Anytime Acceleration Stepsize Schedule for Gradient Descent

Zecheng Li[*], Yilin Li[†]

1 Introduction

Gradient Descent (GD) is a cornerstone algorithm in optimization and machine learning, widely used to minimize smooth convex functions. Its simplicity and scalability make it a standard choice for large-scale problems, from training linear models to deep neural networks. In the classical setting where the objective function $f : \mathbb{R}^d \to \mathbb{R}$ is convex and L-smooth, GD with a fixed stepsize $\alpha \in (0, 2/L)$ guarantees a convergence rate of $\mathcal{O}(1/T)$ in function value after T iterations.

Despite this guarantee, the convergence can be prohibitively slow in practice, particularly when high-precision solutions are needed. This limitation has motivated the development of a broad range of accelerated variants of GD, aiming to reduce the number of iterations required to reach a target accuracy.

Among the most well-known acceleration techniques is Nesterov's accelerated gradient method, which improves the convergence rate to $\mathcal{O}(1/T^2)$ for smooth convex functions. However, this method typically requires either knowledge of the total number of iterations in advance or additional assumptions such as strong convexity. Furthermore, in practical applications, choosing the right acceleration parameters can be non-trivial.

More recently, Altschuler and Parrilo (2023) proposed the *Silver stepsize schedule*, a recursively constructed sequence of stepsizes that accelerates gradient descent without requiring tuning. The Silver schedule achieves a convergence rate of $\mathcal{O}(T^{-\log_2 \rho})$, where $\rho = 1 + \sqrt{2}$, at special iteration counts $T = 2^k - 1$. Despite its elegant construction and improved rate, the Silver schedule has a critical limitation: its convergence guarantees only apply at specific stopping times, and the performance of intermediate iterates can be non-monotonic or even suboptimal.

These limitations motivate a fundamental question in first-order optimization:

*Can we design a stepsize schedule that provides acceleration for gradient descent and guarantees fast convergence at **any** stopping time, without prior knowledge of the total number of iterations?*

Such a schedule would be highly desirable in practice, especially in settings where the computational budget or early-stopping criteria are data-dependent and cannot be fixed in advance. However, constructing such a schedule poses theoretical challenges, as it requires maintaining convergence guarantees uniformly across all time horizons without adaptive tuning.

To address this challenge, the method proposed in *Anytime Acceleration of Gradient Descent* introduces a novel framework based on *primitive schedules.* A primitive schedule is a short sequence of stepsizes that satisfies a certain energy inequality, guaranteeing function descent and gradient control at its final point. Crucially, these primitive blocks can be combined using a specially designed *join operator*, which preserves the desirable convergence properties across concatenations.

By recursively composing primitive schedules and join steps, the authors construct an infinite deterministic stepsize sequence $\hat{s} = [\alpha_1, \alpha_2, \ldots]$, such that *every prefix of $\hat{s}$ remains a valid primitive schedule.* This leads to a convergence guarantee of the form:

$$f(x_T) - f^* = \mathcal{O}\left(\frac{1}{T^\theta}\right), \quad \text{where } \theta = \frac{2\log_2 \rho}{1 + \log_2 \rho} \approx 1.119.$$

Compared to the classical $\mathcal{O}(1/T)$ rate, this provides a strictly faster convergence at every iteration, without requiring knowledge of the stopping time.

[*] zelli@ucdavis.edu

[†] ilnli@ucdavis.edu

2 Related work

The classical gradient descent (GD) algorithm with constant stepsize guarantees an $\mathcal{O}(1/T)$ convergence rate for smooth convex objectives. While optimal in the worst-case setting without additional assumptions, this rate leaves open the possibility of improvement through stepsize design, particularly when the stopping time is unknown in advance. To address this, researchers have explored non-monotonic and recursive stepsize strategies that inject occasional large steps to accelerate convergence.

Among these, the *silver schedule* (Altschuler and Parrilo, 2023) achieves an improved convergence rate of $\mathcal{O}(T^{-\log_2 \varepsilon})$ when evaluated at specific exponentially spaced stopping times. However, it performs poorly at arbitrary iterations and offers no general anytime guarantees. The open question of whether such *anytime acceleration* is possible—i.e., achieving better-than-$\mathcal{O}(1/T)$ rates uniformly across all T—was formalized by Kornowski and Shamir (2024). While subsequent works (Grimmer et al., 2024; Zhang and Jiang, 2024) made progress using schedule composition with known T, they do not solve the anytime case.

The recent breakthrough by Zhang et al. (2024) resolves this open problem by introducing a recursively constructed stepsize policy that guarantees $\mathcal{O}(T^{-1.119})$ convergence for all $T \in \mathbb{N}$, without knowledge of the stopping time. This result builds upon and extends the silver schedule via careful control of gradient norms at intermediate points. In this project, we empirically validate the convergence behavior of this anytime schedule and compare it with classical constant stepsize and silver approaches on synthetic convex optimization problems.

3 Methodology

This section elaborates on the methodology Kornowski and Shamir (2024) which addresses the challenge of designing a gradient-based optimization method that achieves provable acceleration without knowing the total number of iterations in advance. The core contribution is a stepsize scheduling scheme that guarantees improved convergence rates of $o(1/T)$ for all T, effectively combining the benefits of acceleration with the flexibility of anytime algorithms.

3.1 Problem Setting

The authors focus on the standard convex optimization problem:

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

where the objective function $f : \mathbb{R}^d \to \mathbb{R}$ is assumed to be convex and L-smooth. Without loss of generality, the authors normalize $L = 1$, so that the gradient satisfies:

$$\|\nabla f(x) - \nabla f(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^d.$$

The algorithmic approach is standard gradient descent (GD), with the update rule:

$$x_{t+1} = x_t - \omega_t \nabla f(x_t),$$

where $\omega_t > 0$ is the step size at iteration t. Under classical analysis, if a constant step size $\omega_t = \omega \in (0, 2)$ is used, then for any stopping time T, we have:

$$f(x_T) - f^* = \mathcal{O}\left(\frac{1}{T}\right),$$

where $f^* = \min_x f(x)$. However, this $1/T$ rate is known to be tight in the worst case for smooth convex functions. The central question posed by Kornowski and Shamir (2024) is whether one can obtain strictly better rates—i.e., $f(x_T) - f^* = o(1/T)$—uniformly for all T, while preserving the anytime property (i.e., without prior knowledge of T).

3.2 Primitive Schedules

Definition 1 (Primitive Schedule)**.** *A stepsize schedule* $\omega_{1:k} = [\omega_1, \ldots, \omega_k]$ *is said to be* primitive *if it guarantees the following inequality:*

$$A_k(f_k - f^*) + C_k\|\nabla f(x_k)\|^2 + \frac{1}{2}\|x_k - x^*\|^2 \leq \frac{1}{2}\|x_1 - x^*\|^2,$$

where $f_k = f(x_k)$*,* $A_k = \sum_{i=1}^{k-1} \omega_i$*, and* $C_k = \frac{A_k(A_k+1)}{2}$*. Primitive schedules offer convergence guarantees* at their final step*, making them suitable building blocks for anytime acceleration.*

Intuitively, a primitive schedule can be viewed as a self-contained schedule that makes guaranteed progress over its span. By combining multiple such schedules via join steps, longer and more flexible sequences can be constructed while preserving convergence guarantees.

3.3 The Silver Stepsize Schedule

The *silver schedule*, originally proposed by Altschuler and Parrilo (2023), is a recursively defined stepsize sequence that satisfies the primitive condition and enjoys improved convergence rates when used in gradient descent.

Concatenation of Primitive Schedules. Given two primitive sequences s and r with aggregate stepsizes $x = \sum s_i$ and $y = \sum r_j$, the authors define a *join step*:

$$\varphi(x, y) = \frac{(x + y) + \sqrt{(x + y + 2)^2 + 4(x + 1)(y + 1)}}{2},$$

which, when inserted between s and r, guarantees that the concatenated sequence $\texttt{concat}(s, r) = [s, \varphi(x, y), r]$ remains primitive. This construction plays a central role in building larger schedules from smaller ones.

Recursive Construction of *Silver Schedule* Starting from the empty schedule $\bar{s}_0 = [\ ]$, the i-th order *Silver Schedule* is defined recursively as:

$$\bar{s}_{i+1} = \texttt{concat}(\bar{s}_i, \bar{s}_i),$$

producing schedules of length $2^{i+1} - 1$. Since each $\bar{s}_i$ is primitive, and concatenation preserves primitiveness, the limiting schedule s_∞ is also primitive.

