OPT2021: 13th Annual Workshop on Optimization for Machine Learning

On Server-Side Stepsizes in Federated Optimization: Theory Explaining the Heuristics

Grigory Malinovsky GRIGORII.MALINOVSKII@KAUST.EDU.SA
Konstantin Mishchenko KONSTA.MISH@GMAIL.COM
Peter Richtárik PETER.RICHTARIK@KAUST.EDU.SA
KAUST, Saudi Arabia

Abstract

We present a theoretical study of server-side optimization in federated learning. Our results are the first to show that the widely popular heuristic of scaling the client updates with an extra parameter is extremely useful in the context of Federated Averaging (FedAvg) with local passes over the client data. In particular, we prove that whenever the local stepsizes are small and the update direction is given by FedAvg over all clients, one can take a big leap in the obtained direction and improve rates for convex, strongly convex, and non-convex objectives. Especially, we get enhancement in non-convex rate of convergence from $\mathcal{O}(\varepsilon^{-3})$ to $\mathcal{O}(\varepsilon^{-2})$. In contrast, if the local stepsizes are large, we prove that the noise of client sampling can be controlled by using a small server-side stepsize. Together, our results on the advantage of large and small server-side stepsizes give a formal justification for the practice of adaptive server-side optimization in federated learning. Moreover, we consider a variant of algorithm that supports partial client participation where only a subset of devices communicates in each round, which makes the method more practical.

1. Introduction

The unprecedented industrial success of modern machine learning techniques, tools and models can to a large degree be attributed to the abundance of data available for training. Indeed, the most popular and best performing deep learning models rely on a very large number of parameters, and in order to generalize well, need to be trained using optimization algorithms over very large training data sets. Other things equal, the more data we have, the better. A key driving force behind the proliferation of such data is the massive digitization of society of the last few decades. People have access to increasingly more elaborate personal and home smart devices capable of generating, capturing and processing data such as text, images and videos. Similarly, in the sphere of governments and corporations, much of what used to be done through a physical exchange (e.g., via paper/fax/letter) is now performed in a digital form, generating treasure troves of potentially useful data. For example, hospitals collect, store and make us of a variety of patient data, ranging from routine bodily functions to PET scans and genome sequencing.

1.1. Federated learning

The traditional way of learning from this data is to collect it in a single (and often proprietary) data center, where it is subsequently processed using modern machine learning algorithms. However, due to several considerations which keep gaining in importance, such as energy efficiency and privacy, it is often desirable to avoid centralized training altogether, and instead perform the training without the data ever leaving the clients' secure sites. Introduced in 2016 by Konečný et al. [14], Konečný et al. [15], McMahan et al. [18], this is precisely the promise and subject of study of *federated learning (FL)*. In other words, federated learning means efficient machine learning over data stored in a distributed fashion across a network of heterogeneous clients (e.g., mobile phones, smart devices, companies) that captured and own the data, using these clients' machines/devices not only as data sources, but also as computers that can contribute to the training.

1.2. Problem formulation

We consider the standard optimization formulation of federated learning

$$\min_{x\in\mathbb{R}^d} \left[f(x) \overset{\text{def}}{=} \frac{1}{M} \sum_{m=1}^{M} f_m(x) \right], \tag{1}$$

where M is the total number of clients, $x \in \mathbb{R}^d$ represents the parameters of the model we wish to train, and $f_m : \mathbb{R}^d \to \mathbb{R}$ is the loss of model x on the training data owned by client $m \in [M] \overset{\text{def}}{=} \{1, 2, \dots, M\}$. Typically, M is very large.

Since the training data set on each client is necessarily finite, we assume that f_m has the finite-sum structure

$$f_m(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_m^i(x), \tag{2}$$

where $f_m^i : \mathbb{R}^d \to \mathbb{R}$ is the loss of model x on training example $i \in [n] \overset{\text{def}}{=} \{1, 2, \dots, n\}$ stored on client m. We assume that the functions f_m^i are differentiable, and consider the strongly convex, convex and non-convex regimes.

Table 1: Conceptual comparison of results for FedAvg from prior work with our results.

Partial participation	Local training	Data shuffling	**Large** server stepsizes help	**Small** server stepsizes help	Reference
✓	✓	✗	✓	✗	[11]
✓	✓	✗	✓	✗	[27]
✗	✓	✗	✗	✗	[13]
✗	✓	✗	✗	✗	[12]
✗	✓	✓	✗	✗	[20]
✓	✓	✓	✓	✓	**This paper**

1.3. Ingredients of successful federated learning methods

Practical considerations of federated learning systems and vast experimental evidence accrued over the last few years point to several design constraints and algorithmic ingredients which have proved useful in the context of federated learning methods for solving (1)-(2). We now very briefly outline some of them here. More details can be found in the appendix where we review related work.

Partial participation. In federated learning, training is performed through several communication rounds in each of which an orchestrating server chooses a *cohort* of clients that will be participating in the training process in that round. This practice is known as *partial participation*, and is necessary due to practical considerations and limitations, such as limited server capacity, and limited client availability [10]. However, partial participation can be useful also due to the diminishing returns one gets as the number of participating clients grows [5]. Partial participation is a necessity in the cross-device regime where the training is performed over a very large number of clients (i.e., M is very large) most of which will only participate in the entire training procedure at most once. Sampling of clients to form a cohort can be done adaptively so as to choose the most informative clients [6].

Local training. At the beginning of each communication round, each client in the cohort is provided with the latest model by the orchestrating server, which is used as a starting point for *local training*. Local training refers to the common practice in FL of performing several steps of a suitably chosen local optimization procedure, such as one of the many variants of SGD, using its own local training data. Perhaps the simplest approach is to perform a single local GD iteration. If the model updates are simply just aggregated by the server, then the resulting method can be seen as Minibatch SGD, where the minibatches correspond to the cohorts. However, it is typically more efficient to perform *multiple* local steps [18], and to use local optimizers that rely on *incremental* data processing, such as SGD.

Data shuffling. Typically, the local training data set is processed once or several times in an incremental fashion, that is, one data point (or one small minibatch) at a time. However, experimental evidence shows that processing the local data *without replacement* can lead to substantially better results than processing the data *with replacement*. In particular, processing the local training data in an order dictated by a random permutation—a technique known as Random Reshuffling (RR)—is often set as default in modern deep learning and federated learning software [2, 3, 26]. This is in sharp contrast with the *with-replacement* sampling of data employed by SGD. With replacement sampling ensures that the gradient updates are unbiased, and this simplified the analysis. For this reason, SGD is significantly better understood in theory than its better performing but much more poorly understood cousin RR. However, recent results of Mishchenko et al. [19], and extensions of Mishchenko et al. [20] and Yun et al. [28] to distributed training, show that RR can have clear theoretical advantages over SGD.

Server stepsizes. Once local training is finished, the clients in the cohort send their models or model updates to the orchestrating server, which typically aggregates them via averaging. This information is then used to perform *server side* optimization. The simplest approach is to do nothing - that is, to treat the aggregated models as the next global model that is broadcast to the new cohort in the next communication round. However, empirical evidence suggests that it is better to aggregate *model updates*, and treat them as gradient-type information which can be injected into a suitably chosen server side optimization routine [11]. For example, the server may run one step of GD using the aggregated model update as a proxy for the gradient which is not available, with its own server-side stepsize.

Further useful tricks. Additional tricks that are often employed in the context of federated learning include the use of compressed communication [1, 8], drift reduction [7, 11], error compensation [23, 25], server side momentum [9], and adaptive stepsize selection [22]. These techniques are beyond the scope of this paper.

2. Summary of Contributions

Despite the fact that *partial participation*, *local training*, *data shuffling* and *server stepsizes* have all been empirically found to be very useful building blocks of FL methods, most of these techniques are not very well understood in theory even in isolation. Informally speaking, and at the risk of oversimplifying the current state of affairs, we know virtually nothing about *server stepsizes*, very little about *data shuffling*, relatively much more about *local training*, and quite a bit but still "not enough" about *partial participation*.

> The key focus of this paper is to make a substantial advance in the current theoretical understanding of *server stepsizes* in the context of *realistic* federated learning.

In order to theoretically understand the server stepsize phenomenon in a realistic context of techniques commonly used in FL, we study this phenomenon *together* with data shuffling, local training and partial participation. While this makes the analysis substantially harder and different from all[1] existing analyses of FedAvg, we believe it is important to fo so as this will highlight the *interplay* between these algorithmic techniques and their *combined* impact on training.

A brief visual summary of this in the context of selected existing methods is provided in Table 3. We summarize our contributions as follows:

• **New algorithm.** We design a new algorithm, for which we coin the name Nastya (Algorithm 1; see Section 4), which combines all the of the aforementioned practical tricks and techniques in a single method: *partial participation*, *local training*, *data shuffling* and, most importantly, *server stepsizes*. In our method, in each communication round t, the cohort is chosen as a random subset S_t of the set $\{1, 2, \ldots, M\}$ of clients of cardinality $1 \leq C \leq M$, chosen uniformly from all subsets of cardinality C. Each device performs local training via a single pass of incremental GD with *client stepsize* $\gamma > 0$ over the local training data points in an order dictated by a *random permutation*. We allow for two options: i) either the random permutation for all clients is sampled just once and used in all communication rounds (*Shuffle-Once* option), or ii) the random permutation is sampled afresh at the start of each communication round (*Random-Reshuffling* option). At the end of local training, the updated models are communicated back to the server, which uses these updates to form a *gradient estimator*, and applies one step of GD using a server stepsize $\eta > 0$ with this estimator in lieu of the true gradient. The new model is then broadcast to a new cohort in the next communication round, and the process is repeated.

1. Except for the recent work of Mishchenko et al. [20] which we used as an inspiration.

Table 2: The main convergence results obtained in this paper (also see Thm 20).

