{aligned}$$

so that by (i) and since $R_m = \lceil \log_2 \eta^{-1} \rceil \geq 2$,

$$N_K \leq 1 + (2R_m + 1)T_0 \sum_{\ell \in \mathcal{B}} \frac{\tilde{f}_\ell - \tilde{f}_{\ell+1}}{\gamma\delta_\ell^2} = \mathcal{O}(nR_mT_0L\log\omega^{-1})$$

holds with probability $\geq 1 - 2^{-R_v} \geq 1 - \eta > \frac{1}{2}$ for a given $0 < \eta < \frac{1}{2}$. □

Compared to the results discussed in Section 1.2, the order ω of our complexity bounds is the same as that of Berahas et al. [8] which is valid in expectation.

6 Numerical results

In this section, we compare `VRDFON` with other state-of-the-art DFO solvers listed in Table 1 for low to high dimensional problems (**results are averaged over five runs**).

6.1 Implementation details

The `VRDFON` package is publicly available at [33]. It contains `impVRDFON.pdf`, describing new practical enhancements, details of the solvers compared, testing and tuning for `VRDFON`, and extensive numerical results for noisy problems in low to high dimensions.

New practical enhancements: `VRDFON` uses several different directions to enrich `MLS` in the presence of noise. It is well known that the complexity of randomized DFO methods is better than that of deterministic methods by a factor of n in the worst case (cf. [6]); therefore, using random directions seems preferable to using deterministic ones. Even better directions than random directions are random approximate coordinate directions. Improved trust region directions are found by minimizing surrogate quadratic models in adaptively determined subspaces within a trust region. Perturbed random directions are perturbations of random directions by scaled approximate descent directions in adaptively determined subspaces. `VRDFON` constructs surrogate quadratic models in adaptively determined subspaces that can handle medium and large scale problems. Although these models have lower accuracy in higher dimensions, their usefulness has been confirmed in extensive numerical experiments in the presence of strong noise. `VRDFON` changes extrapolation step sizes heuristically when they become too small in the presence of substantial noise to prevent generating zero steps.

6.2 Test problems

For our numerical results, we used 549 unconstrained `CUTEst` test problems from the collection of Gould et al. [23]. To prepare these results, the test environment of Kimiaei & Neumaier [34] was used.

The starting point: As in [35], we choose the starting point $x^0 := 0$ and shift the arguments by

$$\xi_i := (-1)^{i-1}\frac{2}{2+i} \text{ for all } i = 1, \dots, n,$$

to avoid a solver guessing the solution of toy problems with a simple solution (e.g., all zeros or all ones) – there are quite a few of these in the `CUTEst` library. That is, the initial point is chosen by $x^0 := \xi$ and the initial inexact function value is $\tilde{f}_0 := \tilde{f}(x^0)$ while the other inexact function values are computed by $\tilde{f}_\ell := \tilde{f}(x^\ell + \xi)$ for all $\ell \geq 0$. In fact, this choice increases the difficulty of the problems.

Type of noise: In the numerical results reported here, uniform random noise is used, which is consistent with the assumption (A3). The function values are calculated by $\tilde{f} = f + (2 * \text{rand} - 1)\omega$, where f is the true function value and $\omega \geq 0$ is a noise level whose size identifies the difficulty of the noisy problems. Here rand stands for the uniformly distributed random number.

6.3 Performance measures

Efficiency and robustness: Efficiency measures the ability of a solver $s \in \mathcal{S}$ relative to an ideal solver. The number of function evaluations is taken as a suitable cost measure, and the efficiency relative to this measure is called the `nf` efficiency. The **robustness** of a solver counts the number of problems it solves.

Plots for efficiency and robustness: Two important tools for figuring out which solver is **robust** and **efficient** are the data profile of Moré & Wild [40] and the performance profile of Dolan & Moré [20], respectively, see `impVRDFON.pdf`. To see the behaviour of the compared solvers in the presence of low to high noise, we plot the number of problems solved and the efficiency versus the noise level.

Measure for the convergence speed: The quotients

$$q_s := (f_s - f_{\rm opt})/(f_0 - f_{\rm opt}) \quad \text{for } s \in \mathcal{S} \tag{35}$$

are measures to identify the convergence speed of the solver s to reach a minimum of the smooth true function f. These quotients are not available in real applications. Here
• f_s is the best function value found by the solver s,
• f_0 is the function value at the starting point (common for all solvers),
• $f_{\rm opt}$ is the function value at the best known point (in most cases a global minimizer or at least a better local minimizer) found by running a sequence of gradient-based and local/global gradient free solvers; see Appendix B in [35].