Key Properties. Let $\varepsilon = 1 + \sqrt{2}$ denote the *silver ratio*. Then:

$$\sum_{j=1}^{2^i - 1} \omega_j = \varepsilon^i - 1,$$

and for $T = 2^i - 1$, running GD with stepsizes $\bar{s}_i$ ensures:

$$f(x_T) - f^* = \mathcal{O}\left(\frac{1}{T^{\log_2 \varepsilon}}\right),$$

where $\log_2 \varepsilon \approx 1.2716$. This demonstrates superlinear convergence compared to the $1/T$ baseline.

Limitations. Despite its improved rate at specific T, the silver schedule does not provide guarantees for arbitrary t. In particular, intermediate iterates can deviate significantly from optimality due to the presence of large join steps. This motivates the design of the anytime scheme.

3.4 The Anytime Stepsize Schedule

To address the limitations of the silver schedule, the authors design a final anytime schedule $\hat{s}$ that combines multiple silver blocks of increasing order. For each level j, let s_j be the silver schedule of order j (length $2^j - 1$). Then define:

$$\hat{s}_0 = [\,], \quad \hat{s}_j = \texttt{concat}(\hat{s}_{j-1}, \underbrace{\bar{s}_j, \ldots, \bar{s}_j}_{k_j \text{ times}}),$$

where the number of repetitions k_j grows exponentially:

$$k_j = \left\lfloor 2 \cdot 2^{cj} \right\rfloor,$$

for some $c > 0$. The final schedule is the limit:

$$\hat{s} = \lim_{j \to \infty} \hat{s}_j.$$

This schedule guarantees acceleration at infinitely many T, and thanks to the exponential growth in k_j, ensures that arbitrary T is likely to fall near a high-quality point. The authors set $c = \log_2 \varepsilon$ to match the growth rate of the silver schedule.

4 Theoretical Results

4.1 Proof of Main Theorem

In the next section, we will present the proof for the main theorem; Our goal is to prove that, with the stepsize schedule we constructed, for any 1-smooth convex function, we can achieve acceleration over constant stepsize schedule convergence rate $\mathcal{O}(\frac{1}{T})$ at any stopping time ℓ without knowing ℓ beforehand.

Before moving to the proof, first, we introduce some notation we will use in the proof:

- t_i is the index of the i-th join step and t_{i+1} the next one; it is also the length of the i-th subsequence of constructed schedule.
- α_{t_i+1}: the join step connecting two Silver schedules.
- x, y: sum of the stepsize of the previous schedule and stepsize of the next schedule, respectively.
- $\rho = \sqrt{2} + 1$: the silver ratio.
- A_t: cumulative stepsize sum up to time $t - 1$.
- $C_t = \frac{A_t(A_t+1)}{2}$
- o_t is the integer satisfying $\sum_{j=1}^{o_t-1} k_j 2^j < t \leq \sum_{j=1}^{o_t} k_j 2^j$; in other words, to construct a subsequence containing t elements, we need at least the o_t-th subsequence of silver stepsize schedule; using lemma 6, we also conclude that $t_{i+1} = t_i + 2^{o_{t_i}+1} \leq 3t_i$;

In addition to that, we also introduce some lemma that will play important role in the proof of theorem 1, and the detailed proof is provided in the appendix:

Lemma 1. *suppose that each $\boldsymbol{s}_i$ is primitive, then each $\hat{\boldsymbol{s}}_i, i \geq 1$ generated by concatenation $\varphi(x, y)$ is primitive, and the infinite sequence $\hat{\boldsymbol{s}}$ is well-defined and primitive*

Lemma 2. *For the new defined stepsize schedule $\hat{\boldsymbol{s}}$ and for any $t \geq 1$, it holds that*

$$A_{t+1}(\hat{\boldsymbol{s}}) \geq \frac{1}{36} t^{\frac{c+\log_2 \rho}{c+1}},$$

*where $A_t(\hat{\boldsymbol{s}})$ is defined as the sum of first **t-1** term in $\hat{\boldsymbol{s}}$. Moreover, letting o_t be defined as mentioned above:*

$$\sum_{j=1}^{o_t - 1} k_j 2^j < t \leq \sum_{j=1}^{o_t} k_j 2^j,$$

one has

$$2^{o_t} \leq 2t^{\frac{1}{c+1}}.$$

Lemma 3. *Consider $i \geq 1$ and $\alpha \geq 0$, and let $k = 2^i$. Denote by $\bar{\boldsymbol{s}}_i = [\alpha_1, \dots, \alpha_{k-1}]^\top$ the i-th order silver stepsize schedule. Fix $\boldsymbol{x}_0$, set $\alpha_0 = \alpha$ and let $\boldsymbol{x}_1 = \boldsymbol{x}_0 - \alpha_0 \boldsymbol{g}_0$. If $\alpha \geq (\sqrt{2}-1)A_k + \sqrt{2}$, then one has*

$$f_\ell - f_0 \leq 432\alpha^2 \left\| \boldsymbol{g}_0 \right\|^2$$

for any ℓ obeying $1 \leq \ell \leq k-1$.

The general idea to prove theorem 1 is to decompose $f_\ell - f^*$ as:

$$f_\ell - f^* = (f_\ell - f_{t_i+1}) + (f_{t_i+1} - f^*)$$

Note that, we actually decompose the stepsize into 2 part: (1) a complete schedule of primitive stepsize schedule, $\boldsymbol{s}_1$, of length t_i and it is also the largest complete primitive stepsize schedule up to the ℓ-th term; (2) a silver stepsize schedule $\boldsymbol{s}_2$ of length $t_{i+1} - (t_i + 1)$ that starts from initialization $\boldsymbol{x}_{t_i+1}$ but stops at $\ell - t_i$;

We first observe that, by the way we construct this new stepsize schedule, it is still obtained by concatenating primitive stepsize schedules, then by **lemma 2**, for any $i \geq 1$, the subsequence of our new schedule satisfies the property of primitive stepsize schedule:

$$A_{t_{i+1}}(f_{t_{i+1}} - f^*) + C_{t_{i+1}} \left\| \boldsymbol{g}_{t_{i+1}} \right\|^2 + \frac{1}{2} \left\| \boldsymbol{x}_{t_{i+1}} - \boldsymbol{x}^* \right\|^2 \leq \frac{1}{2} \left\| \boldsymbol{x}_1 - \boldsymbol{x}^* \right\|^2,$$

This inequality will lead to two important inequalities:

$$f_{t_{i+1}} - f^* \leq \frac{\left\| \boldsymbol{x}_{t_1} - \boldsymbol{x}^* \right\|^2}{2A_{t_{i+1}}} \leq \frac{\left\| \boldsymbol{x}_{t_1} - \boldsymbol{x}^* \right\|^2}{A_{t_{i+1}}}$$

$$\left\| \boldsymbol{g}_{t_{i+1}} \right\|^2 \leq \frac{\left\| \boldsymbol{x}_{t_{i+1}} - \boldsymbol{x}^* \right\|^2}{2C_{t_{i+1}}} \leq \frac{\left\| \boldsymbol{x}_{t_{i+1}} - \boldsymbol{x}^* \right\|^2}{A_{t_{i+1}}^2}$$

Which are obtained by simply comparing each term in the left hand side of previous inequality with the right hand side.

Next, we take a slight detour to analyze the concatenating term α_{t_i+1}, which connect $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. For simplicity, we denote x as the sum of stepsize of $\boldsymbol{s}_1$ and y as the sum of stepsize of $\boldsymbol{s}_2$, we immediately observe that $x \geq y$, then we have:

$$x = \sum_{j=1}^{t_i} \alpha_j = A_{t_i+1} \geq \frac{1}{36} t_i^{\frac{c+\log_2 \rho}{c+1}}$$

$$y = \sum_{j=t_{i+2}}^{t_{i+1}} \alpha_j = \sum_{j=1}^{2^{o_{t_i}+1}-1} \alpha_j(\boldsymbol{s}_2) = \rho^{o_{t_i}+1} - 1 \leq 2t_i^{\frac{\log_2 \rho}{c+1}}$$

Both inequalities are direct results from **lemma 2**; in the second inequality, we also used the fact that $\boldsymbol{s}_2$ is a silver stepsize schedule and applied the **Key property** mentioned in section **3**.**4**. It follows then:

$$
\begin{aligned}
\alpha_{t_{i+1}} = \varphi(x,y) &= \frac{-(x+y)+\sqrt{(x+y+2)^2+4(x+1)(y+1)}}{2} \\
&= \frac{4(xy+2x+2y+2)}{2(x+y+\sqrt{(x+y+2)^2+4(x+1)(y+1)})} \\
&\leq y+2 \\
&= \rho^{o_{t_{i+1}}}+1
\end{aligned}
$$

Further, by taking derivative of $\varphi(x,y)$ with respect to x, we have $\frac{\partial\varphi(x,y)}{\partial x} \geq 0$, then:

$$\alpha_{t_i+1} = \varphi(x,y) \geq \varphi(y,y) = (\sqrt{2}-1)y+\sqrt{2}$$

With all the results above, we now have an upper bound and lower bound for α_{t_i+1}, and also it is worth mentioning that this lower bound is the condition $\boldsymbol{s}_2$ needs to satisfy to apply **lemma 3** and now we can formally derive the result in Theorem 1.