Regime	Stepsizes	Result[1]
μ-Convex (Thm 6)	$\gamma n \le \eta \le \frac{1}{16L}$	$\mathbb{E}\lVert x_T - x_* \rVert^2 \le \left(1 - \frac{\eta\mu}{2}\right)^T \lVert x_0 - x_* \rVert^2 + \frac{5\gamma^2 nL}{\mu}\Sigma_*^2 + \frac{8\eta}{\mu}\frac{M-C}{C\max\{1,M-1\}}\sigma_*^2$
Convex (Thm 15)	$\gamma n \le \eta \le \frac{1}{16L}$	$\mathbb{E}\left[f(\hat{x}_T) - f(x_*)\right] \le \frac{5\lVert x_0 - x_* \rVert^2}{2\eta T} + 7\gamma^2 nL\Sigma_*^2 + 10\eta\frac{M-C}{C(\max\{1,M-1\})}\sigma_*^2$
Non-convex (Thm 19)	$\gamma \le \frac{1}{2nL}$ & $\eta \le \frac{1}{L}$	$\min_{t=0,\dots,T-1} \mathbb{E}\lVert \nabla f(x_t) \rVert^2 \le \frac{2\left(1+4\eta\gamma^2 n^2 L^3\right)^T}{\eta T}\delta_0 + 2\gamma^2 nL^3 D_*^2 + 4L^2\eta\frac{M-C}{C\max\{1,M-1\}}\Delta_*$

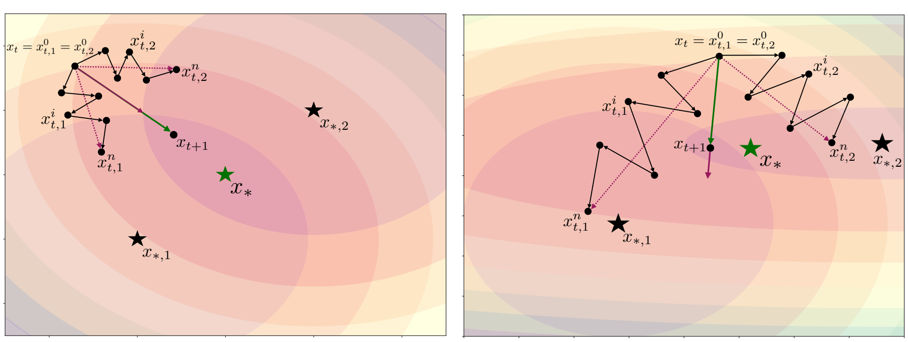
[1] γ = client stepsize; η = server stepsize; M = total # of clients; C = # of participating clients (cohort size); n = # of training data points per client; L = Lipschitz constant of the gradient of f; μ = strong convexity constant of f; T = total # of communication rounds; x_0 = initial model; x_* = optimal model; $\delta_0 = f(x_0) - f_*$; $\Sigma_*^2 = \left(\frac{1}{M}\sum_{m=1}^{M}\sigma_{*,m}^2 + n\sigma_*^2\right)$; $\sigma_*^2 = \frac{1}{M}\sum_{m=1}^{M}\|\nabla f_m(x_*)\|^2$; $\sigma_{*,m}^2 = \frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_m^i(x_*)\right\|^2$; $D_*^2 = \left(\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m} + n\Delta_*\right)$; $\Delta_* = \frac{1}{M}\sum_{m=1}^{M}(f_{*,m} - f_*) \ge 0$; $\Delta_{*,m} = \frac{1}{n}\sum_{i=1}^{n}(f_* - f_{*,m}^i) \ge 0$, where $f_* = \inf f$, $f_{*,m} = \inf f_m$ and $f_{*,m}^i = \inf f_m^i$ are all assumed to be finite (i.e., not $-\infty$).

• **Complexity analysis.** We provide strong complexity analysis of our new algorithm for strongly convex (Theorem 6), convex (Theorem 15) and non-convex (Theorem 19) functions; see Table 2. This is the first theory for a variant of FedAvg that combines the benefits of partial participation, data shuffling, local training and, most importantly, also *server stepsizes*. Most importantly, with a couple exceptions only [11, 27], there are no prior theoretical works analyzing the effect of server stepsizes in FL. The methods in the aforementioned works use local training and partial participation, but do not use data shuffling, and are significantly different from ours.

• **Small client stepsizes, large server stepsizes, and no need for drift reduction.** In particular, Theorems 6, 15 and 19, covering the strongly convex, convex and non-convex regimes, respectively, suggest that the server can use the large $\mathcal{O}(1/L)$ stepsize, where L is the Lipschitz constant of the gradient of f. In the strongly convex and convex regimes, based on our theory, it is optimal for the client stepsize γ to be small, which completely eliminates the second of the three terms in the complexity bounds (see the third column of Table 2) which controls the price one pays due to *data heterogeneity*. Indeed, our theory allows for the client stepsize γ to be small while the server stepsize η can be large (see the second column of Table 2). Note that in all three regimes, and thanks to the fact that we employ a *data shuffling* strategy, this second term depends on the square γ^2 of the client stepsize, which means that we can make this term small without making the client stepsizes infinitesimal. So, thanks to Nastya's use of data shuffling strategies, it does *not* require any explicit drift reduction technique such as SCAFFOLD to handle data heterogeneity [11].

• **Small server stepsizes can be beneficial.** To the best of our knowledge, no prior theoretical work suggests that it might be beneficial to use *small* server stepsizes. Our results (see Theorem 20) suggests that this can be the case when each f_m^i is strongly convex and smooth, and when the strong convexity parameter is very small.

• **Experimental validation of our theoretical predictions.** We provide experimental examination of Nastya and compare it with selected benchmarks. Our goal is not to perform large scale experiments and claim empirical superiority because the algorithmic ingredients embedded in Nastya already *are* being used in practical FL methods precisely because they have already been empirically found to be useful. This allows us to focus on simple experiments which test the theoretical predictions of our theory. Our experimental results confirm our theory, and illustrate the behavior of the methods we test in various settings. Moreover, we go beyond the theory and conduct additional experiments with the adaptive stepsize strategy introduced by Malitsky and Mishchenko [17]. Inspired by Reddi et al. [22], we additionally utilize several server-side optimization subroutines on top of the local updates.



(a) (b)

Figure 1: Illustration of the dependence between server and client stepsizes on a simple example with $M = 2$ clients. $x_{*,1}$ and $x_{*,2}$ are the minimizers of the local functions f_1 and f_2, respectively, and x_* is the minimizer of the global function $f = \frac{1}{2}f_1 + \frac{1}{2}f_2$. **(a)** In the case of small client stepsizes γ, the average of local steps is not large, but at the same time the variance is small and the direction is close to direction of the full gradient, which allows us to go further towards this direction by employing a large server stepsize η. **(b)** In the case of large client stepsizes γ, each client step contributes to the global step, but the variance grows as well, so it is useful to use smaller server stepsize η to reduce this variance. These intuitions are confirmed by our theory.

3. Preliminaries

In this section we introduce several key concepts that will help us to formulate our theoretical results.

3.1. Convexity and smoothness

In all our theoretical results we rely on smoothness, and in some we require convexity or strong convexity.

Definition 1 (L-smoothness) *Function $\phi\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth if it has L-Lipschitz continuous gradient for some $L > 0$*

$$\|\nabla\phi(x) - \nabla\phi(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^d. \tag{3}$$

Definition 2 (Convexity and strong convexity) *Function $\phi\colon \mathbb{R}^d \to \mathbb{R}$ is convex if $\forall x, y \in \mathbb{R}^d$*

$$\phi(y) \geq \phi(x) + \langle\nabla\phi(x), y - x\rangle, \tag{4}$$

and μ-strongly convex if $\forall x, y \in \mathbb{R}^d$

$$\phi(y) \geq \phi(x) + \langle\nabla\phi(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2. \tag{5}$$

In our analysis we use the following assumption.

Assumption 1 *The objective f and the individual losses $f_m^1, \ldots, f_m^n$ are all L-smooth. Further, for all i and $m \in \{1, 2, \ldots, M\}$ and $i \in \{1, 2, \ldots, n\}$, (i) $f_* \overset{def}{=} \inf_x f(x) > -\infty$, (ii) $f_{*,m} \overset{def}{=} \inf_x f_m(x) > -\infty$, and (iii) $f_{*,m}^i \overset{def}{=} \inf_x f_m^i(x) > -\infty$. If f_m^i is convex, we further assume the existence of minimizers $x_* = \arg\min_{x\in\mathbb{R}^d} f(x)$ and $x_{*,m}^i = \arg\min_{x\in\mathbb{R}^d} f_m^i(x)$.*

3.2. Measures of data heterogeneity

While our theory does not require any *assumptions* on data homogeneity, our *results* will reflect the degree to which the data is heterogeneous, and are better for data that is "more" homogeneous. In particular, in the strongly convex and convex regimes we rely on the following notions.

Definition 3 (Variance at the optimum) *The variance of the gradients $\{\nabla f_m\}_{m=1}^M$ at x_* is defined as*

$$\sigma_*^2 \overset{def}{=} \frac{1}{M}\sum_{m=1}^M \|\nabla f_m(x_*)\|^2,$$

where x_* is a minimizer of f. The variance of the gradients $\{\nabla f_m^i\}_{i=1}^n$ at x_* is

$$\sigma_{*,m}^2 \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \left\|\nabla f_m^i(x_*)\right\|^2.$$

An important lemma that allows us to obtain a strong upper bound for variance in the case of sampling without replacement, which our data shuffling methods rely on, was formulated by Mishchenko et al. [19]. We include it here for completeness.

Lemma 4 (Sampling without replacement) *Let $X_1, \ldots, X_n \in \mathbb{R}^d$ be fixed vectors, $\overline{X} \overset{def}{=} \frac{1}{n}\sum_{i=1}^n X_i$ be their average and $\sigma^2 \overset{def}{=} \frac{1}{n}\sum_{i=1}^n \left\|X_i - \overline{X}\right\|^2$ be the population variance. Fix any $k \in \{1, \ldots, n\}$, let $X_{\pi_1}, \ldots X_{\pi_k}$ be sampled uniformly without replacement from $\{X_1, \ldots, X_n\}$ and $\overline{X}_\pi$ be their average. Then, it holds*

$$\mathbb{E}\left[\overline{X}_\pi\right] = \overline{X}, \quad \mathbb{E}\left[\left\|\overline{X}_\pi - \overline{X}\right\|^2\right] = \frac{n-k}{k(n-1)}\sigma^2. \tag{6}$$

For non-convex functions, we use a different notion of data heterogeneity.

