On the complexity of finding stationary points of smooth functions in one dimension

Sinho Chewi[*] SCHEWI@MIT.EDU
Massachusetts Institute of Technology

Sébastien Bubeck SEBUBECK@MICROSOFT.COM
Microsoft Research

Adil Salim ADILSALIM@MICROSOFT.COM
Microsoft Research

Editors: Shipra Agrawal and Francesco Orabona

Abstract

We characterize the query complexity of finding stationary points of one-dimensional non-convex but smooth functions. We consider four settings, based on whether the algorithms under consideration are deterministic or randomized, and whether the oracle outputs $1^{\rm st}$-order or both $0^{\rm th}$- and $1^{\rm st}$-order information. Our results show that algorithms for this task provably benefit by incorporating either randomness or $0^{\rm th}$-order information. Our results also show that, for every dimension $d \ge 1$, gradient descent is optimal among deterministic algorithms using $1^{\rm st}$-order queries only.

Keywords: gradient descent, non-convex optimization, oracle complexity, stationary point

1. Introduction

We consider optimizing a non-convex but smooth function $f : \mathbb{R}^d \to \mathbb{R}$, a task which underlies the spectacular successes of modern machine learning. Despite the fundamental nature of this question, there are still important aspects which remain poorly understood.

To set the stage for our investigation, let $f : \mathbb{R}^d \to \mathbb{R}$ be a β-smooth function with bounded objective gap: $f(0) - \inf f \le \Delta$. Since global minimization of f is, in general, computationally intractable (c.f. Nemirovsky and Yudin, 1983), we focus on the task of outputting an *ε-stationary point*, that is, a point $x^\star \in \mathbb{R}^d$ such that $\|\nabla f(x^\star)\| < \varepsilon$. By a standard rescaling argument (see Lemma 1), it suffices to consider the case $\beta = \Delta = 1$. Then, it is well-known (see, e.g., Nesterov, 2018), that the standard gradient descent (GD) algorithm solves this task in $O(1/\varepsilon^2)$ queries to an oracle for the gradient ∇f. Conversely, Carmon et al. (2020) proved that if the dimension d is sufficiently large, then any randomized algorithm for this task must use at least $\Omega(1/\varepsilon^2)$ queries to a local oracle for f, thereby establishing the optimality of GD in high dimension.

However, the *low-dimensional complexity* of computing stationary points remains open. Indeed, the main limitation of Carmon et al. (2020) is that their lower bound constructions require the ambient dimension to be large: more precisely, they require $d \ge \Omega(1/\varepsilon^2)$ for deterministic algorithms, and $d \ge \widetilde{\Omega}(1/\varepsilon^4)$ for randomized algorithms. The large dimensionality arises because they adapt to the non-convex smooth setting a "chain-like" lower bound construction for optimization of a convex non-smooth function (Nesterov, 2018). The chain-like construction forces certain natural classes of iterative algorithms to explore only one new dimension per iteration, and hence the dimension of the "hard" function in the construction is at least as large as the iteration complexity.

[*] This work was completed while SC was a research intern at Microsoft Research.

In fact, the non-convex and smooth setting shares interesting parallels with the convex and non-smooth setting, despite their apparent differences (in the former setting, we seek an ε-stationary point, whereas in the latter setting, we seek an ε-minimizer). Namely, in both settings the optimal oracle complexity is $\Theta(1/\varepsilon^2)$ in high dimension, and the optimal algorithm is (sub)gradient descent (as opposed to the convex smooth setting, for which accelerated gradient methods outperform GD). However, for the convex non-smooth setting, we know that the large dimensionality $d \geq \Omega(1/\varepsilon^2)$ of the lower bound construction is almost necessary, because of the existence of cutting-plane methods (see, e.g., Bubeck, 2015; Nesterov, 2018) which achieve a better complexity of $O(d \log(1/\varepsilon))$ in dimension $d \leq \widetilde{O}(1/\varepsilon^2)$. This raises the question of whether or not there exist analogues of cutting-plane methods for *non-convex* optimization.

A negative answer to this question would substantially improve our understanding of non-convex optimization, as it would point towards fundamental algorithmic obstructions. As such, the low-dimensional complexity of finding stationary points for non-convex optimization was investigated in a series of works (Vavasis, 1993; Hinder, 2018; Bubeck and Mikulincer, 2020). These results show the existence of algorithms which improve upon GD in dimension $d \leq O(\log(1/\varepsilon))$. This suggests that GD is actually optimal for all $d \geq \Omega(\log(1/\varepsilon))$. To date, there has been little progress on this tantalizing conjecture because the existing low-dimensional lower bounds are delicate, relying on the theory of unpredictable random walks (Vavasis, 1993; Benjamini et al., 1998; Bubeck and Mikulincer, 2020).

Algorithm Class	**Oracle**	**Complexity**	**Lower Bound**	**Upper Bound**
Deterministic	1^{st}	$\Theta(1/\varepsilon^2)$	Theorem 4	GD (well-known)
Randomized	1^{st}	$\Theta(1/\varepsilon)$	Theorem 2	Theorem 3
Deterministic	$0^{\text{th}} + 1^{\text{st}}$	$\Theta(\log(1/\varepsilon))$	Theorem 5	Theorem 6
Randomized	$0^{\text{th}} + 1^{\text{st}}$	$\Theta(\log(1/\varepsilon))$	Theorem 5	Theorem 6

Table 1: Summary of the results of this work.

Our contributions. In this paper, we study the task of finding an ε-stationary point of a smooth and univariate function $f : \mathbb{R} \to \mathbb{R}$. Our results, which are summarized as Table 1, provide a complete characterization of the oracle complexity of this task in four settings, based on whether or not the algorithm is allowed to use external randomness and whether or not the oracle outputs zeroth-order information. In particular, our lower bounds, which hold in dimension one, also hold in every dimension $d \geq 1$. In spite of the simplicity of the setting, we can draw a number of interesting conclusions from the results.

- **Optimality of GD for any dimension $d \geq 1$.** Our results imply that, among algorithms which are deterministic and only use first-order queries, GD is optimal in every dimension $d \geq 1$. This was previously known only for $d \geq \Omega(1/\varepsilon^2)$ (Carmon et al., 2020).

- **Separations between algorithm classes and oracles**. Our results exhibit a natural setting in which both randomization and zeroth-order queries provably improve the query complexity of optimization. It shows, in particular, that at least one of these additional ingredients is *necessary* to improve upon the basic GD algorithm.

- **Finding stationary points for unconstrained optimization**. The methods of Vavasis (1993); Bubeck and Mikulincer (2020) for improving upon the complexity of GD in low dimension are applicable to the constrained case in which the domain of f is the cube $[0,1]^d$, and it is not obvious that they can be applied to unbounded domains. We address this question by characterizing the oracle complexity for the unconstrained case.

Related works. Usually, optimization lower bounds are established for specific classes of algorithms, such as algorithms for which each iterate lies in the span of the previous iterates and gradients (Nesterov, 2018). As noted in Woodworth and Srebro (2017), lower bounds against arbitrary randomized algorithms for convex optimization are trickier and are often loose with regards to the dimension in which the construction is embedded. The complexity of finding stationary points is further studied in Carmon et al. (2021).