The first term of our decomposition,$f_{t_i+1}-f^*$, can simply be upper bounded by $\frac{\left\|\boldsymbol{x}_{t_{i+1}}-\boldsymbol{x}^*\right\|^2}{A_{t_{i+1}}}$ as shown before; since the second term, $f_\ell - f_{t_i+1}$, can be treated as a silver stepsize schedule with an early stopping time, and the lower bound for α_{t_+1} we derived before shows that the condition of **lemma 3** is satistfied and we can apply **lemma 3** to derive an upper bound for $f_\ell - f_{t_i+1}$:

$$f_\ell - f_{t_i+1} \leq 432\alpha_{t_i+1}^2 \left\|\boldsymbol{g}_{t_{i+1}}\right\|^2 = \mathcal{O}(\frac{\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2}{\ell^{\frac{2c}{c+1}}})$$

Again, the detailed derivation is provided in appendix.

Combine previous results, we have:

$$
\begin{aligned}
f_\ell - f^* &= (f_\ell - f_{t_i+1}) + (f_{t_i+1} - f^*) \\
&\leq \mathcal{O}(\frac{\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2}{\ell^{\frac{2c}{c+1}}} + \frac{\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2}{A_{t_i+1}}) \\
&= \mathcal{O}(\frac{\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2}{\ell^{\frac{2\log_2\rho}{1+\log_2\rho}}})
\end{aligned}
$$

by selecting $c = \log_2\rho$ to have these 2 terms merge. The remaining part is to prove that this result also holds for $\ell = 1, 2$, since in **lemma 3**, we require stopping time ℓ satisfy $t_i < \ell \leq t_{i+1}$, which means $\ell \in \cup_{i\geq 1}(t_i, t_{i+1}] = \{3, 4, 5, ...\}$ and we can easily verify this case using property of f as 1-smooth convex function and derive the following result:

$$f_1 - f^* \leq \frac{\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2}{2}$$

$$f_2 - f^* \leq f_1 - f^* + \frac{\alpha_1^2+2\alpha_1}{2}\left\|\boldsymbol{g}_1\right\|^2 \leq (1+\alpha_1^2+2\alpha_1)(f_1-f^*) \leq \frac{9\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2}{2}$$

and completes the proof.

4.2 Extension to Strongly Convex Problem

In this part, we further assume that the function f is μ-strongly convex with convexity parameter $\mu \in (0,1]$ and denote $\kappa = \frac{1}{\mu}$ as the condition number and put forward the following result as acceleration for any time in strongly convex case:

Theorem 1. *There exists a stepsize schedule $\{\alpha_t\}_{i=1}^{\infty}$, generated without knowing the stopping time, such that the gradient descent iterates obey*

$$f(\boldsymbol{x}_T) - f^* \leq \mathcal{O}\left(\exp\left(-\frac{CT}{\kappa^{\varsigma}}\right)\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2\right),$$

where $\varsigma = 1/\vartheta = \dfrac{1+\log_2 \rho}{2\log_2 \rho} < 0.893$, and $C > 0$ is some numerical constant. Here, T denotes an arbitrary stopping time that is unknown a priori.

We first introduce how this new stepsize schedule is generated: In **theorem 1**, we proved that, the gradient descent process using stepsize schedule $\hat{\boldsymbol{s}}$ obeys:

$$f(\boldsymbol{x}_t) - f^* \leq \frac{C_0\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2}{t^{\vartheta}}$$

for some universal constant $C_0 > 0$, our construction then depends on this C_0 and μ; we select $\tau = \tau(\mu)$ to be the smallest integer satistifies $A_{\tau+2}(\hat{\boldsymbol{s}}) \geq \frac{4C_0}{\mu} = 4C_0\kappa$, namely, the sum of first $\tau+1$ stepsize is barely larger than $4C_0\kappa$ and this implies $A_{\tau+1}(\hat{\boldsymbol{s}}) < 4C_0\kappa$; taken together with **lemma 2**, we have:

$$4C_0\kappa \geq A_{\tau+1}(\hat{\boldsymbol{s}}) \geq \frac{1}{36}\tau^{\frac{c+\log_2\rho}{c+1}} = \frac{1}{36}\vartheta$$

Now, consider the first τ stepsize of stepsize schedule we construct in theorem 1 and denote it as $\tilde{\boldsymbol{s}} = [\alpha_1, \alpha_2, ..., \alpha_\tau]$. But this time, we only repeat it infinitely many times without using the concatenating function $\varphi(x,y)$ as before to connect each repeating subsequences $\tilde{\boldsymbol{s}}$: we generate the desired stepsize schedule $\hat{\boldsymbol{s}}^{\boldsymbol{*}}$ only by repeating the first τ stepsize of $\hat{\boldsymbol{s}}$. In other words, our new stepsize schedule $\hat{\boldsymbol{s}}^{\boldsymbol{*}}$ is defined as:

$$\hat{\boldsymbol{s}}^{\boldsymbol{*}} = [\tilde{\boldsymbol{s}},\ \tilde{\boldsymbol{s}},\ \tilde{\boldsymbol{s}},\ \tilde{\boldsymbol{s}}, ...]$$

The general idea to prove the theorem is somewhat similar with the way we prove theorem 1:

- First notice that, for any stopping time ℓ, it can be represent in this form: $i\tau + j$, where $i \geq 1$ and $1 \leq j \leq \tau$;

- When $\tau = 0$, namely, we are implementing $\hat{\boldsymbol{s}}$ starting from $\boldsymbol{x}_0$ and stop at j, $1 \leq j \leq \tau$; this means we can use theorem 1 to obtain the following inequalities: (Note that in the first inequality, We upper bound the polynomial term by an exponential function by choosing the constants appropriately, exploiting that exponential decay eventually dominates polynomial decay.)

$$\begin{aligned} f_j - f^* &\leq \frac{C_0\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2}{j^{\vartheta}} \leq 55C_0\exp\left(-\frac{j}{36C_0\kappa^{\varsigma}}\right)\cdot\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 \quad \text{for all } 1 \leq j \leq \tau; \\ f_{\tau+1} - f^* &\leq \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2}{144\kappa} = \frac{\mu\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2}{144} \end{aligned}$$

- We make use of the nice property of f: 1-smoothness and μ-strong convexity, together we the way we choose τ to obtain recursive relationship between $\|\boldsymbol{x}_{(i-1)\tau+1} - \boldsymbol{x}^*\|^2$ and $\|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\|^2$:

$$\begin{aligned} &\frac{\mu}{2}\|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\|^2 \leq f_{i\tau+1} - f^* \leq \frac{\mu}{4}\|\boldsymbol{x}_{(i-1)\tau+1} - \boldsymbol{x}^*\|^2 \\ \Rightarrow &\|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\|^2 \leq \frac{1}{2}\|\boldsymbol{x}_{(i-1)\tau+1} - \boldsymbol{x}^*\|^2 \end{aligned}$$

- For any $\tau \neq 0, 1 \leq j \leq \tau$, we can treat this process as optimization starting from $x_{(i-1)\tau}$ and uses $\hat{\boldsymbol{s}}$ that stops at the τ-th iteration, hence we get:

$$f_{i\tau+1} - f^* \leq \frac{C_0 \|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\|^2}{j^{\vartheta}} \leq C_0 \|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\|^2$$

Take together all results above, we obtain:

$$\|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\|^2 \leq \frac{1}{2}\|\boldsymbol{x}_{(i-1)\tau+1} - \boldsymbol{x}^*\|^2 \leq \frac{1}{2^i}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 \leq \exp(-\frac{i\tau+1}{576 C_0 \kappa^{\varsigma}})\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2$$

Here we also use the trick to upper bound the polynomial term by an exponential function with appropriate constants.