Definition 5 (Functional dissimilarity) *The variance at the optimum in the non-convex regime is defined as*

$$\Delta_* \overset{def}{=} \frac{1}{M}\sum_{m=1}^M f_{*,m} - f_*,$$

where $f_{*,m} = \inf_x f_m(x)$ and $f_* = \inf_x f(x)$. For each device m, the variance at the optimum is defined as

$$\Delta_{*,m} \overset{def}{=} \frac{1}{n}\sum_{i=1}^n f_{*,m}^i - f_*,$$

where $f_{*,m}^i = \inf_x f_m^i(x)$.

Again, the above is a definition and not an assumption. The concepts are well defined as long as Assumption 1 is satisfied.

Algorithm 1 Nastya: Federated optimization with server stepsize, random shuffling and partial participation

Input: client stepsize $\gamma > 0$; server stepsize $\eta \geq 0$; cohort size $C \in \{1, 2, \ldots, M\}$; initial iterate/model $x_0 \in \mathbb{R}^d$; number of communication rounds $T \geq 1$

Shuffle-Once option: For each client m, sample a permutation $\pi_m = (\pi_m^0, \pi_m^1, \ldots, \pi_m^{n-1})$ of $\{1, 2, \ldots, n\}$

for *communication round* $t = 0, 1, \ldots, T-1$ **do**

Sample a cohort S_t of C clients (server chooses a random set $S_t \subseteq \{1, 2, \ldots, M\}$ of size $|S_t| = C$, uniformly at random)

Send model x_t to all participating clients $m \in S_t$ (server broadcasts x_t to all clients $m \in S_t$) **for** *all clients $m \in S_t$, locally in parallel* **do**

$x_{t,m}^0 = x_t$ (client m initializes local training using the latest global model x_t)

Random-Reshuffling option: Sample a permutation $\pi_m = (\pi_m^0, \pi_m^1, \ldots, \pi_m^{n-1})$ of $\{1, 2, \ldots, n\}$ **for** *all local training data points $i = 0, 1, \ldots, n-1$* **do**

$x_{t,m}^{i+1} = x_{t,m}^i - \gamma \nabla f_m^{\pi_m^i}(x_{t,m}^i)$ (client m makes one pass over its local training data in the order dictated by π_m)

end

$g_{t,m} = \frac{1}{\gamma n}(x_t - x_{t,m}^n)$ (client m computes local update direction $g_{t,m}$)

end

$g_t = \frac{1}{C} \sum_{m \in S_t} g_{t,m}$ (server aggregates the local update directions $g_{t,m}$ discovered by the cohort S_t of clients)

$x_{t+1} = x_t - \eta g_t$ (server updates the model using the aggregated direction g_t and applying server stepsize η)

end

4. The Nastya Algorithm

We now formally describe our Nastya algorithm (see Algorithm 1). Nastya combines several techniques that were empirically found to be useful in FL: *partial participation*, *local training*, *data shuffling* and *server stepsizes*.

In each communication round $t \geq 0$ of Nastya, the cohort S_t is chosen as a random subset of the set $\{1, 2, \ldots, M\}$ of all clients. In particular, we choose a random subset of cardinality C (the cohort size), where $1 \leq C \leq M$, uniformly at random. The server then sends the global model x_t to all clients in the cohort. Setting $C = M$ models the full participation regime.

Each participating client $m \in S_t$ then performs local training using a single pass of incremental GD with *client stepsize* $\gamma > 0$ over the local training data points in an order dictated by a *random permutation*

$$\pi_m = (\pi_m^1, \pi_m^2, \ldots, \pi_m^n)$$

of the local training data set $\{1, 2, \ldots, n\}$. In particular, the following update is iterated for $i = 0, \ldots, n-1$:

$$x_{t,m}^{i+1} = x_{t,m}^i - \gamma \nabla f_m^{\pi_m^i}(x_{t,m}^i),$$

where $x_{t,m}^0$ is initialized to x_t, and $\gamma > 0$ is the client stepsize. That is, we run one pass over the local data using the RR method Mishchenko et al. [19]. This differs from one pass over the data via SGD in that each data point is sampled exactly once.

Note that we allow for two options for how the permutation is formed: i) either the random permutation is sampled just once for all clients, and used in all communication rounds (*Shuffle-Once* option), or ii) the random permutation is sampled afresh at the start of each communication round (*Random-Reshuffling* option). Both have the same theoretical properties in our analysis.

At the end of local training, the updated models $x_{t,m}^n$ are communicated back to the server, which uses these updates to form a *gradient-type estimator* g_t, and applies one step of GD using a server stepsize $\eta > 0$ with this estimator in lieu of the true gradient. Equivalently and this is how we decided to formally state the method, each client $m \in S_t$ sends the following scaled model difference to the server:

$$g_{t,m} = \frac{1}{\gamma n}(x_t - x_{t,m}^n),$$

where $x_{t,m}^n$ is the model found by the client after one pass over the data via RR. The server then aggregates these vectors from all clients in the cohort to form $g_t = \frac{1}{C}\sum_{m \in S_t} g_{t,m}$, and then takes a gradient-type step using this quantity in lieu of the gradient, using server stepsize $\eta > 0$:

$$x_{t+1} = x_t - \eta g_t.$$

The new model is then broadcast to a new cohort in the next communication round, and the process is repeated.

Table 3: Comparison of convergence results for FedAvg from prior work with our results.

Method	Strongly convex	Non-convex	Reference
SCAFFOLD [(1)]	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu M n\epsilon}+\frac{1}{\mu}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{Mn\epsilon^2}+\frac{1}{\epsilon}\right)$	[11]
Local SGD [(1)]	$\tilde{\mathcal{O}}\left(\frac{L}{\mu}+\frac{\sigma^2}{M\mu\varepsilon}+\sqrt{\frac{Ln\left(\sigma^2+n\zeta^2\right)}{\mu^2\varepsilon}}\right)$	✗	[27]
Local SGD	$\tilde{\mathcal{O}}\left(\frac{\sigma_*^2}{M\mu\epsilon}+\frac{\sqrt{L}(n\zeta+\sqrt{n}\sigma)}{\mu\sqrt{\epsilon}}+\kappa n\right)$	$\mathcal{O}\left(\frac{L\sigma_*^2}{M\epsilon^2}+\frac{L(n\zeta+\sqrt{n}\sigma)}{\epsilon^{3/2}}+\frac{Ln}{\epsilon}\right)$	[13]
FedRR	$\tilde{\mathcal{O}}\left(\frac{L}{\mu}+\frac{\sqrt{\kappa n}\left(\sigma_*+\sqrt{n}\zeta\right)}{\mu\sqrt{\varepsilon}}\right)$	✗	[20]
Nastya	$\tilde{\mathcal{O}}\left(\frac{Ln}{\mu}\right)$	$\mathcal{O}\left(\frac{Ln}{\varepsilon}\right)$	**This paper**

[(1)] The analysis is done under the bounded variance assumption: $g_i(x) := \nabla f_i\left(x;\zeta_i\right)$ is unbiased stochastic gradient of f_i with bounded variance $\mathbb{E}_{\zeta_i}\left[\|g_i(x)-\nabla f_i(x)\|^2\right] \le \sigma^2$, for any i, x.
Here we use $\zeta^2 \overset{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M}\|\nabla f_m(x_*)\|^2$.

5. Theory

We now formulate our three main results.

Theorem 6 (Strongly convex regime) *Let Assumption 1 hold, each f_m^i be convex and f be μ-strongly convex. Let $\gamma n \le \eta \le \frac{1}{16L}$. Then for iterates x_t generated by Algorithm 1, we have*

$$\mathbb{E}\|x_T-x_*\|^2 \le \left(1-\frac{\eta\mu}{2}\right)^T\|x_0-x_*\|^2+\frac{5\gamma^2 nL}{\mu}\left(\frac{1}{M}\sum_{m=1}^{M}\sigma_{*,m}^2+n\sigma_*^2\right)+\frac{8\eta}{\mu}\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2.$$

In the full participation regime, the server stepsize restriction can be relaxed to $\eta \le \frac{1}{8L}$.

5.1. Convex regime

Next, we we cover the convex regime.

Theorem 7 *Let Assumption 1 hold, each f_m^i be convex function. Let $\gamma n \le \eta \le \frac{1}{16L}$. Let $\hat{x}_T \overset{def}{=} \frac{1}{T}\sum_{t=1}^{T}x_t$. Then for iterates x_t of Algorithm 1, we have*

$$\mathbb{E}[f(\hat{x}_T)-f(x_*)] \le \frac{5\|x_0-x_*\|^2}{2\eta T}+7\gamma^2 nL\left(\frac{1}{M}\sum_{m=1}^{M}\sigma_{*,m}^2+n\sigma_*^2\right)+10\eta\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2.$$

As it can be seen, we get additional source of variance which is proportional to η and σ_*^2. This term means variance of client sampling. Since this sampling of clients have SGD-type structure, we have that variance is proportional to the first order of server-side stepsize.

5.2. Non-convex regime

Finally, we provide guarantees in the non-convex case.

