DADA: Dual Averaging with Distance Adaptation

Mohammad Moshtaghifar* Anton Rodomanov† Daniil Vankov‡
Sebastian U. Stich§

October 18, 2024

Abstract

We present a novel parameter-free universal gradient method for solving convex optimization problems. Our algorithm—Dual Averaging with Distance Adaptation (DADA)–is based on the classical scheme of dual averaging and dynamically adjusts its coefficients based on the observed gradients and the distance between its iterates to the starting point, without the needing to know any problem-specific parameters. DADA is a universal algorithm that simultaneously works for a wide range of problem classes as long as one is able to bound the local growth of the objective around its minimizer. Particular examples of such problem classes are nonsmooth Lipschitz functions, Lipschitz-smooth functions, Hölder-smooth functions, functions with high-order Lipschitz derivative, quasi-self-concordant functions, and (L_0, L_1)-smooth functions. Furthermore, in contrast to many existing methods, DADA is suitable not only for unconstrained problems, but also constrained ones, possibly with unbounded domain, and it does not require fixing neither the number of iterations nor the accuracy in advance.

Keywords: Convex Optimization, Gradient Methods, Adaptive Algorithms, Parameter-Free Methods, Dual Averaging, Distance Adaption, Universal Methods, Worst-Case Complexity Guarantees

1 Introduction

We consider the following optimization problem:

$$\min_{x \in Q} f(x), \tag{1.1}$$

where $Q \subseteq \mathbb{R}^d$ is a simple and nonempty closed convex set and $f\colon \mathbb{R}^d \to \mathbb{R}$ is a convex function on Q. We assume that this problem has a solution which we denote by x^*. Recall that we assume Q to be simple, which means computability of exact optimal solutions to both minimization problems in 1.1. This setting has received extensive attention due to its widespread applications in modern machine learning and data-driven optimization [1].

*Sharif University of Technology. E-mail: m.moshtaghi@sharif.edu.
†CISPA Helmholtz Center for Information Security. E-mail: anton.rodomanov@cispa.de.
‡Arizona State University. E-mail: dvankov@asu.edu.
§CISPA Helmholtz Center for Information Security. E-mail: stich@cispa.de.

One of the key challenges in solving these problems using gradient-based methods is selecting appropriate hyperparameters, particularly stepsizes, which significantly impact convergence and performance. Traditional methods require extensive hyperparameter tuning, a time-consuming and resource-intensive process. To address this, there has been a growing interest in so-called parameter-free algorithms, which aim to eliminate the need for manual tuning.

Typically, line search techniques have been used to select step sizes in optimization, and they work well for certain function classes, such as Hölder-smooth problems [12]. However, in recent years, several parameter-free approaches have been developed which do not utilize line-search methods. Notably, the classical Subgradient Descent method has been adapted through strategies like bisection search [2] and dynamic step size schedules based on dual averaging [3, 10]. For example, two recent approaches [6, 7] dynamically adjust stepsizes based on estimates of the initial distance to the optimal solution, $D_0 = \|x_0 - x^*\|$. However, these methods often have limitations, such as requiring bounded domain assumptions [7], lacking applicability to constrained optimization problems [3, 10], or being difficult to extend to function classes beyond smooth and nonsmooth functions[6].

In contrast, our method is universal and applicable to a wide variety of problem classes, those not covered by other methods. While existing methods might provide results for nonsmooth and Lipschitz-smooth problems, they are limited beyond that scope. Even for Hölder-smooth problems, very few results are known [8]. Our method, however, not only covers Hölder-smooth problems but also extends to many other problem classes, making it a more flexible and powerful solution.

Contributions. In this paper, we introduce DADA—Dual Averaging with Distance Adaptation—a novel parameter-free optimization algorithm that is universal for solving constrained optimization problems of the form (1.1). DADA is based on the classical Dual Averaging (DA) scheme [11], but with a specially designed, dynamically adjusted estimate of $D_0 = \|x_0 - x^*\|$, leveraging recent techniques inspired by DoG [6] and related works [2, 7]. DADA dynamically adapts step sizes based on these distance estimates, without requiring prior knowledge of problem-specific parameters. Furthermore, our approach applies to both unconstrained problems and problems with simple constraints, whose domains are not required to be bounded, making it a powerful tool across a wide range of applications.

In Section 2 we provide the problem formulation and provide its detailed theoretical analysis. We present our method and outline its foundational structure based on the Dual Averaging scheme [11]. By bounding the growth function, as stated in Theorem 2.1, we derive convergence guarantees that apply to a broad range of function classes. This provides a rigorous basis for understanding the behavior and capability of DADA across diverse problem settings.

To demonstrate the versatility and effectiveness of DADA, in Section 3 we provide convergence estimates across various function classes. Of particular interest are three important classes: Quasi-Self-Concordant (QSC) functions, functions with Lipschitz pth derivative, and (L_0, L_1)-smooth functions. This highlights DADA's ability to deliver competitive performance without needing class-specific adaptations.

Notation. In this text, we work in the space $\mathbb{R}^d$ equipped with the standard inner product $\langle \cdot, \cdot \rangle$ and the general Euclidean (Mahalanobis) norm:

$$\|x\| \coloneqq \langle Bx, x \rangle^{1/2}, \qquad x \in \mathbb{R}^d,$$

where $A \in S^d$ is positive-definite. The corresponding dual norm is defined in the standard way:

$$\|s\|_* \coloneqq \max_{\|x\|=1} \langle s, x \rangle = \langle s, B^{-1}s \rangle^{1/2}, \qquad s \in \mathbb{R}^d.$$

Thus, for any $s, x \in \mathbb{R}^d$, we have Cauchy–Schwarz inequality $|\langle s, x \rangle| \leq \|s\|_* \|x\|$. For a convex function $f \colon \mathbb{R}^d \to \mathbb{R}$, we denote its subdifferential at a point $x \in \mathbb{R}^d$ by $\partial f(x)$.

2 Main Algorithm: Dual Averaging with Distance Adaptation

Measuring the quality of solution. Rather than focusing on bounding the distance to the optimal point x^*, this work focuses on bounding the following function $v(\cdot)$. Let $\bar{x} \in \mathbb{R}^d$ be an arbitrary vector, then for $x \in \mathbb{R}^d$ with a subgradient $\nabla f(x)$ such that, $\|\nabla f(x)\|_* \neq 0$, define

$$v(x) = \frac{1}{\|\nabla f(x)\|_*} \langle \nabla f(x), x - x^* \rangle.$$

This quantity can be interpreted as the distance between the point x^* and the hyperplane $H \coloneqq \{y : \langle \nabla f(x), x - y \rangle\}$ as discussed in [13]. Additionally, we define $v_i = v(x_i)$.

Next, we introduce a function that measures the local growth of f around the solution x^*. For $t \geq 0$, we define

$$\omega(t) = \max_x \{ f(x) - f(x^*) : \|x - x^*\| \leq t \}.$$

For convenience, we also define $\omega(t) = 0$ if $t < 0$. By bounding the $\omega(t)$ we can derive convergence rate estimates that simultaneously apply to a broad range of problem classes. See Section 3 for more details. A useful relation between the function residual $f(x) - f^*$ and the pair of ω and v functions is demonstrated by the following inequality, which allows us to express one via the other:

$$f(x) - f^* \leq \omega(v(x)).$$

This is a simple property of convex functions which is discussed in [13, Section 3.2.2], see also Section B for the explicit proof which we present for the reader's convenience.

It is important to note that both $v(x)$ and $\omega(t)$ are standard quantities commonly used in the optimization literature [13, 14].

Algorithm 2.1 General Dual Averaging Scheme

Input: $x_0 \in Q$, $T \geq 1$, nonnegative coefficients $(a_k)_{k=0}^{T-1}$, $(\beta_k)_{k=1}^{T}$ with β_k nondecreasing
for $k = 1, \ldots, T$ **do**
Compute arbitrary $g_k \in \partial f(x_k)$
$x_k = \operatorname{argmin}_{x \in Q} \left\{ \psi_k(x) = \sum_{i=0}^{k-1} a_i \langle g_i, x - x_i \rangle + \frac{\beta_k}{2} \|x - x_0\|^2 \right\}$
Output: $x_T^* = \operatorname{argmin}_{x \in \{x_0, \ldots, x_T\}} f(x)$

The method. Our proposed approach is based on the general Dual Averaging (DA) scheme [11] shown in Section 2. In our approach, we utilize the following time-varying coefficients:

$$\boxed{a_k = \frac{\bar{r}_k}{\|g_k\|_*}, \quad \beta_k = 2\sqrt{k+1}}, \qquad \text{where} \quad \bar{r}_k = \max\left\{ \max_{1 \leq t \leq k} r_t, \bar{r} \right\}, \; r_t = \|x_0 - x_t\|, \tag{2.1}$$

and $\bar{r}$ is a certain user-specified parameter. In what follows, we assume w.l.o.g. that $g_k \neq 0$ for all $k \geq 0$ since otherwise the exact solution has been found and the method could be successfully terminated.

