Lower Bounds for Finding Stationary Points of Non-Convex, Smooth High-Dimensional Functions*

Yair Carmon yairc@stanford.edu
John C. Duchi jduchi@stanford.edu
Oliver Hinder ohinder@stanford.edu
Aaron Sidford sidford@stanford.edu
Stanford University, Stanford, CA

Abstract

We establish lower bounds on the complexity of finding ϵ-stationary points of smooth, non-convex, high-dimensional functions. For functions with Lipschitz continuous pth derivative, we show that all algorithms—even randomized algorithms observing arbitrarily high-order derivatives—have worst-case iteration count $\Omega(\epsilon^{-(p+1)/p})$. Our results imply that the $O(\epsilon^{-2})$ convergence rate of gradient descent is unimprovable without additional assumptions (e.g. Lipschitz Hessian), and that cubic regularization of Newton's method and pth order regularization in general are similarly optimal. Additionally, we prove that deterministic first-order methods, even applied to arbitrarily smooth functions, cannot achieve convergence rates better than $O(\epsilon^{-8/5})$, which is within $\epsilon^{-1/15}$ of the recently established $\widetilde{O}(\epsilon^{-5/3})$ rate for accelerated gradient descent.

1 Introduction

Finding the global minimum of a general smooth (non-convex) function f is computationally intractable [17, §1.6], and even deciding whether a point x is a local minimum of f is in some cases be NP-complete [16]. Nevertheless, optimization methods designed for minimizing such general smooth functions are in widespread use [22]. The most common measure of performance for such methods is the time required to find an ϵ-stationary point, i.e. a point x such $\|\nabla f(x)\| \le \epsilon$. In this context, stationarity is often—though certainly not always—a good proxy for local optimality [14, 18, 22].

Theoretical guarantees on the number of function/derivative evaluations required to find ϵ-stationary points often feature two appealing characteristics. First, they are polynomial in $1/\epsilon$ and measures of the function's regularity. Second, they do not depend on the dimension of the function's domain. The best-known method with such *dimension-free* rate of convergence is gradient descent [18], which finds an ϵ-stationary point in at most $2L_1\Delta\epsilon^{-2}$ iterations for functions f with L_1-Lipschitz gradient and $f(x^{(0)}) - \inf_x f(x) \le \Delta$.

Developing algorithms with improved ϵ dependence, under different smoothness assumptions, is an area of active research [21, 20, 4, 5, 1, 6, 2]. In this paper we take a complimentary viewpoint, and provide lower bounds on the best ϵ dependence any algorithm can achieve, for several function and algorithm classes. Table 1 summarizes the known upper bounds and our new lower bounds.

Cartis et al. [9, 10, 11] consider lower bounds on the evaluation complexity of specific algorithms and certain structured algorithm classes. In contrast, we present the first lower bounds for finding stationary points of high-dimensional functions applying to (i) all randomized high-order methods (with access to all derivatives of a function f at a query point x) or (ii) all deterministic first-order methods (with access only to gradient and function value). Our results draw deeply from the literature on lower bounds for finding global minimizers of *convex* functions [17, 18, 23].

*This is an extended abstract of a two-part paper sequence [7, 8] with preprints available on arXiv. Throughout, we use PI.k and PII.k to refer to numbered item k in the papers [7] and [8] respectively.

OPTML 2017: 10th NIPS Workshop on *Optimization for Machine Learning* (NIPS 2017).

Table 1: The number of iterations required to find ϵ-stationary points of high dimensional functions

	f has Lipschitz	Upper bound	Lower bound	Gap
A	pth order derivative	$O\left(\epsilon^{-(p+1)/p}\right)$ [4]	$\Omega\left(\epsilon^{-(p+1)/p}\right)$ Thm. 1	$O(1)$
B	gradient and Hessian	$\widetilde{O}\left(\epsilon^{-7/4}\right)$ [6]	$\Omega\left(\epsilon^{-12/7}\right)$ Thm. 2	$\widetilde{O}\left(\epsilon^{-1/28}\right)$
C	qth derivative $\forall q \le p, p \ge 3$	$\widetilde{O}\left(\epsilon^{-5/3}\right)$ [6]	$\Omega\left(\epsilon^{-8/5}\right)$ Thm. 2	$\widetilde{O}\left(\epsilon^{-1/15}\right)$
D	gradient + f convex	$\widetilde{O}\left(\epsilon^{-1}\right)$ †	$\Omega\left(\epsilon^{-1}\right)$ †	$\widetilde{O}\left(1\right)$