Theorem 8 *Let Assumption of smoothness hold. Let $\delta_0 = f(x_0)-f_*$ and $\Delta_{*,m} = \frac{1}{n}\sum_{i=1}^{n}(f_*-f_{*,m}^i)$. Let $\gamma \le \frac{1}{2nL}$ and $\eta \le \frac{1}{L}$. Then for iterates x_t of Algorithm 1, we have*

$$\min_{t=0,\ldots,T-1}\mathbb{E}\|\nabla f(x_t)\|^2 \le \frac{2\left(1+4\eta\gamma^2 n^2L^3\right)^T}{\eta T}\delta_0+2\gamma^2 nL^3\left(\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m}+n\Delta_*\right)+4L^2\eta\frac{M-C}{C\max\{M-1,1\}}\Delta_*.$$

Similarly to analysis in full participation case, we use $\Delta_{*,m}$ and Δ_* instead of $\sigma_{*,m}^2$ and σ_*^2, since point of minimizer cannot be defined.

Client and server stepsizes. Theorems 6, 15 and 19 suggest that the server can use the large $\mathcal{O}(1/L)$ stepsize, where L is the Lipschitz constant of the gradient of f. In all regimes, it is optimal for the client stepsize γ to be small, which completely eliminates the second of the three terms in the complexity bounds, which controls the price one pays due to *data heterogeneity*.

Partial participation. Notice that if the cohort size is equal to M, then $\frac{M-C}{C\max\{1,M-1\}}$ is equal to 0, and this means that the last (third) term in all our complexity results disappears. The last term can thus be interpreted as the price we pay for partial participation. While we can reduce the variance of RR and the client drift by decreasing γ, we cannot make the variance due to client sampling arbitrary small, since it depends on η.

Comparison with existing rates. In Table 3 we compare our results in the strongly convex and non-convex regimes with selected existing results.

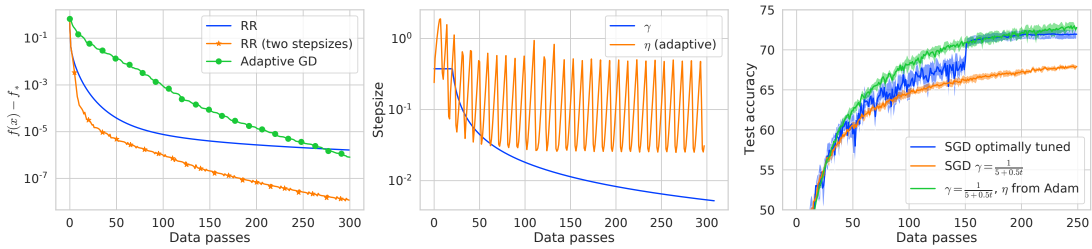


Figure 2: **Left and middle:** We compare running standard Random Reshuffling (RR), adaptive gradient descent (Adaptive GD), and the combination of RR with outer adaptive stepsize (Nastya) (RR (two stepsizes)) on logistic regression. As one can see, the variant with two stepsizes outperforms both of them and does not require more hyper-parameters than RR, and the middle plot shows the exact values of γ and η. **Right:** The right plot shows the training curves of LeNet on CIFAR-10 with minibatch size 1024, where we compare carefully tuned SGD (blue) to poorly tuned SGD (orange) and show that using Adam optimizer with stepsize 10^{-2} after each data pass can significantly improve the poorly tuned version.

6. Benefits of Small Server Stepsize

Our analysis shows that small client stepsizes can control variance. However, the goal of learning is not obtaining the best value of the loss function, but the performance of the model. In recent papers it was shown that small stepsizes are not the best option in term of generalization [24]. Moreover, in some cases we can show that using small server stepsize and large client stepsizes can be beneficial.

Theorem 9 *Assume that all losses $f_{m,i}$ are L-smooth and μ-strongly convex. Define $\alpha = \frac{\eta}{\gamma n}$. Let $\gamma \leq \frac{1}{L}$ and $0 \leq \alpha < 1$. Then for iterates x_t generated by Algorithm 1 we have*

$$\mathbb{E}\left\|x_T - x_*\right\|^2 \leq (1-\alpha+\alpha(1-\gamma\mu)^n)^T \left\|x_0 - x_*\right\|^2 + \frac{\alpha}{(1-\alpha)(1-(1-\gamma\mu)^n)}\gamma^2\frac{\sigma_*^2}{C} + 2\gamma^3\sigma_{\mathrm{rad}}^2\frac{1}{1-(1-\gamma\mu)^n}\sum_{i=0}^{n-1}(1-\gamma\mu)^i.$$

The upper bound depends on α in a nonlinear way, so the optimal value of α would often lie somewhere in the interval $(0,1)$. Furthermore, the last term does not change with α, so the optimal value α^* of α is completely determined by the first two terms.

Let us derive optimal α^* under some approximations. In particular, when for ill-conditioned problems where μ is sufficiently small, it holds $(1-\gamma\mu)^n \approx 1-\gamma\mu n$. Ignoring the last term in the upper bound of Theorem 20, which does not affect the value α^*, and using $\frac{1}{1-\alpha} \leq 2$ for $\alpha \leq \frac{1}{2}$, we simplify the upper bound to

$$(1-\alpha+\alpha(1-\gamma\mu n))^T\left\|x_0 - x_*\right\|^2 + \frac{2\alpha\gamma^2}{1-(1-\gamma\mu n)}\frac{\sigma_*^2}{C} = (1-\alpha\gamma\mu n)^T\left\|x_0 - x_*\right\|^2 + \frac{2\alpha\gamma}{\mu n}\frac{\sigma_*^2}{C}.$$

To have this upper bound smaller than some $\varepsilon \geq 0$, we need to use $\alpha = \mathcal{O}\left(\frac{n\varepsilon}{\gamma}\right)$ and $T = \mathcal{O}(\frac{1}{\alpha\gamma\mu n}\log\frac{1}{\varepsilon})$, where we ignore constants unrelated to $\alpha, \gamma, \varepsilon, \mu$ and n. Thus, the larger local γ, the smaller global α.

7. Experiments

To showcase the speed-up that can be obtained from the server-side stepsizes, we run a toy experiment in the single-node setup, i.e., we consider standard minimization of a finite-sum. We combine the local passes over the data with the adaptive estimation of smoothness proposed by [17]. We run our experiment on ℓ_2-regularized logistic regression with the 'mushrooms' dataset from LibSVM [4]. The results are reported in Figure 2.

We use standard LeNet architecture, which is a 5-layer convolutional neural network, implemented in PyTorch [21] and train them to classify images from the CIFAR-10 dataset [16] with cross-entropy loss. At each iteration, we use a minibatch of size 1024. For the tuned SGD, we start with stepsize 0.2 and divide by 10 at epochs 150 and 200. For the other version, we take SGD with stepsize 0.2 and decrease as $\mathcal{O}(\frac{1}{t})$, where t is the epoch number.

For our method, we treat the full sum of gradients over epoch as an approximation of full gradient and use Adam with stepsize 0.01 to improve this update. We can see that by applying Adam, we can improve the performance of SGD with decreasing stepsize. At the same time, applying it to the tuned stepsize schedule only made the results much worse, so we do not report that line. This highlights that adaptive outer stepsizes are helpful when the base stepsize γ is not chosen well, which is in line with our theory.

References

[1] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Sarit Khirirat, Nikola Konstantinov, and Cédric Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, 2018.

[2] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[3] Léon Bottou. Curiously fast convergence of some stochastic gradient descent algorithms. Unpublished open problem offered to the attendance of the SLDS 2009 conference, 2009.

[4] Chih-Chung Chang and Chih-Jen Lin. LibSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.

[5] Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. *arXiv preprint arXiv:2106.07820*, 2021.

[6] Wenlin Chen, Samuel Horvath, and Peter Richtárik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020.

[7] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. Local SGD: unified theory and new efficient methods. In *NeurIPS*, 2020.

[8] Eduard Gorbunov, Konstantin Burlachenko, Zhize Li, and Peter Richtárik. Marina: Faster non-convex distributed learning with compression. 139:3788–3798, 18–24 Jul 2021.

[9] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[10] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2021.

[11] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[12] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[13] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR, 2020.

[14] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[15] Jakub Konečný, H. Brendan McMahan, Felix Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[17] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 6702–6712. PMLR, 2020.

[18] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[19] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random Reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33:17309–17320, 2020.

[20] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Proximal and federated random reshuffling. *arXiv preprint arXiv:2102.06704*, 2021.

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[22] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2020.

[23] Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback. *arXiv preprint arXiv:2106.05203*, 2021.

[24] Samuel L. Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2020.

[25] Sebastian U. Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

[26] Ruo-Yu Sun. Optimization for deep learning: An overview. *Journal of the Operations Research Society of China*, 8:1–46, 06 2020. doi: 10.1007/s40305-020-00309-6.

[27] Blake E. Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local SGD for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 6281–6292. Curran Associates, Inc., 2020.

[28] Chulhee Yun, Shashank Rajput, and Suvrit Sra. Minibatch vs local SGD with shuffling: Tight convergence bounds and beyond. *arXiv preprint arXiv:2110.10342*, 2021.