We utilize DA in our method because it allows the use of time-varying coefficients, as defined in (2.1). While it would be possible to use a standard (sub)gradient method, doing so would require fixing the number of iterations in advance, which can be restrictive, or paying an additional $\log(T)$ factor in the convergence rate [13].

The classical DA method has two primary variants. The first, Simple DA, uses a constant coefficient $a_i = D$ while the second, Weighted DA, instead of using a constant sequence, adjusts the coefficients using $a_i = \frac{D}{\|g_i\|_*}$. Both variants, however, require prior knowledge of the parameter D (the initial distance between the starting point and the solution), which is often unknown in practice. To address this limitation, in Section 2 we propose a dynamic sequence for $(a_i)_{i=0}^{\infty}$ and $(\beta_i)_{i=1}^{\infty}$, thereby eliminating the need for the parameter D.

Our method estimates the parameter D by dynamically calculating the distance from the initial point x_0, represented by $\bar{r}_t$. This idea has been recently explored in recent works [6, 7], which similarly utilize $\bar{r}_t$ in various ways. Other methods also attempt to estimate this quantity using alternative strategies, such as bisection search [2] or by using different variations of DA [3, 10].

As discussed earlier, our goal is to bound the function $v(\cdot)$ to establish the convergence of our method. Using $v_t = v(x_t)$, we present the following convergence result for our method (see Section D).

Theorem 2.1. *Consider Section 2 for solving problem* (1.1) *using the coefficients from* (2.1)*. Then,*

$$v_T^* \leq \frac{6R}{\sqrt{T}} \left(\frac{8R}{\bar{r}} \right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}},$$

where $v_T^* = \min_{0 \leq t \leq T} v_t$ *and* $R = \max\{\|x_0 - x^*\|, \bar{r}\}$*. Furthermore,* $f(\bar{x}_T) - f^* \leq \omega(v_T^*)$*. To make* $v_T^* \leq \delta$ *for any* $\delta > 0$*, it suffices to make* $T = \max\left\{ \log \frac{8R}{\bar{r}}, \frac{36e^2R^2}{\delta^2} \log^2 \frac{8eR}{\bar{r}} \right\}$ *iterations.*

First, we apply a standard result from DA (Theorem C.1), which holds for any choice of coefficients a_k and β_k. This result establishes a general bound that forms the foundation of our analysis:

$$\forall_{k\leq T}: \quad \sum_{i=0}^{k-1} a_i v_i \|g_i\|_* + \frac{\beta_k}{2}\|x_k - x^*\|^2 \leq \frac{\beta_k}{2} D_0^2 + \sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i}\|g_i\|_*^2, \tag{2.2}$$

where $D_0 = \|x_0 - x^*\|$. Next, we introduce specific time-varying sequences for a_k and β_k as defined by our method in (2.1). Since the classical DA method [11] requires prior knowledge of D_0 to define a_k, we estimate this quantity using $\bar{r}_k$ which originally introduced in [2, 6]. After this step, substituting into (2.2), we obtain the following result:

$$\forall_{k\leq T}: \quad \sum_{i=0}^{k-1} \bar{r}_i v_i + \|x_k - x^*\|^2\sqrt{k} \leq D_0^2\sqrt{k} + (1/4)\sum_{i=0}^{k-1} \frac{\bar{r}_i^2}{\sqrt{i+1}} \leq D_0^2\sqrt{k} + \frac{\bar{r}_{k-1}^2}{2}\sqrt{k}. \tag{2.3}$$

Using the fact that $\bar{r}_i v_i > 0$ for all $0 \leq i \leq k-1$, we can show by induction that $\bar{r}_k$ is bounded by $R = \max\{D_0, \bar{r}\}$ up to a constant factor (see Lemma D.2):

$$\bar{r}_k \leq 8R.$$

This bound is crucial to our analysis, as we need to eliminate $\bar{r}_k$ from the right-hand side of (2.3). Achieving this requires selecting the coefficients precisely as defined in (2.1), which is the primary difference compared to the standard DA method [11]. Next, using the following inequality $\|x_0 - x^*\|^2 - \|x_k - x^*\|^2 \leq 2\|x_k - x_0\|D_0$, we derive the next result for all $k \leq T$:

$$\sum_{i=0}^{k-1} \bar{r}_i v_i \leq \bar{r}_k(2R + \tfrac{1}{2}\bar{r}_k)\sqrt{k} \leq 6\bar{r}_k R\sqrt{k}.$$

After establishing this, the rest of the proof follows straightforwardly by dividing both sides with $\sum_{i=0}^{k-1} \bar{r}_i$ and then using the following inequality:

$$\min_{0\leq t\leq T} \frac{\bar{r}_t}{\sum_{i=0}^{t-1} \bar{r}_i} \leq \frac{(\frac{\bar{r}_T}{\bar{r}_1})^{\frac{1}{T}} \log \frac{e\bar{r}_T}{\bar{r}}}{T}.$$

The detailed proof can be found in Section D.

At this point, we clarify the key differences between our method and approaches like DoG [6]. One obvious difference is that we use DA instead of the classical subgradient method employed by DoG. However, the most important distinction lies in how we handle the sequence of gradients. In DoG, the gradients $\|g_k\|_*$ are accumulated inside β_k, whereas in our approach, they are incorporated into a_k.

This difference is crucial because it allows us to solve the problem without requiring the summation to involve the last gradient, which was an issue in previous analyses. Additionally, this modification makes our method universal, enabling it to work with the growth function ω, which is not known to be the case for DoG, even for deterministic problems.

2.1 Comparison with Recent Parameter-Free Methods.

In this section we mainly take four recent methods on parameter-free into account, DoG [6], DoWG [7], D-Adaptation [3] and Prodigy [10].

Comparison with DoG/DoWG. Both DoG and DoWG employ a similar approach to estimate D_0 in their methods and achieve comparable convergence rates in the smooth and nonsmooth cases. However, neither of them extends to the important cases explored in this paper. Additionally, like our approach, the DoWG method considers only the deterministic case, but with an additional assumption on a bounded domain. They have a different definition of universality, considering only smooth and nonsmooth settings, where in this work by universality we mean using the bound we provided for gorowth function ω, our method converges with the nearly optimal rate for a broad range of convex functions, including Hölder-smooth functions, functions with high-order Lipschitz derivative, quasi-self-concordant functions, and (L_0, L_1)-smooth functions.

It is also worth noting that, as presented in [6], our analysis can be straightforwardly extended to star-convex functions [15] and quasarconvex functions [5], since convexity is used only in the final step of Lemma B.2 via the following inequality:

$$f^* \geq f(x) + \langle g(x), x^* - x \rangle.$$

This can be easily replaced by

$$f^* \geq f(x) + c\langle g(x), x^* - x \rangle,$$

for some $c < \infty$.

Finally, we note that DoG also provides guarantees in the stochastic setting, given that gradients are locally bounded with a known constant. In this work, however, we focus exclusively on the deterministic setting.

Comparison with D-Adaptation/Prodigy. D-Adaptation and Prodigy are similar to our method in their use of Dual Averaging; however, their approaches cannot be extended to the constrained optimization setting and are limited to Lipschitz functions. Nonetheless, their methods yielded notable results in experiments, demonstrating strong empirical performance.

3 Universality of DADA: Examples of Applications

Let us demonstrate that our method is universal in the sense that it simultaneously works for several interesting problem classes without the need for choosing different parameters for each of these functions classes. For simplicity, we assume that $\nabla f(x^*) = 0$ (this happens, in particular, when our problem (1.1) is unconstrained). In what follows, the ϵ-accuracy is measured in terms of the function residual and we also use $\log_+ t \coloneqq 1 + \log t$ to simplify the notation.

Nonsmooth Lipschitz functions. This function class is defined by the inequality $\|\nabla f(x)\|_* \leq L_0$ for all $x \in Q$. For this problem class, DADA requires at most

$$O\left(\frac{L_0^2 R^2}{\epsilon^2} \log_+^2 \frac{R}{r}\right)$$

oracle calls to reach ϵ-accuracy (see Section E.1), which is the standard complexity in this case, up to an extra logarithmic factor [11, 13]. This logarithmic factor is common for all distance-adaptation methods [3, 6, 7, 10].

Lipschitz-smooth functions. Another important class of functions are those with Lipschitz gradient: $\|\nabla f(x) - \nabla f(y)\|_* \leq L_1 \|x - y\|$ for all $x, y \in Q$. In this case, the complexity of our method is

$$O\left(\frac{L_1 R^2}{\epsilon} \log_+^2 \frac{R}{\bar{r}}\right)$$

oracle calls (see Section E.2), which aligns with standard results for Lipschitz-smooth functions, up to the extra logarithmic factor [13, Section 3]. As before, the logarithmic factor is due to the parameter-free nature of the method.