Conventions and notation. A function $f : \mathbb{R}^d \to \mathbb{R}$ is β-*smooth* if it is continuously differentiable and its gradient ∇f is β-Lipschitz. If $d = 1$, we shall write f' instead of ∇f. We use the standard asymptotic notation $\Omega(\cdot)$, $O(\cdot)$, and $\Theta(\cdot)$.

2. Results

In this section, we give detailed statements of our results as well as proof sketches. The full proofs are deferred to the appendix. We also record the following lemma, which allows us to reduce to the case of $\beta = \Delta = 1$.

Lemma 1 *Let $\mathscr{C}_*(\varepsilon; \beta, \Delta, d, \mathscr{O}) \geq 0$ denote the complexity of finding an ε-stationary point over the class of β-smooth functions $f : \mathbb{R}^d \to \mathbb{R}$ with $f(0) - \inf f \leq \Delta$ using an oracle $\mathscr{O}$, where given $x \in \mathbb{R}^d$ the oracle $\mathscr{O}$ returns either $\nabla f(x)$ (first-order information) or $(f(x), \nabla f(x))$ (zeroth- and first-order information). Here, $* \in \{det, rand\}$ is a subscript denoting whether or not the algorithm is allowed to use external randomness; when $* = rand$, the randomized complexity refers to the minimum number of queries required to find an ε-stationary point with probability at least $1/2$. Then, for any $\beta, \Delta, \varepsilon > 0$,*

$$\mathscr{C}_*(\varepsilon; \beta, \Delta, d, \mathscr{O}) = \mathscr{C}_*\bigl(\frac{\varepsilon}{\sqrt{\beta\Delta}}; 1, 1, d, \mathscr{O}\bigr) .$$

Proof Given a β-smooth function $f : \mathbb{R}^d \to \mathbb{R}$ with $f(0) - \inf f \leq \Delta$, define $g : \mathbb{R}^d \to \mathbb{R}$ via $g(x) := \Delta^{-1} f(\sqrt{\Delta/\beta}\, x)$. Then, g is 1-smooth with $g(0) - \inf g \leq 1$, and it is clear that the oracle for g can be simulated using the oracle for f. Moreover, an $\varepsilon/\sqrt{\beta\Delta}$-stationary point for g translates into an ε-stationary point for f. Obviously, the reduction is reversible. ■

Often, we will assume without loss of generality that $f(0) = 1$ and $\beta = \Delta = 1$, so that $f \geq 0$. Also, we may assume that $f'(0) \leq -\varepsilon$, since if $f'(0) \in (-\varepsilon, \varepsilon)$ then 0 is an ε-stationary point of f, and if $f'(0) \geq \varepsilon$ we can replace f by $x \mapsto f(-x)$. We abbreviate $\mathscr{C}_*(\varepsilon; \mathscr{O}) := \mathscr{C}_*(\varepsilon; 1, 1, 1, \mathscr{O})$, and from now on we consider $d = 1$.

Let $\mathscr{O}^{1^{\text{st}}}$ denote the oracle which returns first-order information (given $x \in \mathbb{R}$, it outputs $f'(x)$), and let $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$ denote the oracle which returns zeroth- and first-order information (given $x \in \mathbb{R}$, it outputs $(f(x), f'(x))$). We remark that in the one-dimensional setting, we could instead assume access to an oracle $\mathscr{O}^{0^{\text{th}}}$ which only outputs zeroth-order information, rather than $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$; this is because we can simulate $\mathscr{O}^{1^{\text{st}}}$ to arbitrary accuracy given $\mathscr{O}^{0^{\text{th}}}$ with only a constant factor overhead in the number of oracle queries by using finite differences. For simplicity, we work with $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$ and we will not consider $\mathscr{O}^{0^{\text{th}}}$ further.

2.1. Lower bound for randomized algorithms

We begin with a lower bound construction for randomized algorithms which only use first-order queries. For simplicity, assume that $1/\varepsilon$ is an integer. We construct a family of functions $(f_j)_{j\in[1/\varepsilon]}$, with the following properties. On the negative half-line $\mathbb{R}_-$, each f_j decreases with slope $-\varepsilon$, with $f_j(0) = 1$. We also set the slope of f_j on the positive half-line $\mathbb{R}_+$ to be $-\varepsilon$, but this entails that $f_j(x) < 0$ for $x > 1/\varepsilon$, violating the constraint $f_j(0) - \inf f \le 1$. Instead, on the interval $[j-1, j]$, we modify f_j to increase as much as possible while remaining $O(1)$-smooth, so that $f_j(1/\varepsilon) = f_j(0) = 1$; we can then periodically extend f_j on the rest of $\mathbb{R}_+$.

Due to the periodicity of the construction, we can restrict our attention to the interval $[0, 1/\varepsilon]$. Without prior knowledge of the index j, any algorithm only has a "probability" (made precise in Appendix A.2) of at most ε of finding the interval $[j-1, j]$, which contains all of the ε-stationary points in $[0, 1/\varepsilon]$. Hence, we expect that any randomized algorithm must require at least $\Omega(1/\varepsilon)$ queries to find an ε-stationary point of f_j.

To make this formal, let $\Phi : [0,1] \to \mathbb{R}$ be a smooth function such that $\Phi(0) = 0$, $\Phi(1) = 1$, and $\Phi'(0) = \Phi'(1) = -\varepsilon$. For example, we can take

$$\Phi(x) = \begin{cases} 2\,(1+\varepsilon)\,x^2 - \varepsilon\,x\,, & x \in [0, \frac{1}{2}]\,, \\ 2\,\Phi(\frac{1}{2}) - \Phi(1-x)\,, & x \in [\frac{1}{2}, 1]\,. \end{cases}$$

We can check that Φ satisfies the desired properties and that Φ is β-smooth with $\beta = 4\,(1+\varepsilon) \le 5$ for $\varepsilon \le \frac{1}{4}$. Then, let

$$f_j(x) \coloneqq \begin{cases} 1 - \varepsilon\,x\,, & x \in (-\infty, j-1]\,, \\ 1 - \varepsilon\,(j-1) + \Phi(x-(j-1))\,, & x \in [j-1, j]\,, \\ f_j(j) - \varepsilon\,(x-j)\,, & x \in [j, 1/\varepsilon]\,, \\ f_j(x - 1/\varepsilon)\,, & x \in [1/\varepsilon, \infty)\,. \end{cases}$$

It follows that f_j is also 5-smooth, with $f_j(0) - \inf f_j \le 1$; see Figure 1.

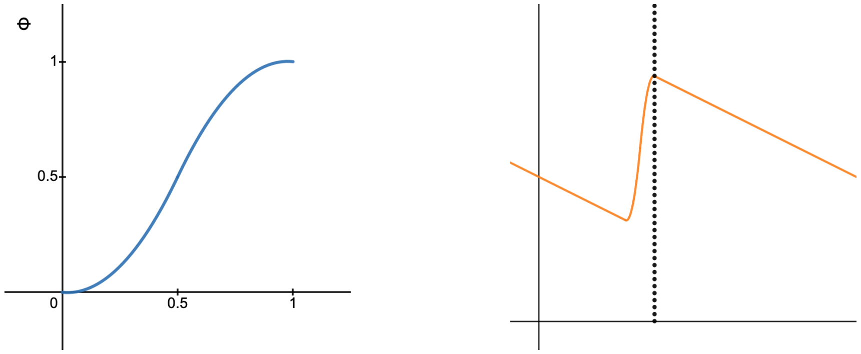

Figure 1: (Left) A plot of Φ. (Right) A plot of f_j, where the dotted line indicates the value of j.