Notes The bounds apply for functions f with $f(x^{(0)}) - \inf_x f(x) \le O(1)$ for initialization $x^{(0)}$. In row A the upper bounds are achieved by deterministic pth-order methods and the lower bounds apply to all randomized methods of arbitrary order. In rows B-D the upper (lower) bounds are achieved by (apply to all) deterministic first-order methods. † The bounds in row D differ from the standard rates for convex optimization since they do not assume bounded distance to $\arg\min f$; see Section PII.3 for further discussion.

2 Lower bound framework

Here we provide a condensed version of Sections PI.2 and PI.3 in [7], defining notation and key concepts.

Function classes Measures of function regularity are crucial in the design and analysis of optimization algorithms, and *smoothness* (continuity of derivatives) is particularly important [19, 21, 4]. Accordingly, for every $p \ge 1$ and parameters $\Delta, L_p > 0$, we consider the function class

$$\mathcal{F}_p(\Delta, L_p)$$

consisting of all functions $f : \mathbb{R}^d \to \mathbb{R}$ (for any $d \in \mathbb{N}$) with L_p-Lipschitz continuous pth order derivative and satisfying $f(0) - \inf_x f(x) \le \Delta$. Our results are "dimension-free": for any accuracy ϵ we construct a hard instance $f : \mathbb{R}^d \to \mathbb{R}$ for some $d \in \mathbb{N}$, but d must grow polynomially with $1/\epsilon$.

Oracle model For any dimension $d \in \mathbb{N}$, an *algorithm* A takes $f : \mathbb{R}^d \to \mathbb{R}$ to a sequence of *iterates* in $\mathbb{R}^d$. We let $\mathsf{A}[f] = \{x^{(t)}\}_{t=1}^{\infty}$ denote the sequence $x^{(t)} \in \mathbb{R}^d$ of iterates that A generates when operating on f. We limit the algorithm's access to the function f by means of an *information oracle*, which reveals only the value of f and its derivatives at the queried point. In a *pth-order deterministic algorithm* the ith iterate $x^{(i)}$ is some fixed function of $\{\nabla^q f(x^{(j)}) \mid 0 \le q \le p, 1 \le j < i\}$, where for $q \in \mathbb{N}$, $\nabla^q f(x)$ denotes the qth derivative of f at point x (an order q tensor). We denote the class of pth-order deterministic algorithms by $\mathcal{A}_{\mathsf{det}}^{(p)}$ and let $\mathcal{A}_{\mathsf{det}} = \cup_{p\in\mathbb{N}} \mathcal{A}_{\mathsf{det}}^{(p)}$. Randomized algorithms are mixtures of $\mathsf{A} \in \mathcal{A}_{\mathsf{det}}$ [17].

Worst-case complexity As we consider finding stationary points of f, the natural measure of performance is the number of iterations required to find a point x such that $\|\nabla f(x)\| \le \epsilon$. Thus for a sequence $\{x^{(t)}\}_{t\in\mathbb{N}}$ we define the *complexity of* $\{x^{(t)}\}_{t\in\mathbb{N}}$ *on* f,

$$\mathsf{T}_\epsilon\big(\{x^{(t)}\}_{t\in\mathbb{N}}, f\big) := \inf\left\{t \in \mathbb{N} \mid \left\|\nabla f(x^{(t)})\right\| \le \epsilon\right\}.$$

With mild abuse of notation, we let $\mathsf{T}_\epsilon\big(\mathsf{A}, f\big) := \mathsf{T}_\epsilon\big(\mathsf{A}[f], f\big)$ denote the complexity of A operating on f, and we define the minimax complexity of algorithm class $\mathcal{A}$ operating on function class $\mathcal{F}$ by

$$\mathcal{T}_\epsilon\big(\mathcal{A}, \mathcal{F}\big) := \inf_{\mathsf{A}\in\mathcal{A}} \sup_{f\in\mathcal{F}} \mathsf{T}_\epsilon\big(\mathsf{A}, f\big).$$

With minor extension for randomized algorithms, this is the quantity we bound in Table 1.