As a result, for stopping time $T = i\tau + j$, apply the similar trick to bound polynomial term with exponential term, we have:

$$\begin{aligned} f_{i\tau+j} - f^* \leq C_0 \left\|\boldsymbol{x}_{i\tau+1} - \boldsymbol{x}^*\right\|^2 &\leq C_0 \exp\left(-\frac{i\tau+1}{576 C_0 \kappa^{\varsigma}}\right)\left\|\boldsymbol{x}_1 - \boldsymbol{x}^*\right\|^2 \\ &\leq C_0 \exp\left(-\frac{i\tau+j}{1152 C_0 \kappa^{\varsigma}}\right)\left\|\boldsymbol{x}_1 - \boldsymbol{x}^*\right\|^2 \end{aligned}$$

Moreover, let $C = \frac{1}{1152 C_0}$ and substitute $i\tau + j$ with original stopping time T, we obtain exactly the result stated in the theorem at the very beginning of the section and complete our proof:

$$f_T - f^* \leq C_0 \exp\left(-\frac{CT}{\kappa^{\varsigma}}\right)\left\|\boldsymbol{x}_1 - \boldsymbol{x}^*\right\|^2$$

5 Experimental Details

We conduct a synthetic experiment to evaluate the empirical performance of different step size schedules in gradient descent. Specifically, we compare the following methods:

- **Constant Step Size:** Standard gradient descent with a fixed step size $\alpha = 0.02$.
- **Silver Schedule:** The step size sequence constructed recursively using the "silver rule" Altschuler and Parrilo (2023), where each schedule is formed by concatenating two smaller schedules with a join step computed via a special merge function $\varphi(x, y)$.
- **Anytime Schedule:** A schedule that repeatedly nests silver blocks of increasing depth j, with the number of repetitions for each depth j chosen as $k_j = \lceil 2 \cdot 2^{cj} \rceil$ where $c = \log_2(1 + \sqrt{2})$.

The objective function is a quadratic form:

$$f(x) = \frac{1}{2}x^\top Ax,$$

where $A \in \mathbb{R}^{d \times d}$ is a diagonal matrix with eigenvalues logarithmically spaced between 1 and 100, normalized to have maximum eigenvalue 1. We set $d = 20$. The gradient is given by $\nabla f(x) = Ax$.

The initial point is set to $x_0 = 4 \cdot \mathbf{1}_d$. For each step size schedule, gradient descent is run for $T = 25$ iterations. The step sizes are clipped when necessary to avoid numerical overflow. The logarithm of the objective value is plotted at each iteration.

As shown in Figure 1, both the Silver and Anytime Schedules significantly outperform the constant step size baseline, achieving faster objective value decay. Notably, the Anytime Schedule maintains this improvement while preserving a flexible, anytime execution model. More simulation studies are in the Appendix 7.

6 Appendix

6.1 Proof of Lemma 2

We first introduce the following lemma:

Lemma 4. *Given the silver stepsize schedule $\boldsymbol{s}$, $\forall i \geq 1$, the i-th subsequence $\bar{\boldsymbol{s}}_i$ statistifies:*

$$\mathbf{1}^\top \bar{\boldsymbol{s}}_i = \rho^i - 1$$

where $\rho_i = \sqrt{2} + 1$. In other words, the sum of the first i stepsize of silver stepsize schedule is always equal to $\rho^i - 1$.

This lemma can be easily proved by induction: it is easy to verify the case when $i = 0$; Assuming this formula holds for $i \leq k$, according to the way we construct $\bar{\boldsymbol{s}}$, it follows that:

$$\begin{aligned}
\mathbf{1}^\top \bar{\boldsymbol{s}}_{k+1} &= \mathbf{1}^\top [\bar{\boldsymbol{s}}_k, \varphi(\bar{\boldsymbol{s}}_k, \bar{\boldsymbol{s}}_k), \bar{\boldsymbol{s}}_k] \\
&= 2 \times (\mathbf{1}^\top \bar{\boldsymbol{s}}_k) + \varphi(\bar{\boldsymbol{s}}_k, \bar{\boldsymbol{s}}_k) \\
&= 2(\rho^k - 1) + (\sqrt{2} - 1)(\rho^k - 1) + \sqrt{2} \\
&= (\sqrt{2} + 1)(\rho^k - 1) + \sqrt{2} \\
&= \rho(\rho^k - 1) + \sqrt{2} \\
&= \rho^{k+1} - \rho + \sqrt{2} \\
&= \rho^{k+1} - 1
\end{aligned}$$

which finishes the induction.

Move back to the proof of **lemma 2**, we first consider the case $o_t = 1$, for which we have $t \leq 2k_1 = 2 \cdot \lceil 2^{c+1} \rceil = 4$. Also note that $\varphi(x, y) > 1$ for all $x, y \geq 0$, we can easily numerically verify that

$$A_{t+1}(\hat{\boldsymbol{s}}) \geq \frac{1}{36} t^{\frac{c+\log_2 \rho}{c+1}}.$$

Sum of Stepsizes	Lower Bound
1.4142	0.0278
3.4142	0.0604
4.8284	0.0950
8.2426	0.1311

Table 1: Case when $o_t = 1$

It is also easy to check that $2^{o_t} = 2 \leq 2t^{\frac{1}{c+1}}$ in this case.

Next, we consider the case when $o_t \geq 2$; we first define a new integer $m \in [1, k_{o_t}]$ satisfying $\sum_{j=1}^{o_t - 1} k_j 2^j + (m-1)2^{o_t} < t < \sum_{j=1}^{o_t - 1} k_j 2^j + m 2^{o_t}$, in other words, m is the smallest number of repetitious o_t-th subsequence of silver stepsize schedule to construct a subsequence of $\hat{\boldsymbol{s}}$ that contains at least t elements; By the way we defining m, we have:

$$
\begin{aligned}
t &\leq \sum_{j=1}^{o_t-1} k_j 2^j + m2^{o_t} \\
&\leq \sum_{j=1}^{o_t-1} 2 \times 2^{jc} \times 2^j + m2^{o_t} \\
&\leq \sum_{j=1}^{o_t-1} 2 \times 2^{j(c+1)} + m2^{o_t} \\
&\leq 4 \times 2^{(o_t-1)(c+1)} + m2^{o_t}
\end{aligned}
$$

Together with **lemma 4**, we also obtain:

$$A_{t+1}(\hat{\boldsymbol{s}}) \geq \sum_{j=1}^{o_t-1} (\rho^j - 1) \cdot k_j + (m-1)(\rho^{o_t} - 1) \geq \frac{1}{2} \cdot 2^{(c+\log_2 \rho)(o_t-1)} + \frac{m-1}{2} \rho^{o_t}$$

Now, we consider 2 cases:

- if $m2^{o_t} \leq 2^{(o_t-1)(c+1)}$, we then have:

$$t \leq 4 \times 2^{(o_t-1)(c+1)} + m2^{o_t} \leq 6 \times 2^{(o_t-1)(c+1)}$$

this means:

$$A_{t+1}(\hat{\boldsymbol{s}}) \geq \frac{1}{2} \cdot 2^{(c+\log_2 \rho)(o_t-1)} = \frac{1}{2} \cdot (2^{(c+1)(o_t-1)})^{\frac{c+\log_2 \rho}{c+1}} \geq \frac{1}{2} \times \frac{1}{6} t^{\frac{c+\log_2 \rho}{c+1}} \geq \frac{1}{36} t^{\frac{c+\log_2 \rho}{c+1}}$$

- In the other case, if $m2^{o_t} > 2^{(o_t-1)(c+1)}$, it is equivalent to $2^{o_t c} > m > 2^{o_t c - c - 1} \geq 1$, it follows then:

$$
\begin{aligned}
36 A_{t+1}(\hat{\boldsymbol{s}}) &\geq 36 \times \frac{m-1}{2} \rho^{o_t} \\
&\geq 9 m \rho^{o_t} \\
&\geq 9 (m 2^{o_t})^{\frac{c+\log_2 \rho}{c+1}} \\
&> (4 \times 2^{(c+1)(o_t-1)} + m2^{o_t})^{\frac{c+\log_2 \rho}{c+1}} \\
&\geq t^{\frac{c+\log_2 \rho}{c+1}}
\end{aligned}
$$

Combine the results in the previous 2 cases, we obtain the result:

$$A_{t+1}(\hat{\boldsymbol{s}}) \geq \frac{1}{36} t^{\frac{c+\log_2 \rho}{c+1}}$$

For the second part of this lemma, according to our way to construct this stepsize schedule, we have:

$$
\begin{aligned}
t \geq \sum_{j=1}^{o_t-1} k_j 2^j \geq \sum_{j=1}^{o_t-1} 2^{jc} 2^j \geq \sum_{j=1}^{o_t-1} 2^{j(c+1)} \geq 2^{(c+1)(o_t-1)} \\
\Rightarrow t^{\frac{1}{c+1}} \geq 2^{o_t-1} \\
\Rightarrow 2t^{\frac{1}{c+1}} \geq 2^{o_t}
\end{aligned}
$$

which completes our proof for this lemma.