We prove the following theorem in Appendix A.2.

Theorem 2 *For all $\varepsilon \in (0, \frac{1}{8})$, it holds that*

$$\mathscr{C}_{\mathrm{rand}}\big(\varepsilon; \mathscr{O}^{1^{\mathrm{st}}}\big) \ge \Omega\Big(\frac{1}{\varepsilon}\Big)\,.$$

2.2. An optimal randomized algorithm

The lower bound construction of the previous section suggests a simple strategy for computing an ε-stationary point of f: namely, just repeatedly pick points uniformly at random in the interval $[0, 1/\varepsilon]$. We now show that such a strategy (together with some additional processing steps) succeeds at obtaining an ε-stationary point in $O(1/\varepsilon)$ queries.

```
Algorithm 1: RANDOMSEARCH
Data: oracle $\mathscr{O}^{1^{\text{st}}}$ for $f$
Result: $\varepsilon$-stationary point $x$
while true do
    draw $x \sim \mathsf{uniform}([0, 2/\varepsilon])$
    if $|f'(x)| < \varepsilon$ then
        output $x$
    else if $f'(x) > 0$ then
        call BINARYSEARCH($\mathscr{O}^{1^{\text{st}}}, 0, x$)
end
```

```
Algorithm 2: BINARYSEARCH
Data: oracle $\mathscr{O}^{1^{\text{st}}}$ for $f$; initial points $x_0 < x_1$
      with $f'(x_0) \le -\varepsilon$ and $f'(x_1) > 0$
Result: $\varepsilon$-stationary point $x$
set $m \leftarrow \frac{x_0 + x_1}{2}$
if $|f'(m)| < \varepsilon$ then
    output $m$
else if $f'(m) \le -\varepsilon$ then
    call BINARYSEARCH($\mathscr{O}^{1^{\text{st}}}, m, x_1$)
else if $f'(m) > 0$ then
    call BINARYSEARCH($\mathscr{O}^{1^{\text{st}}}, x_0, m$)
```

The pseudocode for the algorithms is given as Algorithms 1 and 2. In short, RANDOMSEARCH (Algorithm 1) uses $O(1/\varepsilon)$ queries to find a "good point", i.e., either an ε-stationary point or a point x with $f'(x) > 0$. In the latter case, BINARYSEARCH (Algorithm 2) then locates an ε-stationary point using an additional $O(\log(1/\varepsilon))$ queries.

We prove the following theorem in Appendix A.3.

Theorem 3 *Assume that $f : \mathbb{R} \to \mathbb{R}$ is 1-smooth, $f \ge 0$, $f(0) = 1$, and $f'(0) \le -\varepsilon$. Then,* RANDOMSEARCH *(Algorithm 1) terminates with an ε-stationary point for f using at most $O(1/\varepsilon)$ queries to the oracle with probability at least $1/2$.*

As usual, the success probability can be boosted by rerunning the algorithm. In Figure 2, we demonstrate the performance of RANDOMSEARCH in a numerical experiment as a sanity check.

2.3. Lower bound for deterministic algorithms

Against the class of deterministic algorithms, the construction of Theorem 2 can be strengthened to yield a $\Omega(1/\varepsilon^2)$ lower bound. The idea is based on the concept of a *resisting oracle* $\mathscr{O}^{\text{resist}}$ from Nesterov (2018) which, regardless of the query point x, outputs "$f'(x) = -\varepsilon$". The goal then is to show that for any deterministic sequence of queries $x_1, \dots, x_N$, if $N \le O(1/\varepsilon^2)$, there exists a 1-smooth function $f : \mathbb{R} \to \mathbb{R}$ with $f(0) - \inf f \le \Delta$ which is consistent with the output of the oracle, i.e., satisfies $f'(x_i) = -\varepsilon$ for all $i \in [N]$. Note that this strategy necessarily only provides a lower bound against deterministic algorithms.[1]

[1] In more detail, the argument is as follows. Let $x_1, \dots, x_N$ be the sequence of query points generated by the algorithm when run with $\mathscr{O}^{\text{resist}}$, and suppose we can find a function f which is consistent with the responses of $\mathscr{O}^{\text{resist}}$. Then, for a deterministic algorithm, we can be sure that *had the algorithm been run with the oracle $\mathscr{O}^{1^{\text{st}}}$ for f*, it would have generated the same sequence of query points $x_1, \dots, x_N$, and hence would have never found an ε-stationary point of f among the N query points. This argument fails if the algorithm incorporates external randomness.

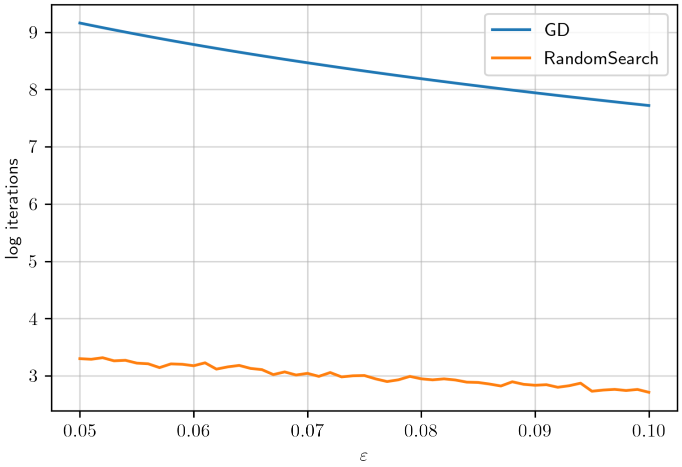

Figure 2: Iteration complexity of gradient descent (GD) vs. one run of RANDOMSEARCH (Algorithm 1) for various choices of ε on an instance of the construction in Section 2.1. The flatter slope of the orange line reflects the improved $O(1/\varepsilon)$ complexity of RANDOMSEARCH over the $O(1/\varepsilon^2)$ complexity of GD.

For simplicity of notation, since the order of the queries does not matter here, we assume that the queries are sorted: $x_1 < \cdots < x_N$. The function f that we construct has slope $-\varepsilon$ at the query points, but rapidly rises in between the query points to ensure that the condition $f(0) - \inf f \le 1$ holds. Moreover, we will ensure that $f'(x) = -\varepsilon$ for $x \le 0$ and that f' is periodic on $\mathbb{R}_+$ with period $1/\varepsilon$; hence, we may assume that all of the queries lie in the informative interval $(0, 1/\varepsilon)$. The key here is that for deterministic algorithms, the intervals on which the function f rises can be adapted to the query points, rather than being selected in advance.

The intuition is as follows. If the algorithm has made fewer than $O(1/\varepsilon^2)$ queries, then there must be $\Omega(1/\varepsilon^2)$ disjoint intervals in $[0, 1/\varepsilon]$ of length at least $\Omega(\varepsilon)$ in which there are no query points. On each such interval, we can grow our function value by $\Omega(\varepsilon^2)$ while staying smooth and with slope $-\varepsilon$ at the start and end of the interval. Hence, we can guarantee that the constructed function f remains above $f(0) - 1$, while answering $f'(x) = -\varepsilon$ at every query point x.