Zero-respecting algorithms For a vector $v \in \mathbb{R}^d$ we let $\operatorname{supp}\{v\} := \{i \in \{1, \dots, d\} \mid v_i \ne 0\}$ denote the support (non-zero indices) of v. Extending the definition, for an order k tensor $T \in \mathbb{R}^{\otimes^k d}$, let $\operatorname{supp}\{T\} := \{i \in \{1, \dots, d\} \mid \exists j_1, \dots, j_{k-1} \text{ s.t. } T_{i,j_1,\dots,j_{k-1}} \ne 0\}$. We say an algorithm A is *pth order zero-respecting* if for any $f : \mathbb{R}^d \to \mathbb{R}$ the sequence $\mathsf{A}[f] = \{x^{(t)}\}_{t\in\mathbb{N}}$ satisfies

$$\operatorname{supp}\left\{x^{(t)}\right\} \subseteq \bigcup_{q\in\{1,\dots,p\}} \bigcup_{s<t} \operatorname{supp}\left\{\nabla^q f(x^{(s)})\right\} \quad \text{for each } t \in \mathbb{N}.$$

For $p = 1$, this definition is equivalent to the requirement that $x_j^{(t)} = 0$ for every j such that $\nabla_j f(x^{(s)}) = 0$ for all $s < t$. We denote the class of pth order zero-respecting algorithms by $\mathcal{A}_{\mathsf{zr}}^{(p)}$, and let $\mathcal{A}_{\mathsf{zr}} = \cup_{p\in\mathbb{N}} \mathcal{A}_{\mathsf{zr}}^{(p)}$.

Common first- and second-order optimization methods are zero-respecting [13, 15, 22, 21, 12]. Our notion of zero-respecting algorithms is a generalization of the assumption that first-order methods only query points in the span of the gradients they observe [18, 3]. Zero-respecting algorithms form the backbone of our analysis.

Zero-chains Nesterov [18] introduced the notion of "chain-like" functions, which are instrumental in establishing lower bounds for optimization [18, 3, 23]. We say $f : \mathbb{R}^d \to \mathbb{R}$ is a *pth order zero-chain* if for any $x \in \mathbb{R}^d$ with $\operatorname{supp}\{x\} \subseteq \{1, \dots, i-1\}$ we have $\cup_{q\in[p]} \operatorname{supp}\{\nabla^q f(x)\} \subseteq \{1, \dots, i\}$. Crucially, zero-respecting algorithms must "discover" coordinates of such functions one by one.

Proposition 1. *For any $p \geq 1$, let f be a pth order zero-chain and let $\mathsf{A} \in \mathcal{A}_{\mathsf{zr}}^{(p)}$. Let $\{x^{(t)}\}_{t\in\mathbb{N}} = \mathsf{A}[f]$ be the iterates of A operating on f. Then $x_j^{(t)} = 0$ for every $t \leq d$ and every $j \geq t$.*

Lower bound strategy To lower bound $\mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{zr}}, \mathcal{F}\big)$ for any function class $\mathcal{F}$, we find a function $f : \mathbb{R}^T \to \mathbb{R}$ such that: **(i)** f is a zero-chain, **(ii)** $f \in \mathcal{F}$, and **(iii)** $\|\nabla f(x)\| > \epsilon$ for every $x \in \mathbb{R}^T$ with $x_T = 0$. By Proposition 1, this immediately implies that every zero-respecting algorithm requires at least $T+1$ iterations to find an ϵ-stationary point of f, and so $\mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{zr}}, \mathcal{F}\big) > T$.