6.2 Proof of Lemma 3

The goal is to show that, given any $i \geq 1$ and $\alpha = \alpha_0$, we consider the i-th subsequence of silver stepsize schedule with length $k = 2^i$ that starts from a fix initialization point $\boldsymbol{x}_0$, if α satistifes $\alpha \geq (\sqrt{2}-1)A_k + \sqrt{2}$, for any stopping time $\ell \in [1, k-1]$, we have the following result:

$$f_\ell - f_0 \leq 432\alpha^2 \|\boldsymbol{g}_0\|^2$$

We also introduce 2 **Key Lemma** that are used in our proof and briefly show the proof:

Lemma 5. *let* $\boldsymbol{s}$ *be a primitive stepsize schedule that start from initial point* $\boldsymbol{x}_0$ *with gradient* $\boldsymbol{g}_0$*, then it have the following property:*

$$A_k(f_k - f^*) + \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{x}_0\|^2 + C_k\|\boldsymbol{g}_k\|^2 \leq \frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}_0\|^2 + \sum_{i=1}^{k-1} \alpha_i \langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle - \frac{A_k}{2}\|\boldsymbol{g}_0\|^2$$

To prove this lemma, we start from the definition of primitive stepsize schedule:

$$A_k(f_k - f^*) + \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + C_k\|\boldsymbol{g}_k\|^2 \leq \frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 + \sum_{i=1}^{k-1} \alpha_i (f_i - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}_0 \rangle + \frac{1}{2}\|\boldsymbol{g}_i\|^2)$$

Also make use of the property of f as a 1-smooth convex function:

$$\sum_{i=1}^{k-1} \alpha_i (f_i - f_0 - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}_0 \rangle + \frac{1}{2}\|\boldsymbol{g}_i - \boldsymbol{g}_0\|^2) \leq 0$$

Subtract the inequalities from $\sum_{i=1}^{k-1} \alpha_i (f_i - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}_0 \rangle + \frac{1}{2}\|\boldsymbol{g}_i\|^2)$, we obtain:

$$\begin{aligned}
&\sum_{i=1}^{k-1} \alpha_i (f_0 - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_0 - \boldsymbol{x}^* \rangle + \langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle - \frac{1}{2}\|\boldsymbol{g}_0\|^2) \\
=&\sum_{i=1}^{k-1} \alpha_i (f_i - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}_0 \rangle + \frac{1}{2}\|\boldsymbol{g}_i\|^2) - \sum_{i=1}^{k-1} \alpha_i (f_i - f_0 - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}_0 \rangle + \frac{1}{2}\|\boldsymbol{g}_i - \boldsymbol{g}_0\|^2) \\
\geq&\sum_{i=1}^{k-1} \alpha_i (f_i - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}^* \rangle + \frac{1}{2}\|\boldsymbol{g}_i\|^2)
\end{aligned}$$

Plug this into the inequalities obtained by the definition of primitive stepsize schedule, we derive the following result:

$$\begin{aligned}
&A_k(f_k - f^*) + \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + C_k\|\boldsymbol{g}_k\|^2 \\
\leq&\frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 + \sum_{i=1}^{k-1} \alpha_i (f_i - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_i - \boldsymbol{x}_0 \rangle + \frac{1}{2}\|\boldsymbol{g}_i\|^2) \\
\leq&\frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 + \sum_{i=1}^{k-1} \alpha_i (f_0 - f^* - \langle \boldsymbol{g}_i, \boldsymbol{x}_0 - \boldsymbol{x}^* \rangle + \langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle - \frac{1}{2}\|\boldsymbol{g}_0\|^2) \\
=&\frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 + (f_0 - f^*)\sum_{i=1}^{k-1} \alpha_i - \sum_{i=1}^{k-1} \langle \alpha_i \boldsymbol{g}_i, \boldsymbol{x}_0 - \boldsymbol{x}^* \rangle + \sum_{i=1}^{k-1} \alpha_i \langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle - \frac{1}{2}\|\boldsymbol{g}_0\|^2 \sum_{i=1}^{k-1} \alpha_i \\
=&\frac{1}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2 + A_k(f_0 - f^*) - \langle \boldsymbol{x}_1 - \boldsymbol{x}_k, \boldsymbol{x}_0 - \boldsymbol{x}^* \rangle + \sum_{i=1}^{k-1} \alpha_i \langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle - \frac{A_k}{2}\|\boldsymbol{g}_0\|^2
\end{aligned}$$

Together with the following observation:

$$\begin{aligned}
&\frac{1}{2}\left\|\boldsymbol{x}_1-\boldsymbol{x}^*\right\|^2-\frac{1}{2}\left\|\boldsymbol{x}_k-\boldsymbol{x}^*\right\|^2-\left\langle\boldsymbol{x}_1-\boldsymbol{x}_k, \boldsymbol{x}_0-\boldsymbol{x}^*\right\rangle \\
=&\frac{1}{2}\left\|\boldsymbol{x}_1\right\|^2-\frac{1}{2}\left\|\boldsymbol{x}_k\right\|^2-\left\langle\boldsymbol{x}_1, \boldsymbol{x}_0\right\rangle+\left\langle\boldsymbol{x}_k, \boldsymbol{x}_0\right\rangle \\
=&\frac{1}{2}\left\|\boldsymbol{x}_1-\boldsymbol{x}_0\right\|^2-\frac{1}{2}\left\|\boldsymbol{x}_k-\boldsymbol{x}_0\right\|^2,
\end{aligned}$$

we obtain the result in **lemma 5**:

$$A_k(f_k-f_0)+C_k\left\|\boldsymbol{g}_k\right\|^2+\frac{1}{2}\left\|\boldsymbol{x}_k-\boldsymbol{x}_0\right\|^2 \leq \frac{1}{2}\left\|\boldsymbol{x}_1-\boldsymbol{x}_0\right\|^2+\sum_{i=1}^{k-1}\alpha_i\left\langle\boldsymbol{g}_i, \boldsymbol{g}_0\right\rangle-\frac{A_k}{2}\left\|\boldsymbol{g}_0\right\|^2$$

Note that, we will use this lemma to upper bound the norm of gradient $\boldsymbol{g}_0$ and f_k-f_0 and next, we introduce the second lemma:

Lemma 6. *let $\boldsymbol{s}=\boldsymbol{\alpha}_{1:k-1}$ be any primitive stepsize schedule. Then one has*

$$\begin{aligned}
C_k\left\|\boldsymbol{g}_k\right\|^2 &\leq \left(\frac{\alpha_0^2}{2}+\frac{(A_k+1)^2}{2}-\alpha_0-\frac{A_k}{2}\right)\left\|\boldsymbol{g}_0\right\|^2 \\
f_k-f_0 &\leq \frac{1}{A_k}\left(\frac{1}{2}\alpha_0^2-\frac{A_k}{2}-\alpha_0+\frac{1}{2}\right)\left\|\boldsymbol{g}_0\right\|^2
\end{aligned}$$

Proof of the lemme directly follows the result of lemma 5; we first provide some basic facts about f as a 1-smooth function and this optimization process:

$$\boldsymbol{x}_1=\boldsymbol{x}_0-\alpha_0\boldsymbol{g}_0;; \tag{1}$$

$$\sum_{i=1}^{k-1}\alpha_i\boldsymbol{g}_i=\boldsymbol{x}_1-\boldsymbol{x}_k=\boldsymbol{x}_0-\boldsymbol{x}_k-\alpha_0\boldsymbol{g}_0; \tag{2}$$

These directly come from the gradient descent algorithm.

$$f_0-f_k \leq \left\langle\boldsymbol{g}_0, \boldsymbol{x}_0-\boldsymbol{x}_k\right\rangle-\frac{1}{2}\left\|\boldsymbol{g}_0-\boldsymbol{g}_k\right\|^2;; \tag{3}$$

This part is the transformed version of basic property of smooth function f:

$$f_i-f_k-\left\langle\boldsymbol{g}_i, \boldsymbol{x}_i-\boldsymbol{x}_j\right\rangle+\frac{1}{2}\left\|\boldsymbol{g}_i-\boldsymbol{g}_k\right\|^2 \leq 0$$

with $i=0$ and $j=k$.