In contrast, the line-search approach discussed in Nesterov's paper [12] achieves a complexity bound of $O\big(\frac{L_1 D_0^2}{\epsilon} + \log \frac{L_1}{T_0}\big)$, where T_0 is equivalent to L_0–the initial point in the line-search method described by Nesterov. We avoid using L_0 here to prevent possible confusion for the reader. This complexity is similar to ours, with the difference that they have an additive logarithmic factor in their rate instead of multiplicative, which is comparatively better. However, line-search approach often more expensive in practice.

Hölder-smooth functions. The previous two functions classes are subclasses of the more general class of Hölder-smooth functions. It is defined by the following inequality: $\|\nabla f(x) - \nabla f(y)\|_* \leq H_\nu \|x - y\|^\nu$ for all $x, y \in Q$, where $\nu \in [0, 1]$ and $H_\nu \geq 0$. Therefore, for $\nu = 0$, we get functions with bounded variation of subgradients (which contains all Lipschitz functions) and for $\nu = 1$ we get L_1-smooth functions.

As shown in Section E.3, our method requires at most

$$O\left(\left[\frac{H_\nu}{\epsilon}\right]^{\frac{2}{1+\nu}} R^2 \log_+^2 \frac{R}{\bar{r}}\right)$$

iterations to achieve ϵ-accuracy, which is close to $O\big(\big[\frac{H_\nu}{\epsilon}\big]^{\frac{2}{1+\nu}} D_0^2 + \log \frac{[H_\nu^2 (\frac{1}{\epsilon})^{1-\nu}]^{\frac{1}{1+\nu}}}{L_0}\big)$, the result achieved by the universal gradient method [12]. However, using the accelerated method results in a better complexity of $O\big(\big[\frac{H_\nu D_0^{1+\nu}}{\epsilon}\big]^{\frac{2}{1+3\nu}} + \log \frac{[D_0^{1-\nu} \big(\frac{1}{\epsilon}\big)^{3(1-\nu)} H_\nu^3]^{\frac{1}{1+3\nu}}}{L_0}\big)$. It is important to note that both of these complexity bounds are derived under the additional assumption that L_0 is sufficiently small (as stated in [12, Theorems 1, 2, 3]). For simplicity, to ensure a straightforward and fair comparison, we therefore also assume that, in the above bounds, $L_0 \leq \gamma(H_\nu, \epsilon) = \big[\frac{1}{\epsilon}\big]^{\frac{1-\nu}{1+\nu}} H_\nu^{\frac{2}{1+\nu}}$ and $\bar{r} \leq D_0$ which means, in particular, that $R = D_0$.

Functions with Lipschitz high-order derivative. Functions in this class have the property that their pth derivative ($p \geq 1$) is Lipschitz, i.e.,

$$\|\nabla^p f(x) - \nabla^p f(y)\|_* \leq L_p \|x - y\|$$

for all $x, y \in \mathbb{R}^d$. For example, p-th power of Euclidean norm [18] and the soft-max $f(x) = \mu \log(\sum_{i=1}^k e^{[\langle a_i, x\rangle + b_i]/\mu})$ are examples of functions in this class.

This class generalizes the Lipschitz-smooth class, and the complexity of our method to achieve ϵ-accuracy is quite similar to this cases. DADA requires at most

$$O\left(\max\left\{\max_{2\leq i\leq p}\left[\frac{p}{i!}\frac{\|\nabla^i f(x^*)\|_*}{\epsilon}\right]^{\frac{2}{i}}, \left[\frac{p}{(p+1)!}\frac{L_p}{\epsilon}\right]^{\frac{2}{p+1}}\right\} R^2 \log_+^2 \frac{R}{\bar{r}}\right),$$

first-order oracle calls to achieve ϵ-accuracy in terms of the function residual. Although line-search gradient methods might be better for Hölder-smooth problems, they are not known to work on this class of function. Detailed proofs and bounds for this class of functions are provided in Section E.4.

Quasi-self-concordant (QSC) functions. A convex function f is said to be QSC with parameter $M \geq 0$, if for any $x \in \mathbb{R}^d$ and arbitrary directions $u, v \in \mathbb{R}^d$ it holds that

$$\nabla^3 f(x)[u, u, v] \leq M \langle \nabla^2 f(x) u, u\rangle \|v\|. \tag{3.1}$$

For example, the soft-max $f(x) = \mu \log(\sum_{i=1}^k e^{[\langle a_i, x\rangle + b_i]/\mu})$ and exponential functions are QSC. For more details and other examples, see [4]. Our method guarantees convergence for QSC functions with the following complexity:

$$O\left(\frac{\|\nabla^2 f(x^*)\|_* R^2}{\epsilon} \log_+^2 \frac{R}{\bar{r}} + (MR)^2 \log_+^2 \frac{R}{\bar{r}} + \log_+ \frac{R}{\bar{r}}\right).$$

In terms of comparisons, second-order methods, such as those explored in [4], are more powerful for minimizing QSC functions, as they leverage additional curvature information. Their complexity bound is $O(MD_0 \log \frac{1}{\epsilon} + \log \frac{D_0 g_0}{\epsilon(f(x_0) - f^*)})$ [4, Corollary 3.4].

However, to our knowledge, this class has not been studied before in the context of first-order methods. The only other first-order methods for which one can prove similar bounds are nonadaptive variants of our scheme, namely the normalized gradient method from [13, Section 5] and the recent variant of this method for constrained problems [14].

(L_0, L_1)-smooth functions. As introduced in [20], a function f is said to be (L_0, L_1)-smooth if there exist constants $L_0 > 0$ and $L_1 \geq 0$ such that for all $x \in \mathbb{R}^n$, we have $\|\nabla^2 f(x)\|_* \leq L_0 + L_1 \|\nabla f(x)\|_*$. The complexity of our method to achieve ϵ-accuracy in this case is

$$O\left(\frac{L_0 R^2}{\epsilon} \log_+^2 \frac{R}{\bar{r}} + (L_1 R)^2 \log_+^2 \frac{R}{\bar{r}} + \log_+ \frac{R}{\bar{r}}\right).$$

This complexity estimate is significantly better than those presented in [16, 20], where the complexity is $\tilde{O}\left(\frac{L_0}{\epsilon^2}\right)$, implying that our method performs much better in this setting.

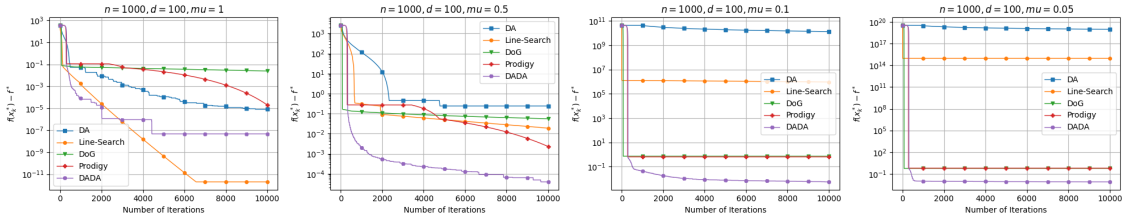


Figure 4.1: Comparison of different methods on the log-sum-exp function with different optimal points x^*.

4 Experiments

To evaluate the performance of our proposed method, DADA, we conduct a series of experiments on convex optimization problems. Our goal is to demonstrate the effectiveness of DADA in achieving competitive convergence rates across various function classes, including Hölder smooth, and (L_0, L_1)-smooth functions. We compare DADA against several parameter-free optimization algorithms, such as DoG [6] and Prodigy [10], highlighting its ability to adapt dynamically without relying on hyperparameter tuning. The experiments also explore the convergence of our method's dynamic distance-based step size to the true value of initial distance D_0.

Note that, the staircase-like pattern in the plots is because, in our algorithm, instead of averaging the x_is, we used the argmin operation. Therefore, we plot $f(x_k^*) - f^*$ where $x_k^* = \text{argmin}_{i \le k} f(x_i)$ for all methods to have a fair comparison. Throughout these experiments, we chose our initial guess for the true distance, denoted as $\bar{r}$, as follows: $\bar{r} = 10^{-6}(1 + \|x_0\|)$. Since DoG uses a similar parameter, we include Fig.4.3 to illustrate the difference between our method and DoG for various values of $\bar{r}$ [1].