From deterministic to zero-respecting algorithms Despite their common vulnerability to zero-chains, zero-respecting algorithms are at least as efficient as deterministic algorithms.

Proposition 2. *For any $p \geq 1$ and any function class $\mathcal{F}$ considered in this paper, $\mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{det}}^{(p)}, \mathcal{F}\big) \geq \mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{zr}}^{(p)}, \mathcal{F}\big)$. Consequently, $\mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{det}}, \mathcal{F}\big) \geq \mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{zr}}, \mathcal{F}\big)$.*

Proposition 2 builds on the classical notion of a resisting oracle with rotation invariance [17, Ch. 7].

3 Lower bounds for high-order methods

We now consider an oracle providing derivatives of arbitrarily high order at the queried point; i.e. algorithm classes $\mathcal{A}_{\mathsf{det}}$ and $\mathcal{A}_{\mathsf{zr}}$. For such algorithms, we show that the hardness of finding stationary points depends fundamentally on the maximum degree of smoothness available, by rescaling a single 'hard' function. For $T \in \mathbb{N}$ and $d \geq T$, we define this (unscaled) hard $\bar{f}_T : \mathbb{R}^d \to \mathbb{R}$ by

$$\bar{f}_T(x) = -\Psi(1)\,\Phi(x_1) + \sum_{i=2}^{T} \left\{\Psi(-x_{i-1})\,\Phi(-x_i) - \Psi(x_{i-1})\,\Phi(x_i)\right\}, \tag{1}$$

where the component functions Ψ and Φ are

$$\Psi(x) := \begin{cases} 0 & x \leq 1/2 \\ \exp\left(1 - \frac{1}{(2x-1)^2}\right) & x > 1/2 \end{cases} \quad \text{and} \quad \Phi(x) = \sqrt{e}\int_{-\infty}^{x} e^{-\frac{1}{2}t^2}\,dt.$$

We illustrate the construction (1) in Fig. P1.1 in [7]. Happily, $\bar{f}_T$ fulfills the three requirements of the lower bound strategy we outlined in Section 2 (see Lemmas P1.2 and P1.3).

Lemma 1. *For any $T \in \mathbb{N}$, the function $\bar{f}_T$ satisfies the following.*

i. ***Zero-chain*** *For every $p \geq 1$, $\bar{f}_T$ is a pth order zero-chain.*

ii. ***Membership in function class*** *$\bar{f}_T \in \mathcal{F}\left(12T, \ell_p\right)$ for every $p \geq 1$, where $\ell_p \leq e^{2.5p\log p + cp}$ for some numerical constant $c < \infty$.*

iii. ***Large gradient*** *For every $x \in \mathbb{R}^T$ such that $|x_i| < 1$ for some $i \in \{1, \dots, T\}$,*

$$\left\|\nabla \bar{f}_T(x)\right\| \geq \left\|\nabla \bar{f}_T(x)\right\|_\infty > 1.$$

For general accuracy ϵ, value bound Δ and pth order Lipschitz parameter L_p, we establish our lower bounds by the following scaling argument. Let $f(x) = (L_p/\ell_p)\sigma^{p+1}\bar{f}_T(x/\sigma)$, where $\sigma > 0$ is a scale parameter to be determined. By Lemma 1.ii, taking $T = \left\lfloor \Delta / \left(12(L_p/\ell_p)\sigma^{p+1}\right)\right\rfloor$ guarantees $f \in \mathcal{F}_p(\Delta, L_p)$ for any σ. By Proposition 1 and Lemma 1.i, every $\mathsf{A} \in \mathcal{A}_{\mathsf{zr}}$ operating on f produces iterates $x^{(1)}, x^{(2)}, \dots$ such that $x_T^{(t)} = 0$ for every $t \leq T$. Consequently, Lemma 1.iii guarantees that $\left\|\nabla f(x^{(t)})\right\| > (L_p/\ell_p)\sigma^p$ for every $t \leq T$. Choosing σ such that $(L_p/\ell_p)\sigma^p = \epsilon$ and substituting back to our choice of T establishes the following result (where we invoke Proposition 2).