Appendix A. Basic Facts and Notation

A.1. Basic facts

For any two vectors $a, b \in \mathbb{R}^d$ and any $\zeta > 0$,

$$2a, b \leq \frac{a}{\zeta} + \zeta b. \tag{7}$$

A consequence of (7) is that for any $a, b \in \mathbb{R}^d$, we have

$$a + b \leq 1 + \zeta a + 1 + \zeta^{-1} b. \tag{8}$$

Using $\zeta = 1$ specifically yields,

$$a + b \leq 2a + 2b. \tag{9}$$

A function $h : \mathbb{R}^d \to \mathbb{R}$ is called μ-convex if for some $\mu \geq 0$ and for all $x, y \in \mathbb{R}^d$, we have

$$h(x) + \nabla h(x), y - x + \frac{\mu}{2} y - x \leq h(y). \tag{10}$$

Function $h : \mathbb{R}^d \to \mathbb{R}$ is called L-smooth if for some $L \geq 0$ and for all $x, y \in \mathbb{R}^d$, we have

$$\|\nabla h(x) - \nabla h(y)\| \leq L \|x - y\| . \tag{11}$$

A useful consequence of L-smoothness is the inequality

$$h(x) \leq h(y) + \nabla h(y), x - y + \frac{L}{2} x - y, \tag{12}$$

holding for all $x, y \in \mathbb{R}^d$. If h is L-smooth and lower bounded by h_*, then

$$\nabla h(x) \leq 2L h(x) - h_*. \tag{13}$$

For any convex and L-smooth function h it holds

$$\|\nabla h(x) - \nabla h(y)\|^2 \leq 2L D_h(x, y). \tag{14}$$

For a convex function $h \colon \mathbb{R}^d \to \mathbb{R}$ and any vectors $y_1, \ldots, y_n \in \mathbb{R}^d$, Jensen's inequality states that

$$h \frac{1}{n} \sum_{i=1}^{n} y_i \leq \frac{1}{n} \sum_{i=1}^{n} h(y_i). \tag{15}$$

Applying this to the squared norm, $h(y) = y$, we get

$$\frac{1}{n} \sum_{i=1}^{n} y_i \leq \frac{1}{n} \sum_{i=1}^{n} y_i. \tag{16}$$

Simple multiplication on both sides of (16) also yields,

$$\sum_{i=1}^{n} y_i \leq n \sum_{i=1}^{n} y_i. \tag{17}$$

We use the following decomposition that holds for any random variable X with $\mathbb{E}\left[\|X\|^2\right] < +\infty$,

$$\mathbb{E}\left[\|X\|^2\right] = \|\mathbb{E}[X]\|^2 + \mathbb{E}\left[\|X - \mathbb{E}[X]\|^2\right]. \tag{18}$$

We will make use of the particularization of (18) to the discrete case: let $y_1, \ldots, y_m \in \mathbb{R}^d$ be given vectors and let $\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y_i$ be their average. Then,

$$\frac{1}{m} \sum_{i=1}^{m} y_i = \bar{y} + \frac{1}{m} \sum_{i=1}^{m} y_i - \bar{y}. \tag{19}$$

Table 4: Summary of notation used.

Symbol	Description
x_t	The iterate used at the start of epoch t.
π	A permutation $\pi_m = \pi_m^0, \pi_m^1, \ldots, \pi_m^{n-1}$ of $\{1, 2, \ldots, n\}$, which is resampled every epoch for Random Reshuffling.
γ	The stepsize used when taking descent steps in an epoch.
$x_{t,m}^i$	The current iterate after i steps in epoch t, for $0 \leq i \leq n$.
g_t	The sum of gradients used over epoch t such that $x_{t+1} = x_t - \eta g_t$.
β	The epoch jumping parameter.
η	The effective epoch stepsize, defined as $\eta \overset{\text{def}}{=} \gamma 1 + \beta$.
σ_t^2	The variance of the individual loss gradients from the average loss at point x_t.
L	The smoothness constant of f and $f_1, f_2, \ldots, f_n$.
δ_t	Functional suboptimality, $\delta_t = f(x_t) - f_*$, where $f_* = \inf_x f(x)$.

A.2. Notation

We define the variance of the local gradients from their average at a point x_t as

$$\sigma_t^2 \overset{\text{def}}{=} \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(x_t) - \nabla f(x_t).$$

A summary of the notation used is given in Table 4.

A.3. Sampling without replacement

First, let us provide some basic properties of sampling without replacement.

Lemma 10 *Let $X_1, \ldots, X_n \in \mathbb{R}^d$ be fixed vectors, $\overline{X} \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} X_i$ be their average and $\sigma^2 \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} \left\| X_i - \overline{X} \right\|^2$ be the population variance. Fix any $k \in \{1, \ldots, n\}$, let $X_{\pi_1}, \ldots X_{\pi_k}$ be sampled uniformly without replacement from $\{X_1, \ldots, X_n\}$ and $\overline{X}_\pi$ be their average. Then, it holds*

$$\mathbb{E}\left[\overline{X}_\pi\right] = \overline{X}, \qquad \mathbb{E}\left[\left\|\overline{X}_\pi - \overline{X}\right\|^2\right] = \frac{n-k}{k(n-1)} \sigma^2. \tag{20}$$

Proof The first claim follows by linearity of the expectation and uniformity of the sampling,

$$\mathbb{E}\left[\overline{X}_\pi\right] = \frac{1}{k} \sum_{i=1}^{k} \mathbb{E}\left[X_{\pi_i}\right] = \frac{1}{k} \sum_{i=1}^{k} \overline{X} = \overline{X}.$$

To show the second claim, let us first establish that for any $i \neq j$ it holds $\mathrm{cov}(X_{\pi_i}, X_{\pi_j}) = -\frac{\sigma^2}{n-1}$. Indeed,

$$\begin{aligned}
\mathrm{cov}(X_{\pi_i}, X_{\pi_j}) &= \mathbb{E}\left[X_{\pi_i} - \overline{X}, X_{\pi_j} - \overline{X}\right] = \frac{1}{n(n-1)} \sum_{l=1}^{n} \sum_{m \neq l} X_l - \overline{X}, X_m - \overline{X} \\
&= \frac{1}{n(n-1)} \sum_{l=1}^{n} \sum_{m=1}^{n} X_l - \overline{X}, X_m - \overline{X} - \frac{1}{n(n-1)} \sum_{l=1}^{n} \left\|X_l - \overline{X}\right\|^2 \\
&= \frac{1}{n(n-1)} \sum_{l=1}^{n} X_l - \overline{X}, \sum_{m=1}^{n} (X_m - \overline{X}) - \frac{\sigma^2}{n-1} \\
&= -\frac{\sigma^2}{n-1}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\overline{X}_\pi - \overline{X} &= \frac{1}{k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} \mathrm{cov}(X_{\pi_i}, X_{\pi_j}) \\
&= \frac{1}{k^2} \mathbb{E}\left[\sum_{i=1}^{k} X_{\pi_i} - \overline{X}\right] + \sum_{i=1}^{k} \sum_{j=1, j\neq i}^{n} \mathrm{cov}(X_{\pi_i}, X_{\pi_j}) \\
&= \frac{1}{k^2}\left(k\sigma^2 - k(k-1)\frac{\sigma^2}{n-1}\right) = \frac{n-k}{k(n-1)}\sigma^2.
\end{aligned}$$

■

Appendix B. Large Server Stepsize

B.1. Strongly convex and general convex case

Lemma 11 *Let Assumption 1 holds and further assume f is μ-strongly convex and each f_m^i is convex. Then*

$$-\frac{1}{Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} \left\langle f_m^{\pi_m^i}\left(x_{t,m}^i\right), x_t - x_* \right\rangle \leq -\frac{\mu}{4}\|x_t - x_*\|^2 - \frac{1}{2}\left(f\left(x_t\right) - f\left(x_*\right)\right) + \frac{L}{2Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} \left\|x_t - x_{t,m}^i\right\|^2.$$

Proof We start with the inner product and decompose it using the three-point identity:

$$\begin{aligned}
\left\langle \nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right), x_t - x_* \right\rangle &= f_m^{\pi_m^i}\left(x_t\right) - f_m^{\pi_m^i}\left(x_*\right) + f_m^{\pi_m^i}\left(x_*\right) - f_m^{\pi_m^i}\left(x_{t,m}^i\right) \\
&\quad + \left\langle \nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right), x_{t,m}^i - x_* \right\rangle - f_m^{\pi_m^i}\left(x_t\right) + f_m^{\pi_m^i}\left(x_{t,m}^i\right) \\
&\quad + \left\langle \nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right), x_t - x_{t,m}^i \right\rangle \\
&= f_m^{\pi_m^i}\left(x_t\right) - f_m^{\pi_m^i}\left(x_*\right) + D_{f_m^{\pi_m^i}}\left(x_*, x_{t,m}^i\right) - D_{f_m^{\pi_m^i}}\left(x_t, x_{t,m}^i\right). \qquad (21)
\end{aligned}$$

Using the representation (21), L-smoothness and μ-strong convexity we have a bound:

$$\begin{aligned}
& -\frac{1}{Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} \left\langle f_m^{\pi_m^i}\left(x_{t,m}^i\right), x_t - x_* \right\rangle \\
& \leq -\frac{1}{Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} \left(f_m^{\pi_m^i}\left(x_t\right) - f_m^{\pi_m^i}\left(x_*\right) + D_{f_m^{\pi_m^i}}\left(x_*, x_{t,m}^i\right) - D_{f_m^{\pi_m^i}}\left(x_t, x_{t,m}^i\right) \right) \\
& \overset{(12)}{\leq} -\left(f\left(x_t\right) - f\left(x_*\right)\right) - \frac{1}{Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} D_{f_{m,\pi_i}}\left(x_*, x_{t,m}^i\right) + \frac{L}{2Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} \left\|x_t - x_{t,m}^i\right\|^2 \\
& \overset{(10)}{\leq} -\frac{\mu}{4}\|x_t - x_*\|^2 - \frac{1}{2}\left(f\left(x_t\right) - f\left(x_*\right)\right) + \frac{L}{2Mn} \sum_{m=1}^{M} \sum_{i=0}^{n-1} \left\|x_t - x_{t,m}^i\right\|^2.
\end{aligned}$$

■

Lemma 12 *Assume that Assumption 1 holds, then*

$$\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 \leq 2\frac{L^2}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 4\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_*)\right\|^2 + 8L(f_m(x_t) - f_m(x_*)).$$

Proof We start with Young's inequality:

$$\begin{aligned}\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 &\overset{(9)}{\leq} 2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\left(\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right) - \nabla f_m^{\pi_m^i}(x_t)\right)\right\|^2 + 2\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_t)\right\|^2 \\ &\overset{(14)}{\leq} 2L^2\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 2\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_t)\right\|^2.\end{aligned}$$

We use Young's inequality and L-smoothness again:

$$\begin{aligned}\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 &\overset{(15),(9)}{\leq} 2L^2\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 4\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_*)\right\|^2 \\ &\quad + 4\frac{1}{C}\sum_{m\in S_t}\|\nabla f_m(x_t) - \nabla f_m(x_*)\|^2 \\ &\overset{(14)}{\leq} 2L^2\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\|x_{m,t}^i - x_t\|^2 + 4\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_*)\right\|^2 \\ &\quad + 8L\frac{1}{C}\sum_{m\in S_t}(f_m(x_t) - f_m(x_*)).\end{aligned}$$