Next, denote $\boldsymbol{a}=\boldsymbol{x}_0-\boldsymbol{x}_k$ and $\boldsymbol{b}=(A_k+1)\boldsymbol{g}_0$, applying Cauchy-Schwartz inequality, we have:

$$\begin{aligned}
&\langle\boldsymbol{a}, \boldsymbol{b}\rangle \leq \|\boldsymbol{a}\|\|\boldsymbol{b}\| \\
\Rightarrow&\left\langle\boldsymbol{x}_0-\boldsymbol{x}_k,(A_k+1)\boldsymbol{g}_0\right\rangle \leq \left\|\boldsymbol{x}_0-\boldsymbol{x}_k\right\|\left\|(A_k+1)\boldsymbol{g}_0\right\|
\end{aligned}$$

Also, for any real number a, b, the following inequality always holds:

$$ab \leq \frac{a^2}{2}+\frac{b^2}{2}$$

let $a = \|\boldsymbol{x}_0 - \boldsymbol{x}_k\|$ and $b = \|(A_k+1)\boldsymbol{g}_0\|$ and plug into the previous inequality:

$$\|\boldsymbol{x}_0 - \boldsymbol{x}_k\|\|(A_k+1)\boldsymbol{g}_0\| \le \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2 + \frac{1}{2}\|(A_k+1)\boldsymbol{g}_0\|^2$$

combine previous 2 results, we have:

$$\begin{aligned}
&\langle \boldsymbol{x}_0 - \boldsymbol{x}_k, (A_k+1)\boldsymbol{g}_0\rangle \le \|\boldsymbol{x}_0 - \boldsymbol{x}_k\|\|(A_k+1)\boldsymbol{g}_0\| \le \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2 + \frac{1}{2}\|(A_k+1)\boldsymbol{g}_0\|^2 \\
\Rightarrow &\langle \boldsymbol{x}_0 - \boldsymbol{x}_k, (A_k+1)\boldsymbol{g}_0\rangle \le \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2 + \frac{1}{2}\|(A_k+1)\boldsymbol{g}_0\|^2 \\
\Rightarrow &(A_k+1)\langle \boldsymbol{x}_0 - \boldsymbol{x}_k, \boldsymbol{g}_0\rangle \le \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2 + \frac{(A_k+1)^2}{2}\|\boldsymbol{g}_0\|^2;; \qquad (4)
\end{aligned}$$

With all 4 basic facts combined together, we obtain the first result:

$$C_k \left\|\boldsymbol{g}_k\right\|^2 \le \left(\frac{\alpha_0^2}{2} + \frac{(A_k+1)^2}{2} - \alpha_0 - \frac{A_k}{2}\right)\left\|\boldsymbol{g}_0\right\|^2$$

The second part of this lemma follows the result of **lemma 5**:

$$\begin{aligned}
&A_k(f_k - f_0) + C_k\left\|\boldsymbol{g}_k\right\|^2 + \frac{1}{2}\left\|\boldsymbol{x}_k - \boldsymbol{x}_0\right\|^2 \le \frac{1}{2}\left\|\boldsymbol{x}_1 - \boldsymbol{x}_0\right\|^2 + \sum_{i=1}^{k-1}\alpha_i\left\langle \boldsymbol{g}_i, \boldsymbol{g}_0\right\rangle - \frac{A_k}{2}\left\|\boldsymbol{g}_0\right\|^2 \\
\Rightarrow &A_k(f_k - f_0) + \frac{1}{2}\left\|\boldsymbol{x}_k - \boldsymbol{x}_0\right\|^2 \\
&\le \frac{1}{2}\left\|\boldsymbol{x}_1 - \boldsymbol{x}_0\right\|^2 + \sum_{i=1}^{k-1}\alpha_i\left\langle \boldsymbol{g}_i, \boldsymbol{g}_0\right\rangle - \frac{A_k}{2}\left\|\boldsymbol{g}_0\right\|^2 \\
((1),(2)) &\le \frac{\alpha_0^2}{2}\left\|\boldsymbol{g}_0\right\|^2 + \left\langle \boldsymbol{x}_0 - \boldsymbol{x}_k - \alpha_0\boldsymbol{g}_0, \boldsymbol{g}_0\right\rangle - \frac{A_k}{2}\left\|\boldsymbol{g}_0\right\|^2 \\
&= (\frac{\alpha_0^2}{2} - \frac{A_k}{2} - \alpha_0)\left\|\boldsymbol{g}_0\right\|^2 + \left\langle \boldsymbol{x}_0 - \boldsymbol{x}_k, \boldsymbol{g}_0\right\rangle \\
&\le (\frac{\alpha_0^2}{2} - \frac{A_k}{2} - \alpha_0)\left\|\boldsymbol{g}_0\right\|^2 + \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{x}_0\|^2
\end{aligned}$$

where the last inequality is derived using the basic inequality $2\langle \boldsymbol{a}, \boldsymbol{b}\rangle \le \|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2$ and complete the proof.

With lemma 5 and lemma 6, now we present the proof of **lemma 3**:

We first Denote $\widehat{A}_j = \rho^j - 1$ be the sum of stepsize of the j-th order silver stepsize schedule for $j \ge 1$. Consider any $\ell \in [1, k-1]$. Then there exist $1 \le p \le i$ and $i > m_1 > m_2 > \cdots > m_p \ge 0$ such that

$$\ell \;=\; \sum_{j=1}^{p} 2^{m_j}.$$

Also, denote

$$\tau_0 = 0 \quad \text{and} \quad \tau_j = \sum_{j'=1}^{j} 2^{m_{j'}},$$

so that $\tau_p = \ell$.

Now consider the stepsize schedule $\alpha_{\tau_j:\tau_{j+1}-1} = [\alpha_i]_{\tau_j \le i < \tau_{j+1}}$, whose length is $\tau_{j+1} - \tau_j = 2^{m_{j+1}}$. By construction, $\alpha_{\tau_j+1:\tau_{j+1}-1}$ corresponds to the (m_{j+1})-th order silver stepsize schedule, and

$$\alpha_{\tau_j+1} \;=\; (\sqrt{2}-1)\,\widehat{A}_{m_{j+1}} + \sqrt{2} \qquad \text{for all } j.$$

Combining this with $\widehat{A}_j = \rho^j - 1$ and the assumption $\alpha \ge (\sqrt{2}-1)A_k + \sqrt{2}$ (recall $\alpha = \alpha_0$) yields

$$\alpha_{\tau_j+1} \;\le\; \frac{\alpha_{\tau_j}}{2} \tag{1}$$

provided that $j \ge 0$ and $m_{j+1} \ge 2$.

Invoking Lemma 6 with this stepsize schedule, we can demonstrate that

$$\widehat{A}_{m_{j+1}}\big(\widehat{A}_{m_{j+1}}+1\big)\,\|\mathbf{g}_{\tau_{j+1}}\|^2 \;\le\; \Big(\alpha_{\tau_j}^2 - 2\alpha_{\tau_j} + \widehat{A}_{m_{j+1}}^2 + \widehat{A}_{m_{j+1}} + 1\Big)\,\|\mathbf{g}_{\tau_j}\|^2 \tag{2}$$

$$f_{\tau_j+1} - f_{\tau_j} \le \frac{1}{\hat{A}_{m_j+1}}\left(\frac{1}{2}\alpha_{\tau_j}^2 - \frac{\hat{A}_{m_j+1}}{2} - \alpha_{\tau_j} + \frac{1}{2}\right)\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \le \frac{1}{2}\alpha_{\tau_j}^2\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \tag{3}$$

It then follows from (2) that

$$\begin{aligned}\left\|\boldsymbol{g}_{\tau_j+1}\right\|^2 &\le \frac{\alpha_{\tau_j}^2 + \hat{A}_{m_j+1}^2 + \hat{A}_{m_j+1} + 1}{\hat{A}_{m_j+1}(\hat{A}_{m_j+1}+1)}\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \\ &\le \left(\frac{\alpha_{\tau_j}^2}{\hat{A}_{m_j+1}(\hat{A}_{m_j+1}+1)} + 1\right)\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \end{aligned} \tag{4}$$

which we would like further control by dividing into two cases:

Case 1: $m_{j+1} \ge 2$. In this case, we have $\hat{A}_{m_{j+1}} \ge \rho^2 - 1 = 2 + 2\sqrt{2}$. Observing that $\alpha_{\tau_{j+1}} = (\sqrt{2}-1)\hat{A}_{m_{j+1}} + \sqrt{2}$ by construction, one can easily verify that

$$\hat{A}_{m_{j+1}}(\hat{A}_{m_{j+1}}+1) \ge (\sqrt{2}+1)\alpha_{\tau_{j+1}}^2,$$

which combined with (4) implies that

$$\left\|\boldsymbol{g}_{\tau_{j+1}}\right\|^2 \le \left(\frac{\alpha_{\tau_j}^2}{\alpha_{\tau_{j+1}}^2}\cdot(\sqrt{2}-1) + 1\right)\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \tag{5}$$

This taken together with the property $\alpha_{\tau_{j+1}} \le \alpha_{\tau_j}/2$ leads to

$$\alpha_{\tau_{j+1}}^2\left\|\boldsymbol{g}_{\tau_{j+1}}\right\|^2 \le \left(\sqrt{2} - \frac{3}{4}\right)\alpha_{\tau_j}^2\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \tag{6}$$