Theorem 1. *There exists a numerical constant $c < \infty$ and $\ell_p \le e^{2.5p \log p + cp}$ such that the following lower bound holds for all $p \in \mathbb{N}$. Let Δ, L_p, and ϵ be positive. Then*

$$\mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{det}}, \mathcal{F}_p(\Delta, L_p)\big) \ge \mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{zr}}, \mathcal{F}_p(\Delta, L_p)\big) \ge \frac{\Delta}{c}\left(\frac{L_p}{\ell_p}\right)^{1/p} \epsilon^{-\frac{1+p}{p}}.$$

Lower bounds for randomized algorithms Adopting the approach of Woodworth and Srebro [23], we can extend Theorem 1 to the class $\mathcal{A}_{\mathsf{rand}}$ of randomized local algorithms, establishing that $\mathcal{T}_\epsilon\big(\mathcal{A}_{\mathsf{rand}}, \mathcal{F}_p(\Delta, L_p)\big) \gtrsim \Delta L_p^{1/p} \epsilon^{-(1+p)/p}$ with an appropriate probabilistic definition of randomized complexity; see Section PI.5. Unfortunately, this technique cannot distinguish high-order methods from first-order methods; it either applies to every algorithm in $\mathcal{A}_{\mathsf{rand}}$ or not at all (see Section PII.6.2). Therefore, we restrict our discussion in the following section to deterministic (or randomized, zero-respecting) first-order methods.

4 Lower bounds for first-order methods

Given the scalability of first-order methods to high dimensions, it is important to understand their complexity. Consequently, we turn to an oracle model providing only function values and gradients at the queried point and investigate algorithm classes $\mathcal{A}^{(1)}_{\mathsf{det}}$ and $\mathcal{A}^{(1)}_{\mathsf{zr}}$. To make the lower bounds more powerful, we consider the more restricted function class

$$\mathcal{F}_{1:p}(\Delta, L_1, ..., L_p) := \cap_{q \in \{1,...,p\}} \mathcal{F}_q(\Delta, L_q).$$

Recent work [1, 5, 6] proposes first-order methods with improved performance guarantees for functions in these classes (see Table 1). Here we provide nearly matching lower bounds.

As in the previous section, we define an unscaled function that is particularly challenging for zero-respecting algorithms, which depends on parameters $T \in \mathbb{N}$, $\mu \le 1$ and $r \ge 1$:

$$\bar{f}_{T,\mu,r}(x) := \frac{\sqrt{\mu}}{2}(x_1 - 1)^2 + \frac{1}{2}\sum_{i=1}^{T}(x_{i+1} - x_i)^2 + \mu \sum_{i=1}^{T} \Upsilon_r(x_i), \tag{2}$$

where the function Υ_r is

$$\Upsilon_r(x) = 120 \int_1^x \frac{t^2(t-1)}{1 + (t/r)^2} dt.$$

We illustrate our construction (2) in Figure PII.1 in [8]; it is a sum of a quadratic chain, proposed by Nesterov [18] for lower bounds in convex optimization, and a separable non-convex function carefully chosen to make the gradient of $\bar{f}_{T,\mu,r}$ larger. We now establish the three components of our lower bound strategy (see Lemmas PII.3–PII.4).

Lemma 2. *For any $T \in \mathbb{N}$, $\mu \in [0, 1]$ and $r \ge 1$, the function $\bar{f}_{T,\mu,r}$ satisfies the following.*

i. ***Zero-chain*** *$\bar{f}_T$ is a first-order zero-chain.*

ii. ***Membership in function class*** *$\bar{f}_{T,\mu,r} \in \mathcal{F}_p\left(\frac{1}{2}\sqrt{\mu} + 10\mu T, (1_{(p=1)} + r^{3-p}\mu)\ell_p\right)$ for any $p \ge 1$, where $\ell_p \le e^{1.5p \log p + cp}$ for some numerical constant $c < \infty$.*

iii. ***Large gradient*** *For every $x \in \mathbb{R}^{T+1}$ such that $x_T = x_{T+1} = 0$, $\left\|\nabla \bar{f}_{T,\mu,r}(x)\right\| > \mu^{3/4}/4$.*