■

Lemma 13 *Suppose that Algorithm 1 is used and Assumption 1 holds. If $\gamma \leq \frac{1}{2Ln}$, then*

$$\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_t - x_{t,m}^i\right\|^2 |x_t\right] \leq 8\gamma^2 n^2 L\left(f(x_t) - f(x_*)\right) + 2\gamma^2 n^2\frac{1}{M}\sum_{m=1}^{M}\|\nabla f_m(x_*)\|^2 + 2\gamma^2 n\frac{1}{M}\sum_{m=1}^{M}\sigma_{*,m}^2.$$

Proof We start from the definition of $x_{t,m}^i$:

$$\begin{aligned}\mathbb{E}\left[\left\|x_{t,m}^i - x_t\right\|^2 |x_t\right] &= \mathbb{E}\left[\left\|\gamma\sum_{j=0}^{i-1}\nabla f_m^{\pi_m^j}\left(x_{t,m}^j\right)\right\|^2 |x_t\right] \\ &\overset{(9)}{\leq} 2\gamma^2\mathbb{E}\left[\left\|\sum_{j=0}^{i-1}\left(\nabla f_m^{\pi_m^j}\left(x_{t,m}^j\right) - \nabla f_m^{\pi_m^j}(x_t)\right)\right\|^2 |x_t\right] + 2\gamma^2\mathbb{E}\left[\left\|\sum_{j=0}^{i-1}\nabla f_m^{\pi_m^j}(x_t)\right\|^2 |x_t\right] \\ &\overset{(15)}{\leq} 2\gamma^2 i\sum_{j=0}^{i-1}\mathbb{E}\left[\left\|\nabla f_m^{\pi_m^j}\left(x_{t,m}^j\right) - \nabla f_m^{\pi_m^j}(x_t)\right\|^2 |x_t\right] + 2\gamma^2\mathbb{E}\left[\left\|\sum_{j=0}^{i-1}\nabla f_m^{\pi_m^j}(x_t)\right\|^2 |x_t\right] \\ &\overset{(11)}{\leq} 2\gamma^2 L^2 i\sum_{j=0}^{i-1}\mathbb{E}\left[\left\|x_{t,m}^j - x_t\right\|^2 |x_t\right] + 2\gamma^2\mathbb{E}\left[\left\|\sum_{j=0}^{i-1}\nabla f_m^{\pi_m^j}(x_t)\right\|^2 |x_t\right].\end{aligned}$$

Now let us look at the last term. We can apply Lemma 10 and get

$$\begin{aligned}\mathbb{E}\left[\left\|\sum_{j=0}^{i-1}\nabla f_m^{\pi_m^j}(x_t)\right\|^2|x_t\right] &= i^2\left\|\nabla f_m(x_t)\right\|^2 + i^2\mathbb{E}\left[\left\|\frac{1}{i}\sum_{j=0}^{i-1}\left(\nabla f_m^{\pi_m^j}(x_t)-\nabla f_m(x_t)\right)\right\|^2|x_t\right]\\ &= i^2\left\|\nabla f_m(x_t)\right\|^2 + \frac{i(n-i)}{n-1}\sigma_{t,m}^2,\end{aligned}$$

where $\sigma_{t,m}^2 \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n\left\|\nabla f_m^i(x_t)-\nabla f_m(x_t)\right\|^2$.
Let us go back:

$$\mathbb{E}\left[\left\|x_{t,m}^i-x_t\right\|^2|x_t\right] \le 2\gamma^2L^2 i\sum_{j=0}^{i-1}\mathbb{E}\left[\left\|x_{t,m}^j-x_t\right\|^2|x_t\right] + 2\gamma^2\left(i^2\left\|\nabla f_m(x_t)\right\|^2+\frac{i(n-i)}{n-1}\sigma_{t,m}^2\right).$$

Summing the terms leads to

$$\begin{aligned}\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i-x_t\right\|^2|x_t\right] &\le 2\gamma^2L^2\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}i\sum_{j=0}^{i-1}\mathbb{E}\left[\left\|x_{t,m}^j-x_t\right\|^2|x_t\right]\\ &\quad+\frac{2\gamma^2}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}i^2\left\|\nabla f_m(x_t)\right\|^2+\frac{2\gamma^2}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\frac{i(n-i)\sigma_{t,m}^2}{n-1}\\ &\le 2\gamma^2L^2\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i-x_t\right\|^2|x_t\right]\cdot\frac{n(n-1)}{2}\\ &\quad+\frac{2\gamma^2}{Mn}\sum_{m=1}^M\left\|\nabla f_m(x_t)\right\|^2\cdot\frac{n(n-1)(2n-1)}{6}+\frac{\gamma^2n(n+1)}{3}\frac{1}{Mn}\sum_{m=1}^M\sigma_{t,m}^2.\end{aligned}$$

Choosing $\gamma \le \frac{1}{2Ln}$, we verify

$$\begin{aligned}\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i-x_t\right\|^2|x_t\right] &\le \frac{4}{3}\left(1-\gamma^2L^2n(n-1)\right)\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i-x_t\right\|^2|x_t\right]\\ &\le \frac{4\gamma^2}{9}\frac{1}{M}\sum_{m=1}^M\left\|\nabla f_m(x_t)\right\|^2\cdot(n-1)(2n-1)+\frac{4\gamma^2(n+1)}{9}\frac{1}{M}\sum_{m=1}^M\sigma_{t,m}^2\\ &\le \gamma^2n^2\frac{1}{M}\sum_{m=1}^M\left\|\nabla f_m(x_t)\right\|^2+\gamma^2n\frac{1}{M}\sum_{m=1}^M\sigma_{t,m}^2.\end{aligned} \tag{22}$$

Using Young's inequality, we get

$$\begin{aligned}\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i-x_t\right\|^2|x_t\right] &\stackrel{(9),(19)}{\le} 2\gamma^2n^2\frac{1}{M}\sum_{m=1}^M\left\|\nabla f_m(x_t)-\nabla f_m(x_*)\right\|^2\\ &\quad+2\gamma^2n^2\frac{1}{M}\sum_{m=1}^M\left\|\nabla f_m(x_*)\right\|^2-\frac{\gamma^2n}{M}\sum_{m=1}^M\left\|\nabla f_m(x_t)\right\|^2\\ &\quad+2\gamma^2n\frac{1}{M}\sum_{m=1}^M\frac{1}{n}\sum_{i=0}^{n-1}\mathbb{E}\left\|\nabla f_m^{\pi_m^i}(x_t)-\nabla f_m^{\pi_m^i}(x_*)\right\|^2\\ &\quad+2\gamma^2n\frac{1}{M}\sum_{m=1}^M\frac{1}{n}\sum_{i=0}^{n-1}\mathbb{E}\left\|\nabla f_m^{\pi_m^i}(x_*)\right\|^2.\end{aligned}$$

Using L-smoothness, we obtain

$$\begin{aligned}\frac{1}{Mn}\sum_{m=1}^M\sum_{i=0}^{n-1}\mathbb{E}\left\|x_{t,m}^i-x_t\right\|^2 &\le 4\gamma^2n^2L\frac{1}{M}\sum_{m=1}^M D_{f_m}(x_t,x_*)+2\gamma^2n\frac{1}{M}\sum_{m=1}^M\sigma_{*,m}^2\\ &\quad+4\gamma^2nL\frac{1}{M}\sum_{m=1}^M\frac{1}{n}\sum_{i=0}^{n-1}D_{f_m^{\pi_m^i}}(x_t,x_*)+2\gamma^2n^2\frac{1}{M}\sum_{m=1}^M\left\|\nabla f_m(x_*)\right\|^2\\ &\stackrel{(12)}{\le} 8\gamma^2n^2L\left(f(x_t)-f(x_*)\right)+2\gamma^2n^2\frac{1}{M}\sum_{m=1}^M\left\|\nabla f_m(x_*)\right\|^2+2\gamma^2n\frac{1}{M}\sum_{m=1}^M\sigma_{*,m}^2.\end{aligned}$$

■

Theorem 14 *Assume that Assumption 1 holds and f is μ-strongly convex function. Let $\gamma n \leq \eta \leq \frac{1}{16L}$. Then for iterates x_t generated by Algorithm 1 we have*

$$\mathbb{E}\|x_T - x_*\|^2 \leq \left(1 - \frac{\eta\mu}{2}\right)^T \|x_0 - x_*\|^2 + \frac{5\gamma^2 nL}{\mu}\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2 + n\|\nabla f_m(x_*)\|^2\right) + \frac{8\eta}{\mu}\sum_{m=1}^{M}\|\nabla f_m(x_*)\|^2.$$

Proof We start from conditional expected value over distribution of client sampling,

$$\begin{aligned}\|x_{t+1} - x_*\|^2 &= \left\| x_t - \eta\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right) - x_*\right\|^2 \\ &= \|x_t - x_*\|^2 - 2\eta\left\langle \frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_{m,\pi_i}\left(x_{t,m}^i\right), x_t - x_*\right\rangle + \eta^2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_{m,\pi_i}\left(x_{t,m}^i\right)\right\|^2.\end{aligned}$$

Using Lemma 12, we get

$$\begin{aligned}\|x_{t+1} - x_*\|^2 &\leq \|x_t - x_*\|^2 - 2\eta\left\langle \frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_{m,\pi_i}\left(x_{t,m}^i\right), x_t - x_*\right\rangle \\ &+ \eta^2\left(2L^2\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\|x_{m,t}^i - x_t\|^2 + 4\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_*)\right\|^2 + 8L\frac{1}{C}\sum_{m\in S_t}\left(f_m(x_t) - f_m(x_*)\right)\right).\end{aligned}$$