To make this precise, let $\ell_i \coloneqq x_{i+1} - x_i$ and define the function

$$\begin{aligned}\Phi_i(x) &\coloneqq -\varepsilon\,(x - x_i) \\ &\qquad + \begin{cases} \frac{1}{2}\,(x - x_i)^2\,, & x \in [x_i, x_i + \frac{\ell_i}{2}]\,, \\ \frac{\ell_i^2}{8} + \frac{\ell_i}{2}\,(x - x_i - \frac{\ell_i}{2}) - \frac{1}{2}\,(x - x_i - \frac{\ell_i}{2})^2\,, & x \in [x_i + \frac{\ell_i}{2}, x_{i+1}]\,. \end{cases}\end{aligned}$$

The construction of Φ_i satisfies the following properties:

1. Φ_i is continuously differentiable and 1-smooth on $[x_i, x_{i+1}]$.

2. $\Phi_i(x_i) = 0$ and $\Phi_i(x_{i+1}) = \ell_i\,(\frac{\ell_i}{4} - \varepsilon)$.

3. $\Phi_i'(x_i) = \Phi_i'(x_{i+1}) = -\varepsilon$.

Write $x_0 \coloneqq 0$ and $x_{N+1} \coloneqq 1/\varepsilon$. Recall that $x_i \in (0, 1/\varepsilon)$, for all $i \in [N]$. We now define

$$f(x) \coloneqq \begin{cases} 1 - \varepsilon\, x\,, & x \in (-\infty, 0]\,, \\ f(x_i) - \varepsilon\,(x - x_i)\,, & x \in [x_i, x_{i+1}] \text{ and } \ell_i < 8\varepsilon \;\; (0 \le i \le N)\,, \\ f(x_i) + \Phi_i(x)\,, & x \in [x_i, x_{i+1}] \text{ and } \ell_i \ge 8\varepsilon \;\; (0 \le i \le N)\,, \\ f(x - 1/\varepsilon) + a\,, & x \in [1/\varepsilon, \infty)\,, \end{cases}$$

where $a \coloneqq f(1/\varepsilon) - f(0)$. See Figure 3 for an illustration of f. We shall prove that when $N \le O(1/\varepsilon^2)$, then the function f is 1-smooth and satisfies $f(0) - \inf f \le 1$, thus completing the resisting oracle construction. It yields the following theorem, which we prove in Appendix A.4.

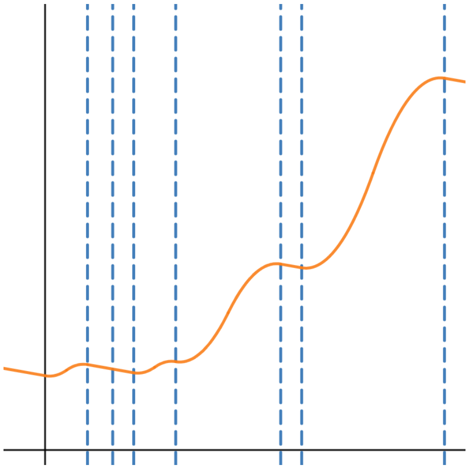

Figure 3: We plot an example of the function f. The dashed lines indicate the query points made by the algorithms.

Theorem 4 *For all $\varepsilon \in (0,1)$, it holds that*

$$\mathscr{C}_{\mathrm{det}}\big(\varepsilon; \mathscr{O}^{1^{\mathrm{st}}}\big) \ge \Omega\Big(\frac{1}{\varepsilon^2}\Big)\,.$$

The lower bound is matched by gradient descent. For the sake of completeness, we provide a proof of the matching $O(1/\varepsilon^2)$ upper bound via gradient descent as Theorem 8 in Appendix A.1.

2.4. Lower bound for randomized algorithms with zeroth-order information

We now turn towards algorithms which use the $0^{\mathrm{th}}+1^{\mathrm{st}}$-order oracle $\mathscr{O}^{0^{\mathrm{th}}+1^{\mathrm{st}}}$. For the lower bound, we again use the family of functions $(f_j)_{j\in[1/\varepsilon]}$ introduced in Section 2.1. The main difference is that given a query point $x \in [0, 1/\varepsilon]$, the value of $f_j(x)$ reveals whether or not the interval $[j-1, j]$ lies to the left of x and hence allows for binary search to determine j. Consequently, the lower bound is only of order $\Omega(\log(1/\varepsilon))$.

We prove the following theorem in Appendix A.5.

Theorem 5 *For all $\varepsilon \in (0, \frac{1}{8})$, it holds that*

$$\mathscr{C}_{\text{det}}\big(\varepsilon; \mathscr{O}^{0^{\text{th}}+1^{\text{st}}}\big) \ge \mathscr{C}_{\text{rand}}\big(\varepsilon; \mathscr{O}^{0^{\text{th}}+1^{\text{st}}}\big) \ge \Omega\Big(\log\frac{1}{\varepsilon}\Big)\,.$$

2.5. An optimal deterministic algorithm with zeroth-order information

Finally, we provide a deterministic algorithm whose complexity matches the lower bound in Theorem 5. At a high level, the idea is to use the zeroth-order information to perform binary search, but the actual algorithm is slightly more involved and requires the consideration of various cases.

We summarize the idea behind the algorithm. First, as described earlier, we may freely assume $f \ge 0$, $f(0) = 1$, and $f'(0) \le -\varepsilon$. Also, we recall that if the algorithm ever sees a point x with either $|f'(x)| < \varepsilon$ or $f'(x) > 0$, then we are done (in the latter case, we can call Algorithm 2: BINARYSEARCH).

1. DECREASEGAP (Algorithm 4) checks the value of $f(2/\varepsilon)$. If $f(2/\varepsilon) \le \frac{3}{4}\,f(0)$, then we have made progress on the objective gap and we may treat $2/\varepsilon$ as the new origin. This can happen at most $O(\log(1/\varepsilon))$ times. Otherwise, we have $f(2/\varepsilon) \ge \frac{3}{4}\,f(0)$, and we move on to the next phase of the algorithm.
2. Set $x_- := 0$ and $x_+ := 2/\varepsilon$. There are two cases: either $\frac{3}{4}\,f(x_-) \le f(x_+) \le f(x_-)$, in which case $f(x_-) - f(x_+) \le \frac{\varepsilon}{4}\,(x_+ - x_-)$, or $f(x_+) \ge f(x_-)$.
3. The first case is handled by BINARYSEARCHII (Algorithm 5). A simple calculation reveals that the condition $0 \le f(x_-) - f(x_+) \le \frac{3}{4}\,(x_+ - x_-)$ together with $f'(x_-) \le -\varepsilon$ implies the existence of an ε-stationary point in $[x_-, x_+]$. We now check the midpoint m of x_- and x_+. If $f(m) \notin [f(x_+), f(x_-)]$, then we arrive at the second case. Otherwise, we replace either x_- or x_+ with m; one of these two choices will cut the value of $f(x_-) - f(x_+)$ by at least half, thereby ensuring that the condition $0 \le f(x_-) - f(x_+) \le \frac{3}{4}\,(x_+ - x_-)$ continues to hold. This can happen at most $O(\log(1/\varepsilon))$ times.
4. Finally, the second case is handled by BINARYSEARCHIII (Algorithm 6). In this case, $f(x_+) \ge f(x_-)$ together with $f'(x_-) \le -\varepsilon$ ensures that there is a stationary point in $[x_-, x_+]$. We then check the value of $f(m)$ where m is the midpoint of x_- and x_+. It is straightforward to check that we can replace either x_- or x_+ with m and preserve the condition $f(x_+) \ge f(x_-)$. This can happen at most $O(\log(1/\varepsilon))$ times.

```
Algorithm 3: ZEROTHORDER
Data: oracle $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$ for $f$
Result: $\varepsilon$-stationary point $x$
set $x_- \leftarrow$ DECREASEGAP($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$, 0)
set $x_+ \leftarrow x_- + 2/\varepsilon$
if $|f'(x_-)| < \varepsilon$ then
    output $x_-$
else if $f(x_+) \le f(x_-)$ then
    call BINARYSEARCHII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$, $x_-$, $x_+$)
else if $f(x_+) > f(x_-)$ then
    call BINARYSEARCHIII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$, $x_-$, $x_+$)
```