The scaling argument used to establish lower bounds for arbitrary $\epsilon, \Delta, L_1, \ldots, L_p > 0$ is similar to the one we sketch for Theorem 1, but is more involved as we must choose additional parameters (μ and r) and satisfy multiple Lipschitz constraints; see Section PII.5.2. We conclude with our final lower bound.

Theorem 2. *There exists a numerical constant $c < \infty$ and $\ell_q \le e^{\frac{3q}{2}\log q + cq}$ such that the following lower bound holds. Let $p \ge 2$, $p \in \mathbb{N}$, and let $\Delta, L_1, L_2, \ldots, L_p, \epsilon$ be positive. Assume additionally that $\epsilon \le (L_1^q/L_q)^{1/(q-1)}$ for each $q \in \{2, \ldots, p\}$. Then*

$$\mathcal{T}_\epsilon\big(\mathcal{A}^{(1)}_{\mathsf{det}} \cup \mathcal{A}^{(1)}_{\mathsf{zr}}, \mathcal{F}_{1:p}(\Delta, L_1, ..., L_p)\big) \ge \frac{\Delta}{c} \begin{cases} \min_{2 \le q \le p} \left\{ \left(\frac{L_1}{\ell_1}\right)^{\frac{3q-5}{5(q-1)}} \left(\frac{L_q}{\ell_q}\right)^{\frac{2}{5(q-1)}} \right\} \epsilon^{-8/5} & p \ge 3 \\ \left(\frac{L_1}{\ell_1}\right)^{\frac{3}{7}} \left(\frac{L_2}{\ell_2}\right)^{\frac{2}{7}} \epsilon^{-12/7} & p = 2. \end{cases}$$

References

[1] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the Forty-Ninth Annual ACM Symposium on the Theory of Computing*, 2017. URL https://arxiv.org/abs/1611.01146.

[2] Z. Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. *arXiv:1708.08694 [math.OC]*, 2017.

[3] Y. Arjevani, S. Shalev-Shwartz, and O. Shamir. On lower and upper bounds in smooth and strongly convex optimization. *Journal of Machine Learning Research*, 17(126):1–51, 2016. URL http://jmlr.org/papers/v17/15-106.html.

[4] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1–2):359–368, 2017.

[5] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for non-convex optimization. *arXiv:1611.00756 [math.OC]*, 2016.

[6] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Convex until proven guilty: dimension-free acceleration of gradient descent on non-convex functions. In *Proceedings of the 33rd International Conference on Machine Learning*, 2017.

[7] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points I. *arXiv: 1710.11606 [math.OC]*, 2017. URL https://arxiv.org/pdf/1710.11606.pdf.

[8] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points II: First-order methods. *arXiv: 1711.00841 [math.OC]*, 2017. URL https://arxiv.org/pdf/1711.00841.pdf.

[9] C. Cartis, N. I. Gould, and P. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *SIAM Journal on Optimization*, 20(6):2833–2852, 2010.

[10] C. Cartis, N. I. Gould, and P. L. Toint. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012.

[11] C. Cartis, N. I. M. Gould, and P. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. *arXiv:1709.07180 [math.OC]*, 2017.

[12] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. MPS-SIAM Series on Optimization. SIAM, 2000.

[13] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.

[14] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent converges to minimizers. In *Proceedings of the Twenty Ninth Annual Conference on Computational Learning Theory*, 2016.

[15] D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[16] K. Murty and S. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[17] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

[18] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

[19] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[20] Y. Nesterov. How to make the gradients small. *Optima*, 88, 2012.

[21] Y. Nesterov and B. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming, Series A*, 108:177–205, 2006.

[22] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[23] B. E. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems 29*, pages 3639–3647, 2016.