Case 2: $m_{j+1} < 2$. In this case, it is readily seen from (4) that

$$\left\|\boldsymbol{g}_{\tau_{j+1}}\right\|^2 \le \left(\frac{\alpha_{\tau_j}^2}{\hat{A}_{m_{j+1}}(\hat{A}_{m_{j+1}}+1)} + 1\right)\left\|\boldsymbol{g}_{\tau_j}\right\|^2 \le \alpha_{\tau_j}^2\left\|\boldsymbol{g}_{\tau_j}\right\|^2. \tag{7}$$

Moreover, we make the observation that

$$\alpha_{\tau_{j+1}} = (\sqrt{2}-1)\hat{A}_{m_{j+1}} + \sqrt{2} \le (\sqrt{2}-1)(\rho-1) + \sqrt{2} = 2,$$

which allows us to reach

$$\alpha_{\tau_{j+1}}^2\left\|\boldsymbol{g}_{\tau_{j+1}}\right\|^2 \le 12\alpha_{\tau_j}^2\left\|\boldsymbol{g}_{\tau_j}\right\|^2. \tag{8}$$

s

Putting (6) and (8) together, we can conclude that for any $j \geq 1$,

$$\alpha_{\tau_j}^2 \left\|\boldsymbol{g}_{\tau_j}\right\|^2 \leq 432 \left(\sqrt{2} - \frac{3}{4}\right)^j \alpha_{\tau_0}^2 \left\|\boldsymbol{g}_{\tau_0}\right\|^2 = 432 \left(\sqrt{2} - \frac{3}{4}\right)^j \alpha^2 \left\|\boldsymbol{g}_0\right\|^2 \tag{9}$$

This taken collectively with (3) gives

$$\begin{aligned} f_\ell - f_0 &= \sum_{j=0}^{p-1} (f_{\tau_{j+1}} - f_{\tau_j}) \leq \frac{1}{2} \sum_{j=0}^{p-1} \alpha_{\tau_j}^2 \left\|\boldsymbol{g}_{\tau_j}\right\|^2 \\ &\leq \left(\frac{1}{2} + 216 \sum_{j \geq 1} \left(\sqrt{2} - \frac{3}{4}\right)^j\right) \alpha^2 \left\|\boldsymbol{g}_0\right\|^2 \leq 432\alpha^2 \left\|\boldsymbol{g}_0\right\|^2 \end{aligned} \tag{10}$$

as claimed.

6.3 Apply lemma 3 to derive upper bound for $f_\ell - f_{t_i+1}$

Note that, in this part, we can treat the optimization process from t_i+2 to ℓ as a optimization using silver stepsize schedule starting at x_{t_i+1} and stopped at $\ell - (t_i + 1)$; also note that, in the process of analyzing α_{t_i+1}, we find a lower bound for α_{t_i+1} as $\alpha_{t_i+1} \geq (\sqrt{2} - 1)y + \sqrt{2}$ while $y = \sum_{j=t_i+2}^{t_{i+1}} \alpha_j$ is the sum of the complete silver stepsize schedule, and this implies that we can safely apply **lemma 3** to upper bound $f_\ell - f_{t_i+1}$.

By the result of **lemma 3**, we have:

$$f_\ell - f_{t_i+1} \leq 432\alpha_{t_i+1}^2 \|\boldsymbol{g}_{t_i+1}\|^2$$

Recall that, we derive 2 important inequalities by definition of primitive stepsize schedule, one of them is:

$$\|\boldsymbol{g}_{t_i+1}\|^2 \leq \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2}{A_{t_i+1}^2}$$

Plug this in, we have:

$$432\alpha_{t_i+1}^2 \|\boldsymbol{g}_{t_i+1}\|^2 \leq \mathcal{O}\bigg(\frac{\alpha_{t_i+1}^2}{A_{t_i+1}^2} \|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2\bigg)$$

By **lemma 2**, we have:

$$A_{t_i+1}(\hat{\boldsymbol{s}}) \geq \frac{1}{36} t_i^{\frac{c+\log_2 \rho}{c+1}}$$

Also recall the upper bound we found for α_{t_i+1}:

$$\alpha_{t_i+1} \leq \rho^{o_t+1} + 1 = \mathcal{O}\big(\rho^{o_{t_i+1}}\big) \leq \mathcal{O}\big(t_i^{\frac{\log_2 \rho}{c+1}}\big)$$

The last step in the inequality above is derived by the second part of **lemma 2**:

$$\begin{aligned} & 2^{o_t} \leq 2t^{\frac{1}{c+1}} \\ \Rightarrow & o_t \leq \log_2 \big(2t^{\frac{1}{c+1}}\big) = 1 + \frac{1}{c+1} \log_2 t \end{aligned}$$

and we make the following transformation for $\rho^{o_{t_i+1}}$:

$$\begin{aligned}
\rho^{o_{t_i+1}} &= \exp\left(o_{t_i+1} \ln\rho\right) \\
&\le \exp\left(\left(1+\frac{1}{c+1}\log_2 t_i\right)\ln\rho\right) \\
&= \rho\times\exp\left(\frac{\ln\rho}{(c+1)\ln 2}\ln t_i\right) \\
&= \rho\times t_i^{\frac{\ln\rho}{(c+1)\ln 2}} \\
&= \rho\times t_i^{\frac{\log_2\rho}{c+1}}
\end{aligned}$$

This means:

$$\mathcal{O}\big(\rho^{o_{t_i+1}}\big) = \mathcal{O}(t_i^{\frac{\log_2\rho}{c+1}})$$

which is the last step in the inequality

Then:

$$\begin{aligned}
432\alpha_{t_i+1}^2\|\boldsymbol{g}_{t_i+1}\|^2 &\le \mathcal{O}\bigg(\frac{\alpha_{t_i+1}^2}{A_{t_i+1}^2}\|\boldsymbol{x}_1-\boldsymbol{x}^*\|^2\bigg) \\
&\le \mathcal{O}\left(\frac{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|^2 t_i^{\frac{2\log_2\rho}{c+1}}}{t_i^{\frac{2(c+\log_2\rho)}{c+1}}}\right) \\
&= \mathcal{O}\left(\frac{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|^2}{t_i^{\frac{2c+2\log_2\rho-2\log_2\rho}{c+1}}}\right) \\
&= \mathcal{O}\left(\frac{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|^2}{t_i^{\frac{2c}{c+1}}}\right) \\
&= \mathcal{O}\left(\frac{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|^2}{\ell^{\frac{2c}{c+1}}}\right)
\end{aligned}$$

Note that, the last step is derive by the fact we mentioned in the beginning of the proof of Theorem 1:

$$\begin{aligned}
&t_{i+1} = t_i + 2^{o_{t_i}+1} \le 3t_i \\
\Rightarrow &t_i \le t_{i+1} \le 3t_i
\end{aligned}$$

and by our assumption, $t_i < \ell \le t_{i+1}$, we have:

$$t_i < \ell \le t_{i+1} \le 3t_i$$

which explains why the last step in our derivation is valid and completes our proof.

7 Additional Experiments

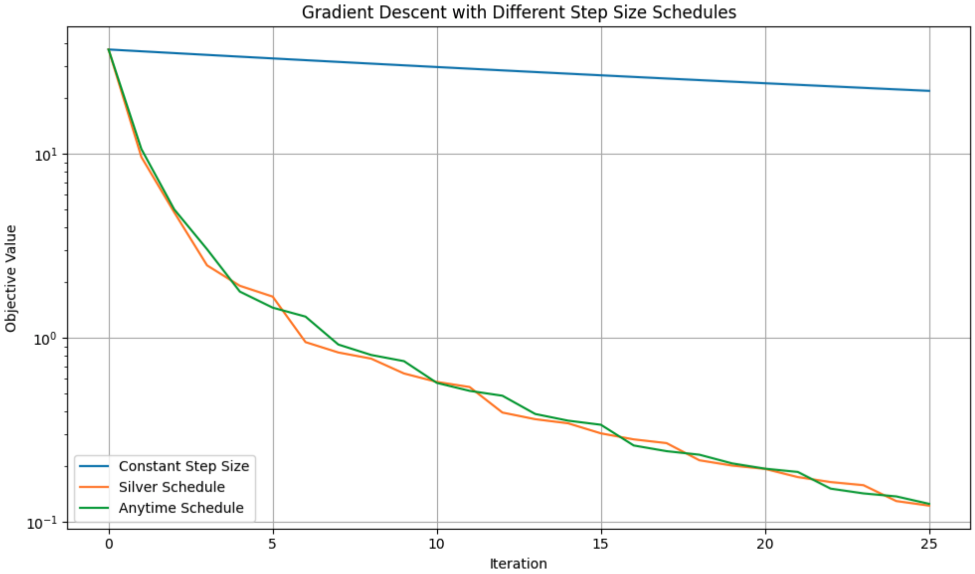

Figure 1: Comparison of different step size schedules in gradient descent on a synthetic quadratic function.