Algorithm 4: DECREASEGAP

```
Data: oracle $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$ for $f$; point $x_0$
Result: either an $\varepsilon$-stationary point $x$ or a point $x$ such that $f(x) \leq f(x_0)$, $f'(x) \leq -\varepsilon$, and
        $f(x + 2/\varepsilon) \geq \frac{3}{4}\, f(x)$
if $|f'(x_0 + 2/\varepsilon)| < \varepsilon$ then
    output $x_0 + 2/\varepsilon$
else if $f'(x_0 + 2/\varepsilon) > 0$ then
    call BINARYSEARCH($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_0, x_0 + 2/\varepsilon$)
else if $f(x_0 + 2/\varepsilon) \geq \frac{3}{4}\, f(x_0)$ then
    output $x_0$
else
    call DECREASEGAP($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_0 + 2/\varepsilon$)
```

Algorithm 5: BINARYSEARCHII

```
Data: oracle $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$ for $f$; points $x_- < x_+$ with $f'(x_-) \leq -\varepsilon$ and $0 \leq f(x_-) - f(x_+) \leq$
      $\frac{\varepsilon}{4}\,(x_+ - x_-)$
Result: an $\varepsilon$-stationary point $x$
set $m \leftarrow \frac{x_- + x_+}{2}$
if $|f'(m)| < \varepsilon$ then
    output $m$
else if $f'(m) > 0$ then
    call BINARYSEARCH($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_-, m$)
else if $f(m) \geq f(x_-)$ then
    call BINARYSEARCHIII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_-, m$)
else if $f(m) \leq f(x_+)$ then
    call BINARYSEARCHIII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, m, x_+$)
else if $f(x_-) - f(m) \leq \frac{1}{2}\,(f(x_-) - f(x_+))$ then
    call BINARYSEARCHII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_-, m$)
else if $f(m) - f(x_+) \leq \frac{1}{2}\,(f(x_-) - f(x_+))$ then
    call BINARYSEARCHII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, m, x_+$)
```

Algorithm 6: BINARYSEARCHIII

```
Data: oracle $\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}$ for $f$; points $x_- < x_+$ with $f'(x_-) \leq -\varepsilon$ and $f(x_+) \geq f(x_-)$
Result: an $\varepsilon$-stationary point $x$
set $m \leftarrow \frac{x_- + x_+}{2}$
if $|f'(m)| < \varepsilon$ then
    output $m$
else if $f'(m) > 0$ then
    call BINARYSEARCH($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_-, m$)
else if $f(m) \geq f(x_-)$ then
    call BINARYSEARCHIII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, x_-, m$)
else
    call BINARYSEARCHIII($\mathscr{O}^{0^{\text{th}}+1^{\text{st}}}, m, x_+$)
```

We prove the following theorem in Appendix A.6.

Theorem 6 *Assume that $f : \mathbb{R} \to \mathbb{R}$ is 1-smooth, $f \geq 0$, $f(0) = 1$, and $f'(0) \leq -\varepsilon$. Then,* ZEROTHORDER *(Algorithm 3) terminates with an ε-stationary point for f using at most $O(\log(1/\varepsilon))$ queries to the oracle.*

3. Conclusion

We have characterized the oracle complexity of finding an ε-stationary point of a smooth univariate function $f : \mathbb{R} \to \mathbb{R}$ in four natural settings of interest. Besides providing insight into the limitations of gradient descent, our results exhibit surprising separations between the power of deterministic and randomized algorithms, and between algorithms that use zeroth-order information and algorithms (like gradient descent) which only use first-order information.

We conclude with a number of open directions for future research.

- The main question motivating this work remains open, namely, for randomized algorithms using zeroth- and first-order information, **is it possible to prove a $\Omega(1/\varepsilon^2)$ complexity lower bound with a construction in dimension $d = O(\log(1/\varepsilon))$?** An affirmative answer to this question would likely build upon the lower bound techniques used in Vavasis (1993); Bubeck and Mikulincer (2020).

  An even more ambitious goal is to fully characterize the query complexity of finding stationary points using zeroth- and first-order information in every fixed dimension d.

- Towards the above question, we also ask: **is there an analogue of gradient flow trapping (Bubeck and Mikulincer, 2020) for unconstrained optimization?**

- We have established that among deterministic algorithms which only use first-order queries, gradient descent is optimal already in dimension one. Although randomized algorithms outperform GD in our setting of investigation, it is unclear to what extent randomness helps in higher dimension. Hence, we make the following bold conjecture: **can one prove a $\Omega(1/\varepsilon^2)$ complexity lower bound for randomized algorithms which only make first-order queries in dimension two?**

Acknowledgments

SC was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

Itai Benjamini, Robin Pemantle, and Yuval Peres. Unpredictable paths and percolation. *Ann. Probab.*, 26(3):1198–1211, 1998.

Sébastien Bubeck and Dan Mikulincer. How to trap a gradient flow. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 940–960. PMLR, 09–12 Jul 2020.

Sébastien Bubeck. Convex optimization: algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Math. Program.*, 184(1-2, Ser. A):71–120, 2020.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: first-order methods. *Math. Program.*, 185(1-2, Ser. A):315–355, 2021.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2006.

Oliver Hinder. Cutting plane methods can be extended into nonconvex optimization. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1451–1454. PMLR, 06–09 Jul 2018.

Arkadii S. Nemirovsky and David B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson.

Yurii Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2018.

Stephen A. Vavasis. Black-box complexity of local minimization. *SIAM J. Optim.*, 3(1):60–80, 1993.

Blake Woodworth and Nathan Srebro. Lower bound for randomized first order convex optimization. *arXiv e-prints*, art. arXiv:1709.03594, 2017.

Appendix A. Proofs

A.1. Preliminaries

The standard approach for proving lower bounds against randomized algorithms is to reduce the task under consideration to a statistical estimation problem, for which we can bring to bear tools from information theory. Namely, we use *Fano's inequality*; we refer readers to Cover and Thomas (2006, Chapter 2) for background on entropy and mutual information.

Theorem 7 (Fano's inequality) *Let m be a positive integer and let $J \sim \mathsf{uniform}([m])$. Then, for any estimator $\widehat{J}$ of J which is measurable w.r.t. some data Y, it holds that*

$$\mathbb{P}\{\widehat{J} \neq J\} \geq 1 - \frac{I(J;Y) + \ln 2}{\ln m}\,,$$

where I denotes the mutual information.