Maximum budgets and stopping tests: We consider a problem **solved** by the solver s if $q_s \leq \varepsilon$ and neither the maximum number `nfmax` of function evaluations nor the maximum allowed time `secmax` in seconds was reached, and **unsolved** otherwise. ε, `secmax` and `nfmax` are chosen so that the best solver can solve at least half of the problems. They depend on the dimension and the noise level because increasing the noise level and dimension extremely increases the difficulty of the problems. Therefore, ε is chosen slightly larger for problems in medium and high dimensions than for problems in low dimensions. The following choices were found valuable:

$$\texttt{secmax} = \begin{cases} 180 & \text{if } 1 \leq n \leq 300, \\ 420 & \text{if } 301 \leq n \leq 5000, \end{cases}$$

$$\texttt{nfmax} = \begin{cases} 2n^2 + 1000n + 5000 & \text{if } 1 \leq n \leq 300, \\ 500n & \text{if } 301 \leq n \leq 5000 \end{cases}$$

and

$$\varepsilon := \begin{cases} 10^{-3} & \text{if } \omega \in \{10^{-4}, 10^{-3}\} \text{ and } n \in [1, 30], \\ 10^{-2} & \text{if } \omega \in \{0.1, 0.9\} \text{ and } n \in [1, 30], \\ 10^{-3} & \text{if } \omega = 10^{-4} \text{ and } n \in [31, 300], \\ 0.05 & \text{if } \omega \in \{0.1, 0.01, 0.001\} \text{ and } n \in [31, 300], \\ 0.05 & \text{if } \omega \in \{10^{-5}, 10^{-4}, 10^{-3}\} \text{ and } n \in [301, 5000]. \end{cases}$$

6.4 Comparison and recommendation

For small to large scale problems, Figures 1 includes three comparisons between more robust and efficient solvers among all solvers listed in Table 1. For considering the data profiles and the performance profiles of compared solvers, see `impVRDFON.pdf`.

In summary, we conclude that `VRDFON` is
• the second efficient solver for problems in small dimensions in most cases;
• the fifth robust solver for problems in small dimensions in most cases;
• the first efficient solver for problems in medium dimensions in most cases;
• the fourth robust solver for problems in medium dimensions in most cases;
• the first efficient solver for problems in large dimensions in most cases;
• the third robust solver for problems in large dimensions in most cases.

As a consequence of this summary, `VRDFON` is highly recommended for unconstrained noisy DFO problems when the computation of the function value is expensive.