7.1 More Simulation Results

In this section, we applied numerical simulation on several 1-smooth functions using 3 types of stepsize schedules: (1) constant stepsize with $\alpha_i = 1$; (2) silver stepsize schedule; and (3) anytime stepsize schedule; The source code used for simulation is available at GitHub Repository. We expect to observe the following phenomena from the results:

- For **any stopping time**, the silver stepsize schedule and anytime schedule should achieve acceleration compared with constant stepsize schedule;

- For **some stopping time** $T = 2^i - 1$, **silver stepsize schedule** should achieve the optimal convergence rate;

- For stopping time T **other than** $2^i - 1$, at least at some of them, **anytime stepsize schedule** outperform silver stepsize schedule.

Following the setting, we select the following 1-smooth convex function to simulate on:

(1) Anisotropic Quadratic Function:

$$f_1(x) = x^\top A x = \sum_{i=1}^{d} \lambda_i x_i^2, \quad A = \operatorname{diag}(\lambda_1, \ldots, \lambda_d),\ \lambda_i \in [0.001, 0.1]$$

(2) Log-Sum-Exp Function:

$$f_2(x) = \log\left(\sum_{i=1}^{m} \exp(a_i^\top x)\right), \quad a_i \in \mathbb{R}^d$$

(3) Multivariate Logistic Loss (Binary Classification):

$$f_3(x) = \sum_{i=1}^{n} \log\left(1 + \exp(-b_i a_i^\top x)\right), \quad b_i \in \{-1, +1\},\ a_i \in \mathbb{R}^d$$

(4) Softmax Cross-Entropy Loss:

$$f_4(x) = -\sum_{i=1}^{n} \log\left(\frac{\exp(a_{y_i}^\top x)}{\sum_{j=1}^{C} \exp(a_j^\top x)}\right), \quad y_i \in \{1, \ldots, C\}$$

For each function, we start from a random initial point and applied different stepsize schedule and record the history of f_t and x_t; we first compare the convergence rate between different schedules across all 4 functions via the following Figure 2:

Figure 2 shows that, across all 4 functions, as we expected, both **silver schedule** and **anytime schedule** outperformed **constant schedule** at any stopping time;

We also observed that, although the convergence rate between silver schedule and anytime schedule at early stage of optimization, silver schedule converges faster than anytime schedule at most of iteration;

Note that, in the case of **Multivariate Logistic Loss** function, the curves showed that at some point, **anytime schedule** performed better than **silver schedule**.

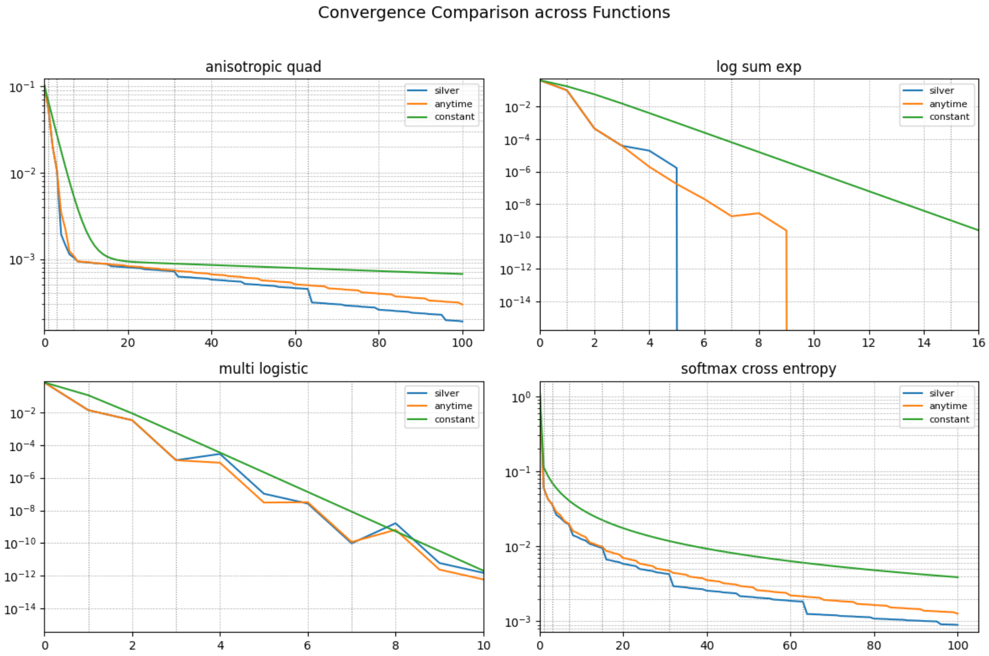

Figure 2: Comparison on Convergence Rates

We also present the trajectories of 3 schedules to help gain some intuition about the differences between them as shown in Figure 3:

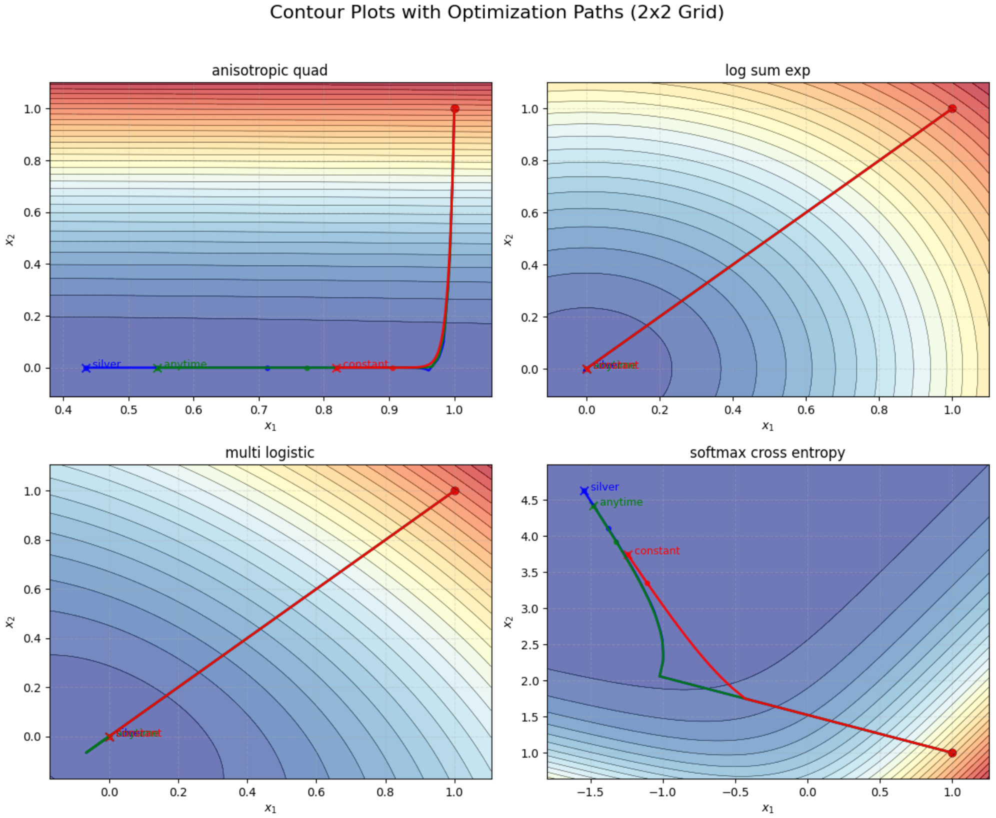

Figure 3: Trajectory and Contour map

References

Altschuler, J. M. and P. A. Parrilo (2023). Acceleration by stepsize hedging ii: Silver stepsize schedule for smooth convex optimization. *arXiv preprint arXiv:2309.16530*.

Grimmer, B., K. Shu, and A. Wang (2024). Composing optimized stepsize schedules for gradient descent. *arXiv preprint arXiv:2410.16249*.

Kornowski, G. and O. Shamir (2024). Open problem: Anytime convergence rate of gradient descent. In *Conference on Learning Theory*, pp. 5335–5339.

Zhang, Z. and R. Jiang (2024). Accelerated gradient descent by concatenation of stepsize schedules. *arXiv preprint arXiv:2410.12395*.

Zhang, Z., J. D. Lee, S. S. Du, and Y. Chen (2024). Anytime acceleration of gradient descent. *arXiv preprint arXiv:2411.17668*.