For the sake of completeness, we also include a proof of the $O(1/\varepsilon^2)$ complexity bound for gradient descent.

Theorem 8 *Suppose that $f : \mathbb{R}^d \to \mathbb{R}$ is* 1*-smooth with $f(0) - \inf f \le 1$. Set $x_0 \coloneqq 0$ and for $k \in \mathbb{N}$, consider the iterates of GD with step size* 1*:*

$$x_{k+1} \coloneqq x_k - \nabla f(x_k)\,.$$

Then,

$$\min_{k=0,1,\dots,N-1} \|\nabla f(x_k)\| \le \sqrt{\frac{2}{N}}\,.$$

Proof Due to the 1-smoothness of f,

$$f(x_{k+1}) - f(x_k) \le \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{1}{2}\,\|x_{k+1} - x_k\|^2 = -\frac{1}{2}\,\|\nabla f(x_k)\|^2\,. \tag{1}$$

Rearranging this and summing,

$$\begin{aligned}\min_{k=0,1,\dots,N-1} \|\nabla f(x_k)\|^2 &\le \frac{1}{N}\sum_{k=0}^{N-1} \|\nabla f(x_k)\|^2 \le \frac{2}{N}\sum_{k=0}^{N-1} \{f(x_k) - f(x_{k+1})\} \\ &\le \frac{2}{N}\,\{f(0) - f(x_N)\} \le \frac{2}{N}\,\{f(0) - \inf f\} \le \frac{2}{N}\,.\end{aligned}$$

■

A.2. Proof of Theorem 2

Proof [Proof of Theorem 2] By making the value of ε larger (up to a factor of 2), we may assume that $1/\varepsilon$ is an integer.

We reduce the optimization task to a statistical estimation problem. Let $J \sim \mathsf{uniform}([1/\varepsilon])$. Since the only regions in which $|f_j'| < \varepsilon$ are contained in intervals of the form $k/\varepsilon + [j-1, j]$ for some $k \in \mathbb{N}$, then finding an ε-stationary point of f_J implies that the algorithm can guess the value of J (exactly).

On the other hand, we lower bound the number of queries required to guess the value of J. Let $x_1, \dots, x_N$ denote the query points of the algorithm, which may also depend on an external source of randomness U. Write $\mathscr{O}_{f_j}(x) = f_j'(x)$ for the output of the oracle for f_j on the query x (we omit the superscript 1^{st} for brevity). Let $\widehat{J}$ be any estimator of J based on $\{x_i, \mathscr{O}_{f_J}(x_i) : i \in [N]\}$. Then, by Fano's inequality (Theorem 7),

$$\mathbb{P}\{\widehat{J} \ne J\} \ge 1 - \frac{I(\{x_i, \mathscr{O}_{f_J}(x_i) : i \in [N]\}; J) + \ln 2}{\ln(1/\varepsilon)}\,.$$

First, suppose that the algorithm is deterministic. This means that each x_i is a deterministic function of $\{x_{i'}, \mathscr{O}_{f_J}(x_{i'}) : i' \in [i-1]\}$. The chain rule for the mutual information implies that

$$\begin{aligned}&I\bigl(\{x_i, \mathscr{O}_{f_J}(x_i) : i \in [N]\}; J\bigr) \\ &\qquad \le \sum_{i=1}^{N} I\bigl(\mathscr{O}_{f_J}(x_i); J \bigm| \{x_{i'}, \mathscr{O}_{f_J}(x_{i'}) : i' \in [i-1]\}\bigr)\,.\end{aligned}$$

On the other hand, there are two possibilities for the i-th term in the summation. Either one of the previous queries already landed in an interval corresponding to J, in which case J is already known and the mutual information is zero, or none of the previous queries have hit an interval corresponding to J. In the latter case, conditionally on the information up to iteration i, J is uniformly distributed on $1/\varepsilon - i$ remaining intervals, and so

$$\begin{aligned} &I\big(\mathscr{O}_{f_J}(x_i); J \bigm| \{x_{i'}, \mathscr{O}_{f_J}(x_{i'}) : i' \in [i-1]\}\big) \\ &\qquad \le H\big(\mathscr{O}_{f_J}(x_i) \bigm| \{x_{i'}, \mathscr{O}_{f_J}(x_{i'}) : i' \in [i-1]\}\big) = h\big(\frac{1}{1/\varepsilon - i}\big)\,, \end{aligned}$$

with h denoting the entropy function $p \mapsto p \ln \frac{1}{p} + (1-p) \ln \frac{1}{1-p}$. The last inequality follows because conditionally, $\mathscr{O}_{f_J}(x_i)$ can only be one of two possible values with probabilities $\frac{1}{1/\varepsilon - i}$ and $1 - \frac{1}{1/\varepsilon - i}$ respectively. If $N \le 1/(2\varepsilon)$, then

$$I\big(\{x_i, \mathscr{O}_{f_J}(x_i) : i \in [N]\}; J\big) \le 2 \sum_{i=1}^N \frac{1}{1/\varepsilon - i} \ln\big(\frac{1}{\varepsilon} - i\big) \le 4N\varepsilon \ln \frac{1}{\varepsilon}\,.$$

Hence,

$$\mathbb{P}\{\widehat{J} \ne J\} \ge 1 - \frac{4N\varepsilon \ln(1/\varepsilon) + \ln 2}{\ln(1/\varepsilon)} > \frac{1}{2} \tag{2}$$

provided that $\varepsilon \le \frac{1}{8}$ and $N \le O(1/\varepsilon)$ for a sufficiently small implied constant. Although we have proven the bound (2) for deterministic algorithms, the bound (2) continues to hold for randomized algorithms simply by conditioning on the random seed U which is independent of J.

We have proven that any randomized algorithm which is guaranteed to find an ε-stationary point of f_J must use at least $N \ge \Omega(1/\varepsilon)$ queries, or

$$\mathscr{C}(\varepsilon; 5, 1, 1, \mathscr{O}^{\rm 1st}) \ge \Omega\big(\frac{1}{\varepsilon}\big)\,.$$

We conclude by applying the rescaling lemma (Lemma 1). ■

A.3. Proof of Theorem 3

First, we analyze the subroutine BINARYSEARCH.

Lemma 9 *Suppose that f is 1-smooth. Then,* BINARYSEARCH *(Algorithm 2) terminates with an ε-stationary point for f using at most $O(\log \frac{x_1 - x_0}{\varepsilon})$ queries to the oracle.*

Proof Since f is 1-smooth, $f(x_0) \le -\varepsilon$ and $f(x_1) > 0$ cannot hold if $x_1 - x_0 \le \varepsilon$. Moreover, each time that BINARYSEARCH fails to find an ε-stationary point for f, the length of the interval $[x_0, x_1]$ is cut in half. The result follows. ■

We also need one lemma about continuous functions on $\mathbb{R}$.