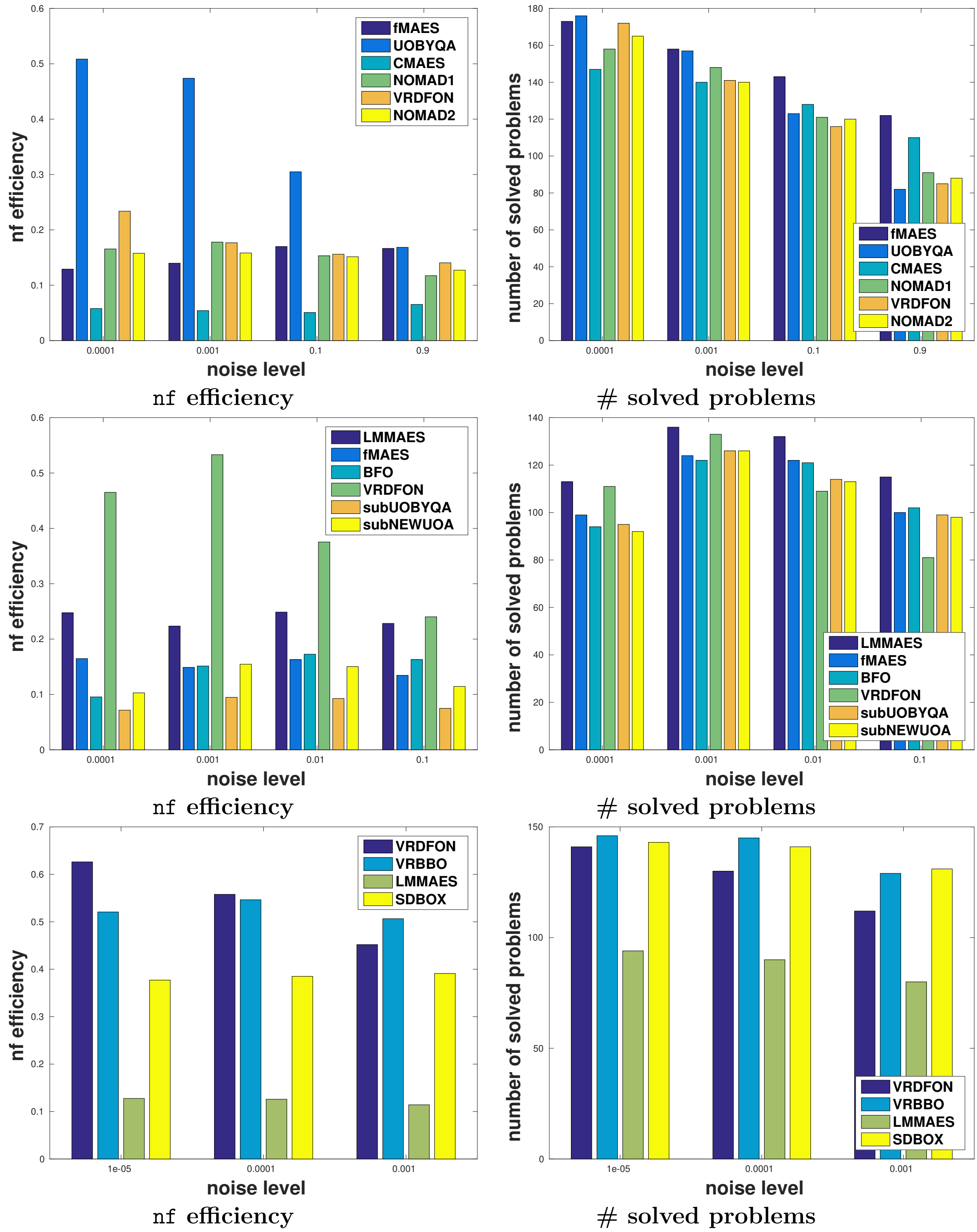

Figure 1: Comparison between more robust and efficient DFO solvers: small dimensions $1 < n \leq 30$ (first row), medium dimensions $30 < n \leq 300$ (second row), and large dimensions (third row) $300 < n \leq 5000$.

References

[1] M. A. Abramson, C. Audet, G. Couture, J. E. Dennis, Jr., S. Le Digabel and C. Tribes. The NOMAD project. Software available at `https://www.gerad.ca/nomad/`.

[2] M. A. Abramson, C. Audet, J. E. Dennis, Jr., and S. Le Digabel. OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. *SIAM J. Optim* **20** (Jan 2009), 948–966.

[3] C. Audet and J. E. Dennis. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.* **17** (January 2006), 188–217.

[4] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *2005 IEEE Congress on Evolutionary Computation*. IEEE (2005).

[5] C. Audet and W. Hare. Springer Series in Operations Research and Financial Engineering. *Derivative-Free and Blackbox Optimization*. 2017.

[6] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim* **24** (January 2014), 1238–1264.

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

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

[9] E. H Bergou, E. Gorbunov, and P. Richtárik. Stochastic three points method for unconstrained smooth minimization, *SIAM J. Optim.* **30** (January 2020), 2726–2749.

[10] H. G. Beyer. Design principles for matrix adaptation evolution strategies. Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. `https://doi.org/10.1145/3377929.3389870`.

[11] H. G. Beyer and B. Sendhoff. Simplify your covariance matrix adaptation evolution strategy. *IEEE Trans. Evol. Comput.* **21** (October 2017), 746–759.

[12] J. Blanchet, C. Cartis, M. Menickelly, K. Scheinberg. Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS J. Comput.* **1**(2) (2019), 92–119.

[13] M. D. Buhmann. Radial basis functions. *Acta Numer.* **9** (January 2000), 1–38.

[14] R. Chen. *Stochastic Derivative-Free Optimization of Noisy Functions.* PhD thesis, Lehigh University (2015). Theses and Dissertations. 2548.

[15] A. R. Conn and K. Scheinberg and L. N. Vicente. Introduction to Derivative-Free Optimization. *Society for Industrial and Applied Mathematics*, 2009.