Log-sum-exp function. As an additional example of convex functions, we consider the log-sum-exp function:

$$\min_{x \in \mathbb{R}^d} f(x) := \mu \log \left(\sum_{i=1}^{n} \exp \left[\frac{a_i^T x - b_i}{\mu} \right] \right).$$

To generate the problem's variables, we follow these steps: First, as in the previous section, we sample x^* uniformly from the sphere ofradius R centered at the origin. Next, we generate i.i.d. vectors a_i with components uniformly distributed in the interval $[-1, 1]$ for $i = 2, \ldots, n$, and similarly for the scalar values b_i. To ensure that x^* is the minimizer of f, we set a_1 and b_1 such that $\nabla f(x^*) = 0$. Specifically, we define a_1 and b_1 as follows:

$$a_1 = -\sum_{i=2}^{n} \exp \left[\frac{a_i^T x^* - b_i}{\mu} \right] a_i, \qquad b_1 = a_1^T x^*, \tag{4.1}$$

which ensures that $\nabla f(x^*) = 0$. The results are shown in Fig. 4.1, where we fix $n = 10^3$, $d = 10^2$, and $R = 1$ with the starting point $x_0 = 0$. We plot the total number of iterations against the function residual for different values of $\mu \in \{1, 0.5, 0.1, 0.05\}$. Additionally,

[1] This value corresponds to r_ϵ in [6]

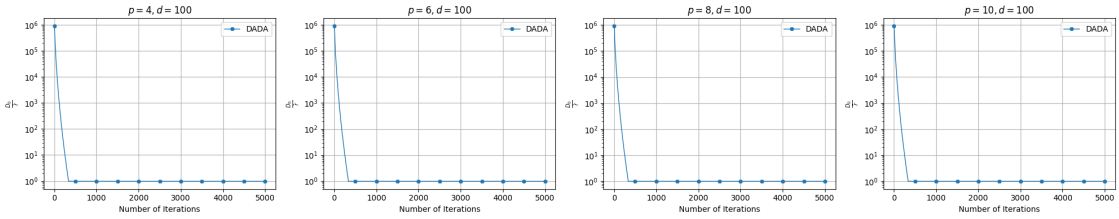


Figure 4.2: The ratio $\frac{D}{\bar{r}_t}$ for the log-sum-exp function with different optimal points x^*.

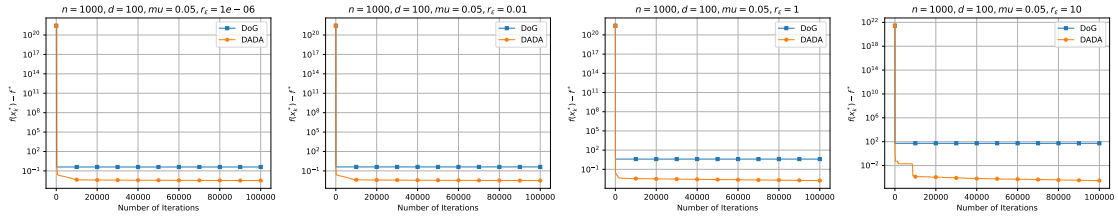


Figure 4.3: Compare to DoG method on the log-sum-exp function with different initial guesses $\bar{r}$.

Fig. 4.2 illustrates the difference between $\bar{r}$ and D_0, highlighting our estimation error at each iteration. Finally, as we discussed before, Fig. 4.3 illustrates how DADA and DoG differ as r_ϵ grows exponentially. It is evident that DoG experiences a decline in performance, while our method remains largely unaffected by this change, highlighting DADA's robustness with respect to the initial guess.

Hölder-Smooth Sample Function. In this section, we focus on solving the following test problem:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n} [\langle a_i, x\rangle - b_i]_+^q , \tag{4.2}$$

where $a_i, b_i \in \mathbb{R}^d$, $q \in [1,2]$, and $[x]_+ = \max(0,x)$.

Note that f is a Hölder-smooth function with the parameter $\nu = q-1$. This allows us to evaluate the effectiveness of parameter-free algorithms for different values of ν in Hölder-smooth functions. By varying $q \in [1,2]$, we demonstrate the robustness of DADA in achieving convergence over this spectrum of convex functions.

The data for our problem is generated randomly, following the procedure in [17]. First, we sample x^* uniformly from the sphere of radius $0.95R$ centered at the origin. Next, we generate i.i.d. vectors a_i with components uniformly distributed in $[-1,1]$. To ensure that $\langle a_n, x^*\rangle < 0$, we invert the sign of a_n if necessary. We then sample positive reals s_i uniformly from $[0, -0.1c_{\min}]$, where $c_{\min} := \min_i \langle a_i, x^*\rangle < 0$, and set $b_i = \langle a_i, x^*\rangle + s_i$. By construction, x^* is a solution to the problem with $f^* = 0$. Moreover, the origin $x_0 = 0$ lies outside the polyhedron, since there exists a j (corresponding to $c_{\min}$) such that $b_j = c_{\min} + s_j \le 0.9c_{\min} < 0$.

In this section, we fix $n = 10^4$, $d = 10^3$ and $R = 10^6$. As shown in Fig. 4.4, as q increases and approaches 2, the performance of DoG declines and fails to converge as

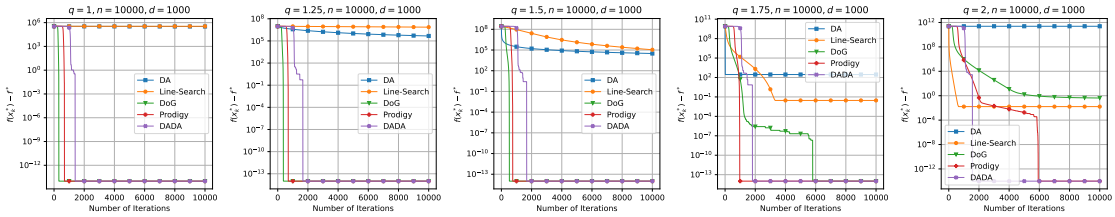


Figure 4.4: Comparison of different methods on the polyhedron feasibility problem.

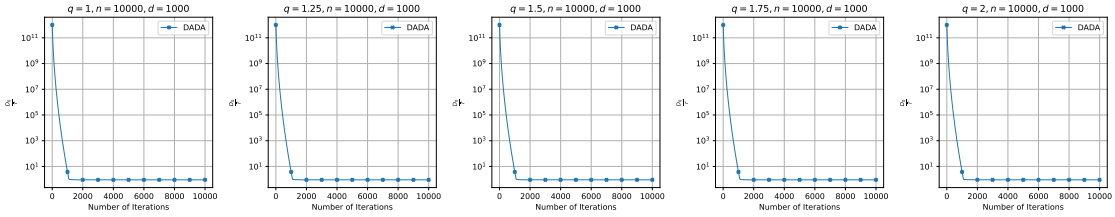


Figure 4.5: The ratio $\frac{D}{\bar{r}_t}$ for the polyhedron feasibility problem.

effectively as it does for smaller values of q.

p-th Power of Norm. As an example of (L_0, L_1)-smooth functions, we consider the following test problem:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{p}\|x\|^p, \tag{4.3}$$

for different choices of p and a starting point such that $\|x_0\| = R$, where $R = 10$. As shown in [19], f is (L_0, L_1)-smooth for any $L_1 > 0$, with $L_0 = \left(\frac{p-2}{L_1}\right)^{p-2}$.

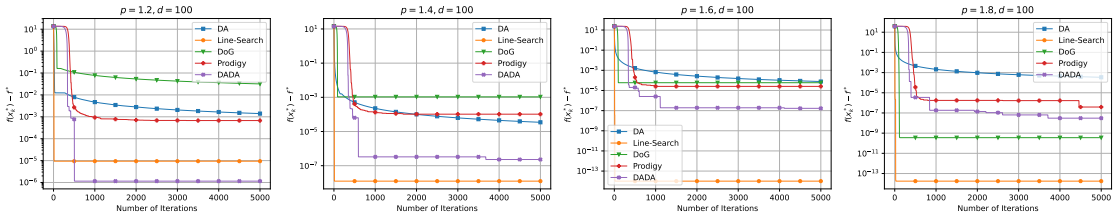
We consider two cases: $p < 2$ and $p > 2$, as shown in Fig.4.6, where you can see that for $p > 2$, the convergence speed of DoG and Prodigy decreases. However, this slowdown does not occur with DADA, which aligns with the theoretical results discussed in Section3 for (L_0, L_1)-smooth functions.

References

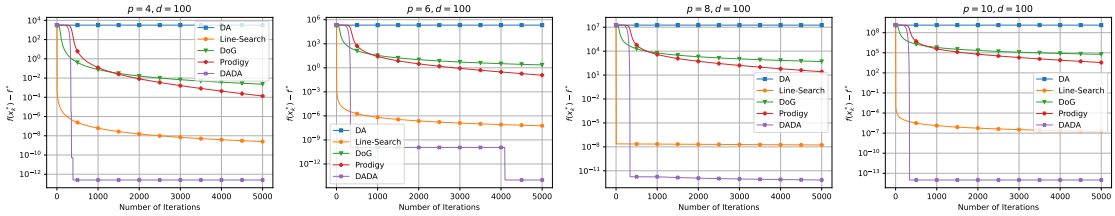
[1] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. URL: `https://doi.org/10.1137/16M1080173`.