Lemma 10 *Let $g : \mathbb{R} \to \mathbb{R}$ be continuous, let I be a compact and non-empty interval, and let $\varepsilon > 0$. Then, there is a finite collection of disjoint closed intervals which cover $I \cap \{g \ge \varepsilon\}$ and which are contained in $I \cap \{g \ge 0\}$.*

Proof For each $x \in S := I \cap \{g \ge \varepsilon\}$, by continuity of g there exists a closed interval $I_x \subseteq I$ such that x belongs to the interior of I_x and such that $g \ge 0$ on I_x. The collection $(I_x)_{x \in S}$ covers the compact set S, so we can extract a finite subcover. The connected components of the union of the finite subcover consist of disjoint closed intervals. ■

We are now ready to prove Theorem 3.

Proof [Proof of Theorem 3] Let $x \sim \mathsf{uniform}([0, 2/\varepsilon])$. If $|f'(x)| < \varepsilon$, then we are done, and if $f'(x) > 0$, then Lemma 9 shows that BINARYSEARCH terminates with an ε-stationary point of f using $O(\log(1/\varepsilon))$ queries. What remains to show is that x satisfies either $|f'(x)| < \varepsilon$ or $f'(x) > 0$ with probability at least $\Omega(\varepsilon)$, which implies that Algorithm 1 succeeds using $O(1/\varepsilon)$ queries with probability at least $1/2$.

Let $\mathfrak{m}$ denote the Lebesgue measure restricted to $[0, 2/\varepsilon]$. Then,

$$\begin{aligned} 1 \ge f(0) - f(2/\varepsilon) &= -\int_{[0,2/\varepsilon]} f' \\ &\ge \varepsilon\, \mathfrak{m}\{f' \le -\varepsilon\} - \varepsilon\, \mathfrak{m}\{|f'| < \varepsilon\} - \int_{[0,2/\varepsilon] \cap \{f' \ge \varepsilon\}} f' \,. \end{aligned}$$

From Lemma 10, we can cover the set $[0, 2/\varepsilon] \cap \{f' \ge \varepsilon\}$ with a union of disjoint closed intervals $\bigcup_{k=1}^K I_k \subseteq [0, 2/\varepsilon] \cap \{f' \ge 0\}$. On I_k, the smoothness of f ensures that

$$-\int_{I_k} f' \ge -\mathfrak{m}(I_k)\, \underbrace{f'(\inf I_k)}_{\le \varepsilon} - \int_{I_k} (x - \inf I_k)\, \mathrm{d}x \ge -\varepsilon\, \mathfrak{m}(I_k) - \frac{1}{2}\, \mathfrak{m}(I_k)^2 \,.$$

Write $\ell_k := \mathfrak{m}(I_k) = \sup I_k - \inf I_k$. Note that $\sum_{k=1}^K \ell_k \le \mathfrak{m}\{f' \ge 0\}$. Thus,

$$\begin{aligned} -\int_{[0,2/\varepsilon] \cap \{f' \ge \varepsilon\}} f' &\ge -\varepsilon \sum_{k=1}^K \ell_k - \frac{1}{2} \sum_{k=1}^K \ell_k^2 \ge -\varepsilon \sum_{k=1}^K \ell_k - \frac{1}{2} \Big(\sum_{k=1}^K \ell_k\Big)^2 \\ &\ge -\varepsilon\, \mathfrak{m}\{f' \ge 0\} - \frac{1}{2}\, \mathfrak{m}\{f' \ge 0\}^2 \,. \end{aligned}$$

Now suppose that $\mathfrak{m}\{|f'| < \varepsilon \text{ or } f' \ge \varepsilon\} \le c_0$, where $c_0 > 0$ is a constant to be chosen later. In this case, the inequalities above imply

$$1 + 2c_0\varepsilon + \frac{1}{2}\, c_0^2 \ge \varepsilon\, \mathfrak{m}\{f' \le -\varepsilon\} \ge \varepsilon \left(\frac{2}{\varepsilon} - \mathfrak{m}\{|f'| < \varepsilon \text{ or } f' \ge \varepsilon\}\right)$$

which, when rearranged, yields

$$1 + 3c_0\varepsilon + \frac{1}{2}\, c_0^2 \ge 2 \,.$$

If c_0 is a sufficiently small absolute constant, we arrive at a contradiction.

We conclude that $\mathfrak{m}\{|f'| < \varepsilon \text{ or } f' \geq \varepsilon\} \geq c_0$, which means that the random point x will be good in the sense that either $|f'(x)| < \varepsilon$ or $f'(x) \geq \varepsilon$. The probability that Algorithm 1 fails to obtain a good random point in N tries is at most $(1 - c_0\varepsilon/2)^N$, which can be made at most $1/2$ by taking $N = \Theta(1/\varepsilon)$. We conclude that with probability at least $1/2$, using

$$O\big(\frac{1}{\varepsilon} + \log\frac{1}{\varepsilon}\big) = O\big(\frac{1}{\varepsilon}\big) \quad \text{queries}\,,$$

Algorithm 1 finds an ε-stationary point. ■

A.4. Proof of Theorem 4

Proof [Proof of Theorem 4] The goal is to show that when $N \leq O(1/\varepsilon^2)$, the resisting oracle construction succeeds, and hence no deterministic algorithm can find an ε-stationary point of an arbitrary 1-smooth function with objective gap at most 1 using N queries.

For the resisting oracle construction, the crux of the matter is to show that $a = f(1/\varepsilon) - f(0) \geq 0$. Indeed, if this holds, then since f is clearly bounded below by 0 on $[0, 1/\varepsilon]$ it will follow that $f \geq 0$ on all of $\mathbb{R}$, and hence $f(0) - \inf f \leq 1$.

Let I be the set of indices $i \in [N]$ for which $\ell_i \geq 8\varepsilon$. Since f has slope $-\varepsilon$ on all of the linear pieces, then over all of the linear pieces the value of f drops by at most 1 on the interval $[0, 1/\varepsilon]$. The goal is to show that

$$\sum_{i\in I}\{f(x_{i+1}) - f(x_i)\} \overset{!}{\geq} 1\,.$$

To prove this, write

$$\frac{1}{\varepsilon} = \sum_{i=1}^{N} \ell_i = \sum_{i\in I}\ell_i + \sum_{i\in I^{\mathsf{c}}}\ell_i \leq \sum_{i\in I}\ell_i + 8\varepsilon\,|I^{\mathsf{c}}|\,.$$

There are two cases to consider. If $|I^{\mathsf{c}}| \geq \frac{1}{16\varepsilon^2}$ queries, then we are done, as the algorithm has made $\Omega(1/\varepsilon^2)$ queries. Otherwise, $|I^{\mathsf{c}}| \leq \frac{1}{16\varepsilon^2}$, in which case

$$\frac{1}{2\varepsilon} \leq \sum_{i\in I}\ell_i\,.$$

In this second case, we now have

$$\begin{aligned}\sum_{i\in I}\{f(x_{i+1}) - f(x_i)\} = \sum_{i\in I}\Phi_i(x_{i+1}) = \sum_{i\in I}\ell_i\,\big(\frac{\ell_i}{4} - \varepsilon\big) \geq \frac{1}{8}\sum_{i\in I}\ell_i^2 \\ \geq \frac{1}{8\,|I|}\Big(\sum_{i\in I}\ell_i\Big)^2 \geq \frac{1}{32\varepsilon^2\,|I|}\,.\end{aligned}$$

This is greater than 1 provided $|I| \leq \frac{1}{32\varepsilon^2}$.