Taking conditional expectation over sampling S_t, we get

$$\begin{aligned}\mathbb{E}_{S_t}\|x_{t+1} - x_*\|^2 &\leq \|x_t - x_*\|^2 - \mathbb{E}_{S_t}2\eta\left\langle \frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_{m,\pi_i}\left(x_{t,m}^i\right), x_t - x_*\right\rangle \\ &+ \eta^2\mathbb{E}_{S_t}\left(2L^2\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 4\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_*)\right\|^2 + 8L\frac{1}{C}\sum_{m\in S_t}\left(f_m(x_t) - f_m(x_*)\right)\right) \\ &\leq \|x_t - x_*\|^2 - 2\eta\left\langle \frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\nabla f_{m,\pi_i}\left(x_{t,m}^i\right), x_t - x_*\right\rangle \\ &+ \eta^2\left(2L^2\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 4\mathbb{E}_{S_t}\left\|\frac{1}{C}\sum_{m\in S_t}\nabla f_m(x_*)\right\|^2 + 8L(f(x_t) - f(x_*))\right) \\ &\overset{(10)}{\leq} \|x_t - x_*\|^2 - 2\eta\left\langle \frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\nabla f_{m,\pi_i}\left(x_{t,m}^i\right), x_t - x_*\right\rangle \\ &+ \eta^2\left(2L^2\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 4\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2 + 8L(f(x_t) - f(x_*))\right).\end{aligned}$$

Using Lemma 11, we obtain

$$\begin{aligned}\mathbb{E}_{S_t}\|x_{t+1} - x_*\|^2 &\leq \|x_t - x_*\|^2 - 2\eta\left(-\frac{\mu}{4}\|x_t - x_*\|^2 - \frac{1}{2}\left(f(x_t) - f(x_*)\right) + \frac{L}{2Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\|x_t - x_{t,m}^i\|^2\right) \\ &+ \eta^2\left(2L^2\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\|x_{t,m}^i - x_t\|^2 + 4\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2 + 8L(f(x_t) - f(x_*))\right).\end{aligned}$$

Rearranging the terms, we obtain:

$$\begin{aligned}\mathbb{E}_{S_t}\|x_{t+1} - x_*\|^2 &= \|x_t - x_*\|^2\left(1 - \frac{\eta\mu}{2}\right) - \eta\left(1 - 8\eta L\right)\left(f(x_t) - f(x_*)\right) \\ &+ \eta L\left(1 + 2\eta L\right)\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\|x_{m,t}^i - x_t\|^2 + 4\eta^2\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2.\end{aligned}$$

Using the tower property of conditional expectation and Lemma 13, we get

$$\begin{aligned}\mathbb{E}\left[\|x_{t+1}-x_*\|^2|x_t\right] &= \|x_t-x_*\|^2\left(1-\frac{\eta\mu}{2}\right)+4\eta^2\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2\\ &\quad-\eta\left(1-8\eta L-(1+2\eta L)\,8\gamma^2n^2L^2\right)(f(x_t)-f(x_*))\\ &\quad+2\eta\,(1+2\eta L)\,\gamma^2nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right).\end{aligned} \tag{23}$$

Taking $\gamma\le\frac{1}{16nL}$ and $\eta\le\frac{1}{16L}$, we derive

$$\eta\left(1-8\eta L-(1+2\eta L)\,8\gamma^2n^2L^2\right)(f(x_t)-f(x_*))\ge 0.$$

Taking full expectation yields

$$\mathbb{E}\left[\|x_{t+1}-x_*\|^2\right]\le\mathbb{E}\,\|x_t-x_*\|^2\left(1-\frac{\eta\mu}{2}\right)+\frac{5}{2}\eta\gamma^2nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right)+4\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2.$$

Unrolling this recursion, we have

$$\mathbb{E}\|x_T-x_*\|^2\le\left(1-\frac{\eta\mu}{2}\right)^T\|x_0-x_*\|^2+\frac{5\gamma^2nL}{\mu}\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right)+\frac{8\eta}{\mu}\sum_{m=1}^{M}\|\nabla f_m(x_*)\|^2.$$

■

Theorem 15 *Let Assumption 1 hold, each f_m^i be convex function. Let $\gamma n\le\eta\le\frac{1}{16L}$. Let $\hat{x}_T\overset{def}{=}\frac{1}{T}\sum_{t=1}^{T}x_t$. Then for iterates x_t of Algorithm 1, we have*

$$\mathbb{E}[f(\hat{x}_T)-f(x_*)]\le\frac{5\,\|x_0-x_*\|^2}{2\eta T}+7\gamma^2nL\left(\frac{1}{M}\sum_{m=1}^{M}\sigma_{*,m}^2+n\sigma_*^2\right)+10\eta\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2.$$

B.2. General convex case

Proof We start from equation (23) with $\mu=0$:

$$\begin{aligned}\mathbb{E}\left[\|x_{t+1}-x_*\|^2|x_t\right] &= \|x_t-x_*\|^2+4\eta^2\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2\\ &\quad-\eta\left(1-8\eta L-(1+2\eta L)\,8\gamma^2n^2L^2\right)(f(x_t)-f(x_*))\\ &\quad+2\eta\,(1+2\eta L)\,\gamma^2nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right).\end{aligned}$$

Using $\gamma n\le\eta\le\frac{1}{16L}$, we obtain $-\left(1-8\eta L-(1+2\eta L)\,8\gamma^2n^2L^2\right)\le-\frac{4}{10}$

$$\begin{aligned}\mathbb{E}\left[\|x_{t+1}-x_*\|^2|x_t\right] &= \|x_t-x_*\|^2+4\eta^2\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2-\frac{4\eta}{10}(f(x_t)-f(x_*))\\ &\quad+\frac{5}{2}\eta\gamma^2nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right).\end{aligned}$$

Taking full expectation, we get

$$\begin{aligned}\mathbb{E}\|x_{t+1}-x_*\|^2 &= \mathbb{E}\,\|x_t-x_*\|^2+4\eta^2\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2-\frac{4\eta}{10}\mathbb{E}\,(f(x_t)-f(x_*))\\ &\quad+\frac{5}{2}\eta\gamma^2nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right).\end{aligned}$$

Rearranging the terms leads us to

$$\frac{4\eta}{10}\mathbb{E}\left(f(x_t)-f(x_*)\right)=\mathbb{E}\left\|x_t-x_*\right\|^2-\mathbb{E}\|x_{t+1}-x_*\|^2+4\eta^2\frac{M-C}{C\max\left\{M-1,1\right\}}\sigma_*^2 + \frac{5}{2}\eta\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right).$$

Averaging from 0 to $T-1$, we get

$$\begin{aligned}\frac{4\eta}{10}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left(f(x_t)-f(x_*)\right)&=\frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbb{E}\left\|x_t-x_*\right\|^2-\mathbb{E}\|x_{t+1}-x_*\|^2\right)+4\eta^2\frac{M-C}{C\max\left\{M-1,1\right\}}\sigma_*^2\\&+\frac{5}{2}\eta\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right)\\&=\frac{1}{T}\left(\mathbb{E}\left\|x_0-x_*\right\|^2-\mathbb{E}\|x_T-x_*\|^2\right)+4\eta^2\frac{M-C}{C\max\left\{M-1,1\right\}}\sigma_*^2\\&+\frac{5}{2}\eta\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{*,m}^2+n\|\nabla f_m(x_*)\|^2\right).\end{aligned}$$

Using Jensen inequality (15), we have

$$\mathbb{E}[f(\hat{x}_T)-f(x_*)]\leq\frac{5\left\|x_0-x_*\right\|^2}{2\eta T}+7\gamma^2 nL\left(\frac{1}{M}\sum_{m=1}^{M}\sigma_{*,m}^2+n\sigma_*^2\right)+10\eta\frac{M-C}{C\max\{M-1,1\}}\sigma_*^2.$$

■

B.3. General non-convex case

Finally, we provide guarantees in the non-convex case.

Lemma 16 *Assume that Assumption 1. For uniform sampling of cohort S_t we have*

$$\mathbb{E}_{S_t}\eta^2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2\leq 2L(f(x)-f_*)+2\frac{M-C}{C(M-1)}\Delta_*.$$

Proof We start with Proposition 3 in **?**]:

$$\mathbb{E}\left[\|g\|^2\right]\leq 2A\left(f(x)-f_*\right)+B\cdot\|\nabla f(x)\|^2+C,$$

where $A=\frac{M-C}{C(M-1)}L$, $B=\frac{M(C-1)}{C(M-1)}$, $C=2A\Delta_*$ and

$$g=\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right). \tag{24}$$

Using L-smoothness, we have

$$\|\nabla f(x)\|^2\leq 2L\left(f(x)-f_*\right).$$

Note that

$$\frac{M-C}{C(M-1)}+\frac{M(C-1)}{C(M-1)}=1.$$

Combining all things together, we obtain

$$\mathbb{E}_{S_t}\eta^2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2\leq 2L(f(x)-f_*)+2\frac{M-C}{C(M-1)}\Delta_*.$$

■

Lemma 17 *Suppose that Algorithm 1 is used and Assumption 1 holds. If $\gamma \le \frac{1}{2Ln}$, then*

$$\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_t - x_{t,m}^i\right\|^2 |x_t\right] \le 4\gamma^2 n^2 L\left(f(x_t) - f_*\right) + 2\gamma^2 n^2 L\Delta_* + 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m}.$$

Proof We start from equation (22):

$$\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i - x_t\right\|^2 |x_t\right] \le \gamma^2 n^2 \frac{1}{M}\sum_{m=1}^{M}\|\nabla f_m(x_t)\|^2 + \gamma^2 n\frac{1}{M}\sum_{m=1}^{M}\sigma_{t,m}^2.$$