[2] Y. Carmon and O. Hinder. Making sgd parameter-free. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178, pages 2360–2389, 2022. URL: `https://proceedings.mlr.press/v178/carmon22a.html`.

[3] A. Defazio and K. Mishchenko. Learning-rate-free learning by d-adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 7449–7479, 2023. URL: `https://proceedings.mlr.press/v202/defazio23a.html`.



(a) $p < 2$



(b) $p > 2$

Figure 4.6: Comparison of different methods' convergence on the function $f = \frac{1}{p}\|x\|^p$ with different values of p.

[4] N. Doikov. Minimizing quasi-self-concordant functions by gradient regularization of newton method, 2023. eprint: 2308.14742 (math.OC). URL: https://arxiv.org/abs/2308.14742.

[5] O. Hinder, A. Sidford, and N. Sohoni. Near-optimal methods for minimizing star-convex functions and beyond. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125, pages 1894–1938, 2020. URL: https://proceedings.mlr.press/v125/hinder20a.html.

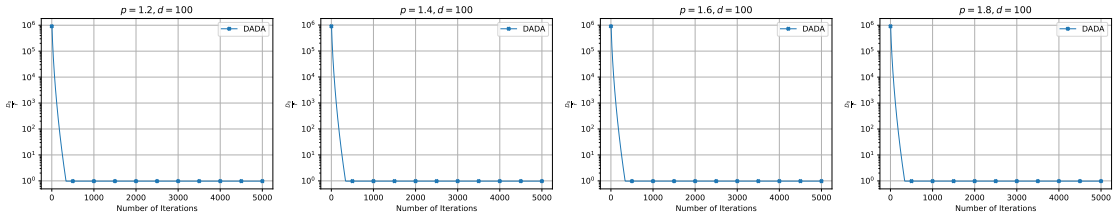
[6] M. Ivgi, O. Hinder, and Y. Carmon. DoG is SGD's best friend: a parameter-free dynamic step size schedule. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14465–14499, 2023. URL: https://proceedings.mlr.press/v202/ivgi23a.html.

[7] A. Khaled, K. Mishchenko, and C. Jin. DoWG unleashed: an efficient universal parameter-free gradient descent method. In *Advances in Neural Information Processing Systems*, volume 36, pages 6748–6769, 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/15ce36d35622f126f38e90167de1a350-Paper-Conference.pdf.

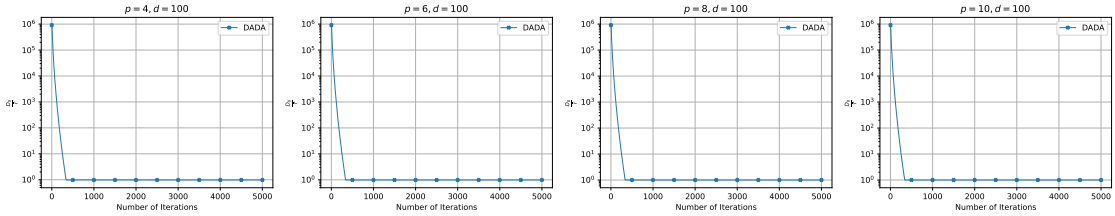
[8] T. Li and G. Lan. A simple uniformly optimal method without line search for convex optimization, 2024. URL: https://arxiv.org/abs/2310.10082.

[9] Z. Liu and Z. Zhou. Stochastic nonsmooth convex optimization with heavy-tailed noises. *ArXiv*, abs/2303.12277, 2023. URL: https://api.semanticscholar.org/CorpusID:257663403.

[10] K. Mishchenko and A. Defazio. Prodigy: an expeditiously adaptive parameter-free learner. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 35779–35804, 2024. URL: https://proceedings.mlr.press/v235/mishchenko24a.html.



(a) $p < 2$



(b) $p > 2$

Figure 4.7: The ratio $\frac{D}{\bar{r}_t}$ for the polyhedron feasibility problem. for the function $f = \frac{1}{p}\|x\|^p$ with different values of p.

[11] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 2005. URL: `https://api.semanticscholar.org/CorpusID:14935076`.

[12] Y. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152:381–404, 2015. URL: `https://api.semanticscholar.org/CorpusID:18062781`.

[13] Y. Nesterov. *Lectures on Convex Optimization*. Springer Publishing Company, Incorporated, 2nd edition, 2018, pages 221–259. ISBN: 3319915770. URL: `https://api.semanticscholar.org/CorpusID:14935076`.

[14] Y. Nesterov. Primal subgradient methods with predefined step sizes. *Journal of Optimization Theory and Applications*, 2024. DOI: `10.1007/s10957-024-02456-9`. URL: `https://arxiv.org/abs/2308.14742`.

[15] Y. Nesterov and B. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108:177–205, 2006. URL: `https://api.semanticscholar.org/CorpusID:7964929`.

[16] A. Reisizadeh, H. Li, S. Das, and A. Jadbabaie. Variance-reduced clipping for non-convex optimization. *ArXiv*, abs/2303.00883, 2023. URL: `https://api.semanticscholar.org/CorpusID:257280493`.

[17] A. Rodomanov, X. Jiang, and S. U. Stich. Universality of adagrad stepsizes for stochastic optimization: inexact oracle, acceleration and variance reduction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL: `https://openreview.net/forum?id=rniiAVjHi5`.

[18] A. Rodomanov and Y. Nesterov. Smoothness parameter of power of euclidean norm. *Journal of Optimization Theory and Applications*, 185:303–326, 2019. URL: `https://api.semanticscholar.org/CorpusID:198968030`.

[19] D. Vankov, A. Rodomanov, A. Nedich, L. Sankar, and S. U. Stich. Optimizing (L_0, L_1)-smooth functions by gradient methods, 2024. arXiv: 2410.10800 [math.OC]. URL: https://arxiv.org/abs/2410.10800.

[20] J. Zhang, T. He, S. Sra, and A. Jadbabaie. Why gradient clipping accelerates training: a theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL: https://openreview.net/forum?id=BJgnXpVYwS.

A Related Work

Throughout this paper, we use the term "parameter-free algorithms" to describe optimization algorithms that do not have any tuning parameters.

One notable contribution to parameter-free optimization is by [2] which their method employs a bisection search strategy to approximate the optimal stepsize, eliminating the need for explicit tuning. However, this technique introduces a logarithmic factor, $\log(T)$, into the convergence rate, as a result of the iterative bisection process. In contrast [3, 10] use Dual Averaging to estimate the distance between the initial point and the optimal solution, $D_0 = \|x_0 - x^*\|$. This estimate is then incorporated into the step size. Although this approach is promising for unconstrained problems, it is not directly applicable to constrained optimization.

Another method introduced by [6], also estimates d_0 through a different sequence of estimates, denoted as $(\bar{r}_i)_{i=0}^{\infty}$. While this approach shares similarities with our method in its estimation process, it uses Gradient Descent scheme instead of Dual Averaging. A closely related work [7] builds on the [6] method by introducing a weighted gradient sum for estimating d_0 using the same sequence $(\bar{r}_i)_{i=0}^{\infty}$. This method focuses primarily on unconstrained optimization, though it also provides theoretical results for constrained cases, given the assumption of bounded domain. However, these methods are only applicable to certain classes of convex functions, such as smooth and nonsmooth functions, motivating the development of more general, parameter-free methods that do not rely on this knowledge and can be extended to a broader range of convex function classes.

B Auxiliary Results

The following Lemma has been proved in prior works such as [9, Lemma 30]. We include the proof here for the reader's convenience.

Lemma B.1. *Let $(d_i)_{i=0}^{\infty}$ be a positive nondecreasing sequence. Then for any $T \geq 1$,*

$$\min_{0 \leq t \leq T} \frac{d_t}{\sum_{i=0}^{t-1} d_i} \leq \frac{\left(\frac{d_T}{d_1}\right)^{\frac{1}{T}} \log \frac{e d_T}{d_0}}{T}.$$

Proof. Let $A_t \coloneqq \sum_{i=0}^{t-1} \frac{d_i}{d_t}$ for all $t \in \mathbb{N}^+$ where $A_0 = 0$. Then we know

$$d_t A_t - d_{t-1} A_{t-1} = d_{t-1},$$

which implies that

$$A_t - \frac{d_{t-1}}{d_t} A_{t-1} = \frac{d_{t-1}}{d_t}.$$

By summing up for all $1 \leq t \leq T$ we get

$$A_T + \sum_{t=0}^{T-1} \left(1 - \frac{d_t}{d_{t+1}}\right) A_t = \sum_{t=0}^{T-1} \frac{d_t}{d_{t+1}}.$$

Since $(d_i)_{i=0}^{\infty}$ is a non-decreasing sequence, we have

$$\left(\max_{0\le t\le T} A_t\right)\left(1+\sum_{t=0}^{T-1}\left\{1-\frac{d_t}{d_{t+1}}\right\}\right)\ge\sum_{t=0}^{T-1}\frac{d_t}{d_{t+1}}.$$