In summary, the resisting oracle construction is valid provided $|I| \le \frac{1}{32\varepsilon^2}$ and $|I^{\mathrm{c}}| \le \frac{1}{16\varepsilon^2}$. Since $|I| + |I^{\mathrm{c}}| = N$, any deterministic algorithm which finds an ε-stationary point must use at least $N \ge \min\{\frac{1}{32\varepsilon^2}, \frac{1}{16\varepsilon^2}\} = \frac{1}{32\varepsilon^2}$ queries, or

$$\mathscr{C}_{\det}(\varepsilon; \mathscr{O}^{1^{\mathrm{st}}}) \ge \frac{1}{32\varepsilon^2}\,.$$

■

A.5. Proof of Theorem 5

Proof [Proof of Theorem 5] The proof is very similar to the proof of Theorem 2. We follow the proof up to the point where

$$\begin{aligned} &I\big(\mathscr{O}_{f_J}(x_i); J \bigm| \{x_{i'}, \mathscr{O}_{f_J}(x_{i'}) : i' \in [i-1]\}\big) \\ &\qquad \le H\big(\mathscr{O}_{f_J}(x_i) \bigm| \{x_{i'}, \mathscr{O}_{f_J}(x_{i'}) : i' \in [i-1]\}\big)\,, \end{aligned}$$

where now $\mathscr{O}_{f_J}(x) = \{f_J(x), f_J'(x)\}$ returns zeroth- and first-order information. The key point now is that since x_i is deterministic (conditioned on previous queries), $\mathscr{O}_{f_J}(x_i)$ can only take a constant number of possible values, and so the above entropy term is $O(1)$ (as opposed to Theorem 2, in which the entropy term was of order $O(\varepsilon \log(1/\varepsilon))$). Plugging this into Fano's inequality (Theorem 7), we obtain

$$\mathbb{P}\{\widehat{J} \ne J\} \ge 1 - \frac{O(N) + \ln 2}{\ln(1/\varepsilon)} > \frac{1}{2}\,,$$

provided that $\varepsilon \le \frac{1}{8}$ and $N \le O(\log(1/\varepsilon))$. This proves that $\Omega(\log(1/\varepsilon))$ queries to $\mathscr{O}^{0^{\mathrm{th}}+1^{\mathrm{st}}}$ are necessary to find an ε-stationary point, even for a randomized algorithm. ■

A.6. Proof of Theorem 6

We prove the correctness of the algorithms in reverse order, beginning with BinarySearchIII.

Lemma 11 *Let $f : \mathbb{R} \to \mathbb{R}$ be 1-smooth. Then,* BinarySearchIII *(Algorithm 6) terminates with an ε-stationary point of f using $O(\log \frac{x_+ - x_-}{\varepsilon})$ queries to the oracle.*

Proof Due to the 1-smoothness of f, if $x_+ - x_- < \varepsilon$, then $f' < 0$ on the interval $[x_-, x_+]$, which contradicts the hypothesis $f(x_+) \ge f(x_-)$. Hence, BinarySearchIII can only recursively call itself at most $O(\log \frac{x_+ - x_-}{\varepsilon})$ times. If it calls BinarySearch, then by Lemma 9 this only uses an additional $O(\log \frac{x_+ - x_-}{\varepsilon})$ queries. ■

Lemma 12 *Let $f : \mathbb{R} \to \mathbb{R}$ be 1-smooth. Then,* BinarySearchII *(Algorithm 5) terminates with an ε-stationary point of f using $O(\log \frac{x_+ - x_-}{\varepsilon})$ queries to the oracle.*

Proof First, we check that when BINARYSEARCHII calls itself, the preconditions of BINARYSEARCHII continue to be met. Suppose for instance that $0 \le f(x_-) - f(m) \le \frac{1}{2}(f(x_-) - f(x_+))$. Since $0 \le f(x_-) - f(x_+) \le \frac{\varepsilon}{4}(x_+ - x_-)$ by hypothesis, then

$$0 \le f(x_-) - f(m) \le \frac{\varepsilon}{8}(x_+ - x_-) = \frac{\varepsilon}{4}(x_- - m),$$

which is what we wanted to show. The other case is similar.

Next, we argue that BINARYSEARCHII terminates. The hypotheses of BINARYSEARCHII imply that there is an $\varepsilon/2$-stationary point in the interval $[x_-, x_+]$. Indeed, if this were not the case, then $f' \le -\varepsilon/2$ on the entire interval, so $f(x_+) = f(x_-) + \int_{[x_-,x_+]} f' \le f(x_-) - \frac{\varepsilon}{2}(x_+ - x_-)$, but this contradicts the assumption $f(x_-) - f(x_+) \le \frac{\varepsilon}{4}(x_+ - x_-)$. Therefore, if $x_+ - x_- < \frac{\varepsilon}{2}$, it would follow that $f'(x_-) > -\varepsilon$, which contradicts the hypothesis $f'(x_-) \le -\varepsilon$. Since the value of $x_+ - x_-$ is cut in half each time that BINARYSEARCHII calls itself, we conclude that this can happen at most $O(\log \frac{x_+ - x_-}{\varepsilon})$ times. If BINARYSEARCHII calls either BINARYSEARCH or BINARYSEARCHIII, then by Lemma 9 and Lemma 11, this uses at most an additional $O(\log \frac{x_+ - x_-}{\varepsilon})$ queries to the oracle. ■

Lemma 13 *Let $f : \mathbb{R} \to \mathbb{R}$ be 1-smooth. Then,* DECREASEGAP *(Algorithm 4) terminates, either with an ε-stationary point of f, or with a point x such that $f(x) \le f(x_0)$ and $f(x + 2/\varepsilon) \ge \frac{3}{4} f(x)$, using $O(\log \frac{1}{\varepsilon})$ queries to the oracle.*

Proof Each time DECREASEGAP calls itself, the value of $f(x_0)$ decreases by a factor of $\frac{3}{4}$. If $f'(x_0) \le -\varepsilon$, then from (1) we deduce that $f(x_0) \ge \frac{1}{2}|f'(x_0)|^2 \ge \varepsilon^2/2$. Hence, DECREASEGAP can call itself at most $O(\log \frac{1}{\varepsilon^2}) = O(\log \frac{1}{\varepsilon})$ times. If it calls BINARYSEARCH, then by Lemma 9 this uses an additional $O(\log \frac{1}{\varepsilon})$ queries to the oracle. ■

Finally, we are ready to verify the correctness of ZEROTHORDER (Algorithm 3).

Proof [Proof of Theorem 6] From Lemma 13, if $|f'(x_-)| > \varepsilon$ then we must have $f'(x_-) \le -\varepsilon$ and $f(x_+) \ge \frac{3}{4} f(x_-)$. There are two cases. If $f(x_+) \le f(x_-)$, then we know that

$$0 \le f(x_-) - f(x_+) \le \frac{1}{4} f(x_-) \le \frac{1}{4} = \frac{\varepsilon}{8}(x_+ - x_-)$$

so the preconditions of BINARYSEARCHII are met; by Lemma 12, ZEROTHORDER terminates with an ε-stationary point of f using $O(\log \frac{1}{\varepsilon})$ additional queries. In the other case $f(x_+) \ge f(x_-)$, by Lemma 11, ZEROTHORDER again terminates with an ε-stationary point of f using $O(\log \frac{1}{\varepsilon})$ additional queries. This concludes the proof. ■