[16] A. R. Conn and Ph. L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In *Nonlinear Optimization and Applications*, pp. 27–47. Springer US (1996).

[17] C. Davis. Theory of positive linear dependence. *Amer. J. Math.* **76** (October 1954), 733.

[18] P. Deuflhard and G. Heindl. Affine invariant convergence theorems for newton's method and extensions to related methods. *SIAM J. Numer. Anal.* **16** (February 1979), 1–10.

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

[20] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.* **91** (January 2002), 201–213.

[21] K. J. Dzahini. Expected complexity analysis of stochastic direct-search. *Comput. Optim. Appl.* **81** (2022), 179–200.

[22] C. Elster and A. Neumaier. A grid algorithm for bound constrained optimization of noisy functions. *IMA J. Numer. Anal.* **15** (1995), 585–608.

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

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

[25] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Complexity and global rates of trust-region methods based on probabilistic models. *IMA J. Numer. Anal.* **38** (August 2017), 1579–1597.

[26] S. Gratton, Ph. L. Toint, and A. Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optim. Methods Softw.* **26** (October 2011), 873–894.

[27] L. Grippo and F. Rinaldi. A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations. *Comput. Optim. Appl.* **60** (June 2014), 1–33.

[28] L. Grippo and M. Sciandrone. Nonmonotone derivative-free methods for nonlinear equations. *Comput. Optim. Appl.* **37** (March 2007), 297–328.

[29] N. J. Higham. Optimization by direct search in matrix computations. *SIAM J. Matrix Anal. Appl.* **14** (April 1993), 317–333.

[30] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *J. Glob. Optim.* **14** (1999), 331–355.

[31] W. Huyer and A. Neumaier. SNOBFIT – stable noisy optimization by branch and fit. *ACM. Trans. Math. Softw.* **35** (July 2008), 1–25.

[32] W. Huyer and A. Neumaier. MINQ8: general definite and bound constrained indefinite quadratic programming. *Comput. Optim. Appl.* **69** (October 2017), 351–381.

[33] M. Kimiaei. (2022). The VRDFON solver. Software available at `https://github.com/GS1400/VRDFON`.

[34] M. Kimiaei and A. Neumaier. Testing and tuning optimization algorithm. Preprint, Vienna University, Fakultät für Mathematik, Universität Wien, Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria (2019).

[35] M. Kimiaei and A. Neumaier. Efficient unconstrained black box optimization. *Math. Program. Comput.* **14** (February 2022), 365–414.

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

[37] I. Loshchilov, T. Glasmachers, and H. G. Beyer. Large scale black-box optimization by limited-memory matrix adaptation. *IEEE Trans. Evol. Comput.* **23** (April 2019), 353–358.

[38] S. Lucidi and M. Sciandrone. A derivative-free algorithm for bound constrained optimization. *Comput. Optim. Appl.* **21** (2002), 119–142.

[39] L. Lukšan, L and J. Vlcek. Sparse and partially separable test problems for unconstrained and equality constrained optimization. ICS AS CR, 1999. 30 s. Technical Report, V-767.

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

[41] A. Neumaier, H. Fendl, H. Schilly, and Thomas Leitner. VXQR: derivative-free unconstrained optimization based on QR factorizations. *Soft Comput.* **15** (September 2010), 2287–2298.

[42] M. Porcelli and P. Toint. Global and local information in structured derivative free optimization with BFO. *arXiv: Optimization and Control* (2020).

[43] M. J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Math. Program.* **92** (May 2002), 555–582.

[44] M. J. D. Powell. Developments of NEWUOA for minimization without derivatives. *IMA. J. Numer. Anal.* **28** (February 2008), 649–664.

[45] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Global. Optim.* **56** (July 2012), 1247–1293.

[46] V. J. Torczon. *Multidirectional search: A direct search algorithm for parallel machines.* PhD thesis, Diss., Rice University (1989).

[47] B. Van Dyke and T. J. Asaki. Using QR decomposition to obtain a new instance of mesh adaptive direct search with uniformly distributed polling directions. *J. Optim. Theory Appl* **159** (June 2013), 805–821.

[48] S. M. Wild, R. G. Regis, and C. A. Shoemaker. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM J. Sci. Comput.* **30** (January 2008), 3197–3219.

[49] M. H Wright. Direct search methods: Once scorned, now respectable. *Pitman Research Notes in Math. Series* (1996), 191–208.