Using AM-GM inequality we have $\sum_{t=0}^{T-1}\frac{d_t}{d_{t+1}}\ge T\left(\frac{d_0}{d_T}\right)^{\frac{1}{T}}$. Therefore,

$$\left(\max_{0\le t\le T} A_t\right)\left(1+\sum_{t=0}^{T-1}\left\{1-\frac{d_t}{d_{t+1}}\right\}\right)\ge T\left(\frac{d_0}{d_T}\right)^{\frac{1}{T}}.$$

Note that $1-\frac{1}{x}\le\log x$, then

$$\sum_{t=0}^{T-1}\left\{1-\frac{d_t}{d_{t+1}}\right\}\le\sum_{i=0}^{T-1}\log\frac{d_{t+1}}{d_t}=\log\frac{d_T}{d_0}.$$

Hence, we obtain

$$\max_{0\le t\le T} A_t\ge\frac{T\left(\frac{d_0}{d_T}\right)^{\frac{1}{T}}}{1+\log\frac{d_T}{d_0}}.$$

Therefore,

$$\min_{0\le t\le T}\frac{d_t}{\sum_{i=0}^{t-1}d_i}\le\frac{\left(\frac{d_T}{d_0}\right)^{\frac{1}{T}}\left(1+\log\frac{d_T}{d_0}\right)}{T}=\frac{\left(\frac{d_T}{d_0}\right)^{\frac{1}{T}}\log\frac{ed_T}{d_0}}{T}.$$ □

This Lemma has been established in [13, Lemma 3.2.1] and the proof included here for the reader's convenience.

Lemma B.2. *For any $x\in\mathbb{R}^d$ we have $f(x)-f^*\le\omega(v(x))$.*

Proof. We have two cases:

1. If $\langle g(x),x-x^*\rangle<0$, then $v(x)<0$ and
$$f^*\ge f(x)+\langle g(x),x^*-x\rangle\ge f(x),$$
which implies that
$$f(x)-f^*\le 0=\omega(v(x)).$$
2. If $\langle g(x),x-x^*\rangle\ge 0$ then we consider point $\bar{y}$ as follows
$$\bar{y}=x^*+v(x)\frac{g(x)}{\|g(x)\|_*}.$$
Note that we have
$$\langle g(x),\bar{y}-x\rangle=\langle g(x),x^*-x\rangle+\frac{\langle g(x),x-x^*\rangle}{\|g(x)\|_*}\frac{\|g(x)\|_*^2}{\|g(x)\|_*}=0,$$
And also $\|\bar{y}-x^*\|=v(x)$. Therefore,
$$f(\bar{y})\ge f(x)+\langle g(x),\bar{y}-x\rangle=f(x).$$
Hence
$$f(x)-f^*\le f(\bar{y})-f^*\le\omega(\|\bar{y}-x^*\|)=\omega(v(x)).$$ □

C Analysis of Dual Averaging

Theorem C.1. *Using Section 2 for T steps we can derive the following bound:*

$$\forall_{k\leq T}: \sum_{i=0}^{k-1} a_i v_i \|g_i\|_* \leq \frac{\beta_k}{2} D_0^2 + \sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i}\|g_i\|_*^2 - \frac{\beta_k}{2}\|x_k - x^*\|^2,$$

where $D_0 = \|x_0 - x^*\|$ be the initial gap.

Proof. Indeed,

$$\begin{aligned}\psi_{k+1}(x) &= \psi_k(x) + a_k\langle g_k, x - x_k\rangle + \frac{\beta_{k+1}-\beta_k}{2}\|x - x_0\|^2 \\ &\geq \psi_k(x_k) + \frac{\beta_k}{2}\|x - x_k\|^2 + a_k\langle g_k, x - x_k\rangle + \frac{\beta_{k+1}-\beta_k}{2}\|x - x_0\|^2,\end{aligned}$$

where the final inequality is due to the fact that ψ_i is β_i-strongly convex and $\bar{x}_i$ is its minimizer. Now we put $x = x_{i+1}$ in the above inequality:

$$\begin{aligned}\psi_{k+1}(x_{k+1}) &\geq \psi_k(x_k) + \frac{\beta_k}{2}\|x_{k+1} - x_k\|^2 + a_k\langle g_k, x_{k+1} - x_k\rangle + \frac{\beta_{k+1}-\beta_k}{2}\|x_{k+1} - x_0\|^2 \\ &\geq \psi_k(x_k) - \frac{a_k^2}{2\beta_k}\|g_k\|_*^2 + \frac{\beta_{k+1}-\beta_k}{2}\|x_{k+1} - x_0\|^2,\end{aligned}$$

which the last inequality comes from Hölder inequality:

$$\frac{\beta_k}{2}\|x_{k+1} - x_k\|^2 + \frac{a_k^2}{2\beta_k}\|g_k\|_*^2 \geq a_k\|x_{k+1} - x_k\|\|g_k\|_* \geq a_k\langle g_k, x_k - x_{k+1}\rangle.$$

By doing the same for $i = 0, \ldots, k-1$ we derive the following inequality:

$$\begin{aligned}\psi_k(x_k) &\geq -\sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2 + \sum_{i=0}^{k-1}\frac{\beta_{i+1}-\beta_i}{2}\|x_{i+1} - x_0\|^2 \\ &\geq -\sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2.\end{aligned}$$

Finally, using the strong convexity of ψ_i, we can complete our proof:

$$\begin{aligned}\sum_{i=0}^{k-1} a_i\langle g_i, x^* - x_i\rangle + \frac{\beta_k}{2}D^2 = \psi_k(x^*) &\geq \psi_k(x_k) + \frac{\beta_k}{2}\|x_k - x^*\|^2 \\ &\geq -\sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2 + \frac{\beta_k}{2}\|x_k - x^*\|^2,\end{aligned}$$

which implies that,

$$\sum_{i=0}^{k-1} a_i\langle g_i, x_i - x^*\rangle \leq \frac{\beta_k}{2}D_0^2 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2 - \frac{\beta_k}{2}\|x_k - x^*\|^2.$$

Therefore,

$$\sum_{i=0}^{k-1} a_i v_i\|g_i\|_* \leq \frac{\beta_k}{2}D_0^2 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2 - \frac{\beta_k}{2}\|x_k - x^*\|^2. \qquad \square$$

D Proof of Theorem 2.1

Lemma D.1. *Consider Section 2 run for T steps using the coefficients defined in* (2.1)*. Then we have the following inequality,*

$$\forall_{k\leq T}: \quad r_k \leq 2\|x_0 - x^*\| + \frac{\bar{r}_{k-1}}{\sqrt{2}}.$$

Proof. Using Theorem C.1 we can get an upper bound on $\|x_k - x^*\|$ as follows:

$$\sum_{i=0}^{k-1} a_i v_i \|g_i\|_* + \frac{\beta_k}{2}\|x_k - x^*\|^2 \leq \frac{\beta_k}{2}\|x_0 - x^*\|^2 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2.$$

Since x^* is the minimizer of f one can conclude that that $v_i \geq 0$ and hence

$$\frac{\beta_k}{2}\|x_k - x^*\|^2 \leq \frac{\beta_k}{2}\|x_0 - x^*\|^2 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2,$$

therefore,

$$\|x_k - x^*\|^2 \leq \|x_0 - x^*\|^2 + \frac{1}{\beta_k}\sum_{i=0}^{k-1}\frac{a_i^2}{\beta_i}\|g_i\|_*^2.$$

Substituting $\beta_i = 2\sqrt{i+1}, a_i = \frac{\bar{r}_i}{\|g_i\|_*}$ into the above inequality, we get

$$\|x_k - x^*\|^2 \leq \|x_0 - x^*\|^2 + \frac{1}{4\sqrt{k+1}}\sum_{i=0}^{k-1}\frac{\bar{r}_i^2}{\sqrt{i+1}},$$

and by using following inequality,

$$\sum_{i=0}^{k-1}\frac{1}{\sqrt{i+1}} \leq \int_0^k \frac{1}{\sqrt{x}}dx = 2\sqrt{k},$$

we get,

$$\|x_k - x^*\|^2 \leq \|x_0 - x^*\|^2 + \frac{\bar{r}_{k-1}^2}{2}\sqrt{\frac{k}{k+1}} \leq \|x_0 - x^*\|^2 + \frac{\bar{r}_{k-1}^2}{2}.$$

By getting square root from both sides of the above inequality we get,

$$\begin{aligned}\|x_k - x^*\| &\leq \sqrt{\|x_0 - x^*\|^2 + \frac{\bar{r}_{k-1}^2}{2}} \\ &\leq \|x_0 - x^*\| + \frac{\bar{r}_{k-1}}{\sqrt{2}}.\end{aligned}$$

Therefore, by adding $\|x_0 - x^*\|$ to both sides and using triangle inequality we can finish our proof,

$$r_k = \|x_k - x_0\| \leq \|x_k - x^*\| + \|x_0 - x^*\| \leq 2\|x_0 - x^*\| + \frac{\bar{r}_{k-1}}{\sqrt{2}}. \qquad \square$$

Lemma D.2. *Consider Section 2 run for T steps using the coefficients defined in* (2.1)*. Then we have,*

$$\forall_{k\leq T}: \quad r_k \leq 8\|x_0 - x^*\|.$$

Proof. Using Lemma D.1 we have,

$$r_k \leq 2\|x_0 - x^*\| + \frac{\bar{r}_{k-1}}{\sqrt{2}}.$$

As we need an upper bound for $\bar{r}_k$ and not r_k, we use induction here to prove that $\bar{r}_k \leq 8\|x_0 - x^*\|$. Suppose that we know $\bar{r}_{k-1} \leq 8\|x_0 - x^*\|$ and we prove this inequality holds for $\bar{r}_k$:

$$r_k \leq 2\|x_0 - x^*\| + \frac{\bar{r}_{k-1}}{\sqrt{2}} \leq 2\|x_0 - x^*\| + \frac{8}{\sqrt{2}}\|x_0 - x^*\| \leq 8\|x_0 - x^*\|.$$

Therefore, by using the definition of $\bar{r}_k$ we can proof our claim,

$$\bar{r}_k = \max\left\{\max_{i\leq k} r_i, \bar{r}\right\} = \max\left\{r_k, \bar{r}_{k-1}\right\} \leq \max\left\{8\|x_0 - x^*\|, \bar{r}_{k-1}\right\} = 8\|x_0 - x^*\|,$$

where the last equality comes from $\bar{r}_{k-1} \leq 8\|x_0 - x^*\|$. □

of Theorem 2.1. From Theorem C.1, we know:

$$\forall_{k\leq T}: \quad \sum_{i=0}^{k-1} a_i v_i \|g_i\|_* + \frac{\beta_k}{2}\|x_k - x^*\|^2 + \leq \frac{\beta_k}{2}\|x_0 - x^*\|^2 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2.$$

Note that:

$$\begin{aligned}\|x_0 - x^*\|^2 - \|x_k - x^*\|^2 &= \left(\|x_0 - x^*\| - \|x_k - x^*\|\right)\left(\|x_0 - x^*\| + \|x_k - x^*\|\right)\\ &\leq 2\|x_k - x_0\|\|x_0 - x^*\| = 2\|x_k - x_0\|D_0.\end{aligned}$$

Hence we have following bound on error term:

$$\sum_{i=0}^{k-1} a_i v_i \|g_i\|_* \leq \beta_k \|x_k - x_0\| D_0 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2.$$

Now let us define t^* as follows

$$t^* = \operatorname*{argmin}_{1\leq s\leq T} \frac{\bar{r}_s}{\sum_{i=0}^{s-1}\bar{r}_i}.$$

Using $a_i = \frac{\bar{r}_i}{\|g_i\|_*}$, $\beta_i = 2\sqrt{i+1}$ and $n = t^*$ we get the following inequality:

$$\begin{aligned}\sum_{i=0}^{t^*-1}\bar{r}_i v_i &\leq 2\|x_{t^*} - x_0\|R\sqrt{t^*} + (1/4)\sum_{i=0}^{t^*-1}\frac{\bar{r}_i^2}{\sqrt{i+1}}\\ &\leq 2\bar{r}_{t^*}R\sqrt{t^*} + (1/4)\sum_{i=0}^{t^*-1}\frac{\bar{r}_i^2}{\sqrt{i+1}}\\ &\leq \bar{r}_{t^*}\left(2R\sqrt{t^*} + \frac{\bar{r}_{t^*}}{4}\sum_{i=0}^{t^*-1}\frac{1}{\sqrt{i+1}}\right)\\ &\leq \bar{r}_{t^*}\left(2R\sqrt{t^*} + \frac{\bar{r}_{t^*}}{2}\sqrt{t^*}\right).\end{aligned}$$

By using Lemma B.1 we obtain

$$v_T^* \leq \frac{\sum_{i=0}^{t^*-1} \bar{r}_i v_i}{\sum_{i=0}^{t^*-1} \bar{r}_i} \leq \frac{\bar{r}_{t^*}}{\sum_{i=0}^{t^*-1} \bar{r}_i} \left(2R\sqrt{t^*} + \frac{\bar{r}_{t^*}}{2}\sqrt{t^*}\right) \leq \frac{2\left(R + \frac{\bar{r}_{t^*}}{4}\right)\sqrt{t^*}}{T} \left(\frac{\bar{r}_T}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{e\bar{r}_T}{\bar{r}},$$

therefore, using Lemma D.2 we can finish our proof

$$v_T^* \leq \frac{2\left(R + \frac{\bar{r}_{t^*}}{4}\right)\sqrt{t^*}}{T} \left(\frac{\bar{r}_T}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{e\bar{r}_T}{\bar{r}} \leq \frac{6R}{\sqrt{T}} \left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}}.$$

Using Lemma B.2, we get

$$\omega(v_T^*) = \min_{0 \leq i \leq T} \omega(v_i) \geq \min_{0 \leq i \leq T} (f(x_i) - f^*) = f(\bar{x}_T) - f^*. \qquad \square$$

E Growth-Function Bounds for Various Problem Classes

E.1 Nonsmooth Lipschitz Functions

Lemma E.1. *Assume that f is an L_0-Lipschitz continuous function. Then, for any $t \in \mathbb{R}$,*

$$\omega(t) \leq L_0 t.$$

Proof. Indeed, for any $x \in \mathbb{R}^d$ such that $\|x - x^*\| \leq t$, we have

$$f(x) - f^* \leq L_0 \|x - x^*\| \leq L_0 t. \qquad \square$$

In the case that we have lipschitz functions we will get an upper bound on error term immediately using Theorem 2.1 and Lemma E.1,

$$f(x_T^*) - f^* \leq \omega(v_T^*) \leq L_0(v_T^*) \leq \frac{6RL_0}{\sqrt{T}} \left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}}.$$

E.2 Lipschitz-Smooth Functions

Lemma E.2. *Assume that f has Lipschitz Gradients with constant L_1, then*

$$\omega(t) \leq \frac{L_1}{2} t^2 + \|\nabla f(x^*)\|_* t.$$

for all $x \in \mathbb{R}^n$.

Proof. Since f has Lipschitz Gradients, it is a L-Smooth function, which implies that

$$f(x) - f(x^*) \leq \langle \nabla f(x^*), x - x^* \rangle + \frac{L_1}{2} \|x - x^*\|^2,$$

therefore, we can get the following inequality,

$$\begin{aligned} f(x) - f(x^*) &\leq \langle \nabla f(x^*), x - x^* \rangle + \frac{L_1}{2} \|x - x^*\|^2 \\ &\leq \|\nabla f(x^*)\|_* \|x - x^*\| + \frac{L_1}{2} \|x - x^*\|^2. \end{aligned}$$

Hence

$$\omega(t) \leq \frac{L_1}{2} t^2 + \|\nabla f(x^*)\|_* t. \qquad \square$$

In the case that we have $f \in C^{1,1}_{L_1}(\mathbb{R}^d)$, using Lemma E.2 we have,

$$f(x_T^*) - f^* \leq \omega(v_T^*) \leq \frac{L_1}{2} H^2 + \|\nabla f(x^*)\|_* H,$$

where $H = \frac{6R}{\sqrt{T}} \left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}}$.

E.3 Hölder-Smooth Functions

Lemma E.3. *Suppose that f is Hölder Smooth, then*

$$\omega(t) \leq \|\nabla f(x^*)\|_* t + \frac{H_\nu}{\nu + 1} t^{\nu+1}.$$

Proof. First, using the definition of Hölder smooth functions we have

$$f(x) - f(x^*) \leq \langle \nabla f(x^*), x - x^* \rangle + \frac{H_\nu}{\nu + 1} \|x - x^*\|^{\nu+1},$$

therefore, using Cauchy-Schwarz inequality we obtain

$$f(x) - f(x^*) \leq \|\nabla f(x^*)\|_* \|x - x^*\| + \frac{H_\nu}{\nu + 1} \|x - x^*\|^{\nu+1}.$$

Hence

$$\begin{aligned}
[b]\omega(t) &= \max_x \{f(x) - f(x^*) : \|x - x^*\| \leq t\} \\
&\leq \max_x \left\{ \|\nabla f(x^*)\|_* \|x - x^*\| + \frac{H_\nu}{\nu + 1} \|x - x^*\|^{\nu+1} : \|x - x^*\| \leq t \right\} \\
&\leq \|\nabla f(x^*)\|_* t + \frac{H_\nu}{\nu + 1} t^{\nu+1}.
\end{aligned}$$

□

Thus in the case that f is a Hölder Smooth function, using Lemma E.3 we have

$$f(x_T^*) - f^* \leq \omega(v_T^*) \leq \frac{H_\nu}{\nu + 1} S^{\nu+1} + \|\nabla f(x^*)\|_* S,$$

where $S = \frac{6R}{\sqrt{T}} \left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}}$.

E.4 Functions with Lipschitz High-Order Derivative

As already mentioned in [14] we have the following bound on growth function ω in the case that the objective function's pth derivative is L_p-Lipschitz:

$$\omega(t) \leq \sum_{i=1}^{p} \frac{1}{i!} \|\nabla^i f(x^*)\|_* t^i + \frac{L_p}{(p+1)!} t^{p+1},$$

which directly implies that

$$f(x_T^*) - f^* \leq \omega(v_T^*) \leq \sum_{i=1}^{p} \frac{1}{i!} \|\nabla^i f(x^*)\|_* H^i + \frac{L_p}{(p+1)!} H^{p+1},$$

where $H = \frac{6R}{\sqrt{T}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}}$. To obtain the complexity bound and achieve ϵ-accuracy, we make each term less than $\frac{\epsilon}{p+1}$, resulting in:

$$\frac{1}{i!}\|\nabla^i f(x^*)\|_* H^i \leq \frac{\epsilon}{p+1}.$$

Thus, we get:

$$\left(\frac{(p+1)}{i!}\frac{(6R)^i\|\nabla^i f(x^*)\|_*(p+1)}{\epsilon}\left(\frac{4R}{\bar{r}}\right)^{\frac{i}{T}}\log^i\frac{4eR}{\bar{r}}\right)^{\frac{2}{i}} \leq T.$$

For the last term, we obtain a similar bound, replacing $\|\nabla^i f(x^*)\|_*$ with L_p. Hence, our method achieves the following complexity:

$$O\left(\max\left\{\max_{2\leq i\leq p}\left[\frac{p}{i!}\frac{\|\nabla^i f(x^*)\|_* R^i}{\epsilon}\right]^{\frac{2}{i}}\log_+^2\frac{R}{\bar{r}}, \left[\frac{p}{(p+1)!}\frac{L_p R^{p+1}}{\epsilon}\right]^{\frac{2}{p+1}}\log_+^2\frac{R}{\bar{r}}\right\}\right).$$

E.5 Quasi-Self-Concordant Functions

We use the following lemma from [4] to prove the convergence of our method on Quasi-Self-Concordant functions.

Lemma E.4. [4, Lemma 2.7] *Let f be a Quasi-Self-Concordant function with the parameter M, then for any x, y the following inequality holds*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \langle \nabla^2 f(x)(y-x), y-x\rangle \varphi(M\|y-x\|),$$

where $\varphi(t) \coloneqq \frac{e^t - t - 1}{t^2}$.

Lemma E.5. *Let f be a Quasi-Self-Concordant function with the parameter M, then for any x, y such that $\|x - y\| \leq \frac{1}{M}$ the following inequality holds*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{3}{4}\langle \nabla^2 f(x)(y-x), y-x\rangle.$$

Proof. Using the fact that $e^t - t - 1 \leq \frac{3}{4}t^2$ for all $t \in [0, 1]$ we obtain

$$\varphi(M\|y-x\|) \leq \frac{3}{4},$$

therefore, using Cauchy-Schwarz and properties of spectral norm inequality we will get

$$\begin{aligned} f(y) &\leq f(x) + \langle \nabla f(x), y - x\rangle + \langle \nabla^2 f(x)(y-x), y-x\rangle \varphi(M\|y-x\|) \\ &\leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{3}{4}\langle \nabla^2 f(x)(y-x), y-x\rangle \\ &\leq f(x) + \|\nabla f(x)\|_*\|y-x\| + \frac{3}{4}\|\nabla^2 f(x)\|_*\|y-x\|^2, \end{aligned}$$

which implies that

$$f(y) - f(x) \leq \|\nabla f(x)\|_*\|y-x\| + \frac{3}{4}\|\nabla^2 f(x)\|_*\|y-x\|^2. \qquad \square$$

If we choose $T_s > \max\left\{\log \frac{8R}{\bar{r}}, 36e^2(MR)^2 \log^2 \frac{8eR}{\bar{r}}\right\}$ we can conclude that

$$\frac{6R}{\sqrt{T}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}} \leq \frac{1}{M}.$$

Define variable $H_s := \frac{6R}{\sqrt{T}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}}$. Now using Theorem 2.1 we have

$$f(x^*_{T_s}) - f^* \leq \omega(v^*_{T_s}) \leq \omega\left(H_s\right).$$

Since $H_s \leq \frac{1}{M}$ we can use Lemma E.7 in the following way

$$\begin{aligned}
\omega\left(H_s\right) &= \max_x \left\{f(x) - f(x^*) : \|x - x^*\| \leq H_s\right\} \\
&\leq \max_x \left\{\|\nabla f(x^*)\|_*\|x - x^*\| + \frac{3}{4}\|\nabla^2 f(x^*)\|_*\|x - x^*\|^2 : \|x - x^*\| \leq H_s\right\} \\
&\leq \|\nabla f(x^*)\|_* H_s + \frac{3}{4}\|\nabla^2 f(x^*)\|_* H_s^2.
\end{aligned}$$

Hence

$$f(x^*_{T_s}) - f^* \leq \|\nabla f(x^*)\|_* H_s + \frac{3}{4}\|\nabla^2 f(x^*)\|_* H_s^2,$$

which implies our method's convergence for $T_s > \max\left\{\log \frac{8R}{\bar{r}}, 36e^2(MR)^2 \log^2 \frac{8eR}{\bar{r}}\right\}$.

E.6 (L_0, L_1)-Smooth Functions

Lemma E.6. *Let f be a (L_0, L_1)-smooth and x, y be arbitrary points, then the following inequality holds*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L_0 + L_1\|\nabla f(x)\|_*}{L_1^2}\left(e^{L_1\|y-x\|} - L_1\|y - x\| - 1\right).$$

Proof. You can find the proof in [19, Lemma 2.2]. □

Lemma E.7. *Let f be a (L_0, L_1)-smooth and x, y be arbitrary points such that $\|y - x\| \leq \frac{1}{L_1}$, then the following inequality holds*

$$f(y) - f(x) \leq \|\nabla f(x)\|_*\|y - x\| + \frac{3(L_0 + L_1\|\nabla f(x)\|_*)}{4}\|y - x\|^2.$$

Proof. Using Lemma E.6 and the fact that $e^t - t - 1 \leq \frac{3}{4}t^2$ for all $t \in [0, 1]$ we obtain

$$\begin{aligned}
f(y) &\leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{3(L_0 + L_1\|\nabla f(x)\|_*)}{4}\|y - x\|^2 \\
&\leq f(x) + \|\nabla f(x)\|_*\|y - x\| + \frac{3(L_0 + L_1\|\nabla f(x)\|_*)}{4}\|y - x\|^2,
\end{aligned}$$

which implies that

$$f(y) - f(x) \leq \|\nabla f(x)\|_*\|y - x\| + \frac{3(L_0 + L_1\|\nabla f(x)\|_*)}{4}\|y - x\|^2.$$ □

Using $T_s > \max\left\{\log \frac{8R}{\bar{r}}, 36e^2(L_1R)^2 \log^2 \frac{8eR}{\bar{r}}\right\}$ we can conclude that

$$\frac{6R}{\sqrt{T_s}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T_s}} \log \frac{8eR}{\bar{r}} \leq \frac{1}{L_1}.$$

Define variables $H_s := \frac{6R}{\sqrt{T_s}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T_s}} \log \frac{8eR}{\bar{r}}$ and $g_* = \|\nabla f(x^*)\|_*$. Now using Theorem 2.1 we have

$$f(x^*_{T_s}) - f^* \leq \omega(v^*_{T_s}) \leq \omega(H_s).$$

Since $H_s \leq \frac{1}{L_1}$ we can use Lemma E.7 in the following way

$$\begin{aligned}
\omega(H_s) &= \max_x \{f(x) - f(x^*) : \|x - x^*\| \leq H_s\} \\
&\leq \max_x \left\{g_*\|x - x^*\| + \frac{3(L_0 + L_1 g_*)}{4}\|x - x^*\|^2 : \|x - x^*\| \leq H_s\right\} \\
&\leq g_* H_s + \frac{3(L_0 + L_1 g_*)}{4} H_s^2.
\end{aligned}$$

Hence

$$f(x^*_{T_s}) - f^* \leq g_* H_s + \frac{3(L_0 + L_1 g_*)}{4} H_s^2,$$

which implies our method's convergence for $T_s > \max\left\{\log \frac{8R}{\bar{r}}, 36e^2(L_1R)^2 \log^2 \frac{8eR}{\bar{r}}\right\}$.