Using L-smoothness, we get

$$\begin{aligned}\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i - x_t\right\|^2 |x_t\right] &\le 2\gamma^2 n^2 L\frac{1}{M}\sum_{m=1}^{M}(f_m(x_t) - f_{*,m}) + 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\frac{1}{n}\sum_{i=0}^{n-1}(f_m^i(x_t) - f_{*,m}^i)\\ &\le 2\gamma^2 n^2 L\frac{1}{M}\sum_{m=1}^{M}(f_m(x_t) - f_*) + 2\gamma^2 n^2 L\frac{1}{M}\sum_{m=1}^{M}(f_* - f_{*,m})\\ &+ 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\frac{1}{n}\sum_{i=0}^{n-1}(f_m^i(x_t) - f_*) + 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\frac{1}{n}\sum_{i=0}^{n-1}(f_* - f_{*,m}^i)\\ &\le 4L\gamma^2 n^2(f(x_t) - f_*) + 2\gamma^2 n^2 L\Delta_* + 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m}.\end{aligned}$$

■

Lemma 18 *Suppose that there exists constants $a, b, c \ge 0$ and nonnegative sequences $(s_t)_{t=0}^T, (q_t)_{t=0}^T$ such that for any $t \in \{0, 1, \dots, T\}$*

$$s_{t+1} \le 1 + as_t - bq_t + c. \tag{25}$$

Then if $a > 0$ we have,

$$\min_{t=0,\dots,T-1} q_t \le \frac{1 + a^T}{bT}s_0 + \frac{c}{b}. \tag{26}$$

And if $a = 0$ we have,

$$\frac{1}{T}\sum_{t=0}^{T-1} q_t \le \frac{s_0}{bT} + \frac{c}{b}. \tag{27}$$

Proof The first part of the proof (for $a > 0$) is a distillation of the recursion solution in Lemma 2 of [?] and we closely follow their proof. Let $w_{-1} = w_0 > 0$ be arbitrary. Define

$$w_t \overset{\text{def}}{=} \frac{w_0}{1 + a^t}.$$

Note that $w_t 1 + a = w_{t-1}$. Multiplying both sides of (25) by w_t,

$$\begin{aligned}w_t s_{t+1} &\le 1 + aw_t s_t - bw_t q_t + cw_t\\ &= w_{t-1}s_t - bw_t q_t + cw_t.\end{aligned}$$

Rearranging,

$$bw_t q_t \le w_{t-1}s_t - w_t s_{t+1} + cw_t.$$

Summing up as t varies from 0 to $T-1$ and noting that the sum telescopes,

$$\sum_{t=0}^{T-1} bw_t q_t \le \sum_{t=0}^{T-1} w_{t-1}s_t - w_t s_{t+1} + c\sum_{t=0}^{T-1} w_t = w_0 s_0 - w_{T-1}s_T + c\sum_{t=0}^{T-1} w_t \le w_0 s_0 + c\sum_{t=0}^{T-1} w_t.$$

Let $W_T = \sum_{t=0}^{T-1} w_t$. Dividing both sides by W_T we have,

$$\frac{1}{W_T}\sum_{t=0}^{T-1} b w_t q_t \leq \frac{w_0 s_0}{W_T} + c. \tag{28}$$

We now separate the proof into two cases:

[leftmargin=0.2in,itemsep=0.01in]**If** $a > 0$: Note that the left-hand side of (28) satisfies

$$b \min_{t=0,\ldots,T-1} q_t \leq \frac{1}{W_T}\sum_{t=0}^{T-1} b w_t q_t. \tag{29}$$

And for the right hand-side of (28) we have,

$$W_T = \sum_{t=0}^{T-1} w_t \geq T \min_{t=0,\ldots,T-1} w_t = T w_{T-1} \geq T w_T = \frac{T w_0}{1+a^T}. \tag{30}$$

Substituting with (30) in (29) and dividing both sides by b we get,

$$\min_{t=0,\ldots,T-1} q_t \leq \frac{1+a^T}{bT} s_0 + \frac{c}{b}.$$

If $a = 0$: then $w_t = w_0$ for all t and hence $w_T = T$, then (29) is equivalent to

$$\frac{1}{T}\sum_{t=0}^{T-1} b q_t \leq \frac{s_0}{T} + c.$$

Dividing both sides by b yields the lemma's claim.

■

- **Theorem 19** *Let Assumption 1 hold. Let* $\delta_0 = f(x_0) - f_*$ *and* $\Delta_{*,m} = \frac{1}{n}\sum_{i=1}^{n}(f_* - f_{*,m}^i)$*. Let* $\gamma \leq \frac{1}{2nL}$ *and* $\eta \leq \frac{1}{L}$*. Then for iterates* x_t *of Algorithm 1, we have*

$$\min_{t=0,\ldots,T-1} \mathbb{E}\left\|\nabla f(x_t)\right\|^2 \leq \frac{2\left(1+4\eta\gamma^2 n^2 L^3\right)^T}{\eta T}\delta_0 + 2\gamma^2 n L^3\left(\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m} + n\Delta_*\right) + 4L^2\eta\frac{M-C}{C\max\{M-1,1\}}\Delta_*.$$

Proof We start from L-smoothness (12):

$$\begin{aligned}
f(x_{t+1}) &\overset{(12)}{\leq} f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t\rangle + \frac{L}{2}\|x_{t+1}-x_t\|^2 \\
&= f(x_t) - \left\langle \nabla f(x_t), \eta\frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\rangle + \frac{L}{2}\left\|\eta\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\|^2 \\
&= f(x_t) - \eta\left\langle \nabla f(x_t), \frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\rangle + \frac{L}{2}\eta^2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\|^2.
\end{aligned}$$

Taking conditional expectation over cohort S_t, we get

$$\begin{aligned}
\mathbb{E}_{S_t} f(x_{t+1}) &\leq f(x_t) - \mathbb{E}_{S_t}\eta\left\langle \nabla f(x_t), \frac{1}{Cn}\sum_{m\in S_t}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\rangle + \mathbb{E}_{S_t}\frac{L}{2}\eta^2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\|^2 \\
&\leq f(x_t) - \eta\left\langle \nabla f(x_t), \frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\rangle + \mathbb{E}_{S_t}\frac{L}{2}\eta^2\left\|\frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1}\nabla f_m^{\pi_m^i}(x_{t,m}^i)\right\|^2.
\end{aligned}$$

Using $2\langle a,b\rangle = \|a+b\|^2 - \|a\|^2 - \|b\|^2$, we have

$$\begin{aligned}
\mathbb{E}_{S_t} f(x_{t+1}) &= f(x_t) + \mathbb{E}_{S_t}\frac{L}{2}\eta^2 \left\| \frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1} \nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 \\
&\quad - \left(\frac{\eta}{2}\|\nabla f(x_t)\|^2 + \frac{\eta}{2}\left\|\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2\right) + \frac{\eta}{2}\left\|\nabla f(x_t) - \frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 \\
&\le f(x_t) + \frac{L}{2}\eta^2\mathbb{E}_{S_t}\left\| \frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1} \nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 \\
&\quad - \left(\frac{\eta}{2}\|\nabla f(x_t)\|^2 + \frac{\eta}{2}\left\|\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2\right) + \frac{\eta}{2}\left\|\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\left(\nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right) - \nabla f_m^{\pi_m^i}(x_t)\right)\right\|^2.
\end{aligned}$$

Using L-smoothness, we get

$$\begin{aligned}
\mathbb{E}_{S_t} f(x_{t+1}) &\le f(x_t) + \frac{L}{2}\eta^2\mathbb{E}_{S_t}\left\| \frac{1}{Cn}\sum_{m\in S_t}\sum_{n=0}^{n-1} \nabla f_m^{\pi_m^i}\left(x_{t,m}^i\right)\right\|^2 \\
&\quad - \frac{\eta}{2}\|\nabla f(x_t)\|^2 + \frac{\eta}{2}L^2\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\left\|x_{t,m}^i - x_t\right\|^2.
\end{aligned}$$

Utilizing Lemma 16 and taking conditional expectation using tower property, we get

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1})|x_t\right] &\le f(x_t) + \frac{L}{2}\eta^2\left(2L(f(x) - f_*) + 2\frac{M-C}{C(M-1)}\Delta_*\right) \\
&\quad - \frac{\eta}{2}\|\nabla f(x_t)\|^2 + \frac{\eta}{2}L^2\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=0}^{n-1}\mathbb{E}\left[\left\|x_{t,m}^i - x_t\right\|^2|x_t\right].
\end{aligned}$$

Applying Lemma 17 we get

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1})|x_t\right] &\le f(x_t) + \frac{L}{2}\eta^2\left(2L(f(x) - f_*) + 2\frac{M-C}{C(M-1)}\Delta_*\right) \\
&\quad - \frac{\eta}{2}\|\nabla f(x_t)\|^2 + \frac{\eta}{2}L^2 4L\gamma^2 n^2(f(x_t) - f_*) + 2\gamma^2 n^2 L\Delta_* + 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m}.
\end{aligned}$$

Using $\eta \le \frac{1}{L}$ and taking full expectation, we have

$$\begin{aligned}
\mathbb{E}\left[f(x_{t+1})\right] &\le \mathbb{E}\left[f(x_t)\right] - \frac{\eta}{2}\mathbb{E}\|\nabla f(x_t)\|^2 \\
&\quad + \frac{L}{2}\eta^2\left(2\frac{M-C}{C(M-1)}\Delta_*\right) + \left(1 + 4\eta\gamma^2 n^2 L^3\right)\mathbb{E}(f(x_t) - f_*) + 2\gamma^2 n^2 L\Delta_* + 2\gamma^2 nL\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m}.
\end{aligned}$$

Applying Lemma 18 from Mishchenko et al. [19], we get

$$\min_{t=0,\ldots,T-1}\mathbb{E}\left\|\nabla f(x_t)\right\|^2 \le \frac{2\left(1 + 4\eta\gamma^2 n^2 L^3\right)^T}{\eta T}\delta_0 + 2\gamma^2 nL^3\left(\frac{1}{M}\sum_{m=1}^{M}\Delta_{*,m} + n\Delta_*\right) + 4L^2\eta\frac{M-C}{C\max\{M-1,1\}}\Delta_*.$$

■

Appendix C. Small Server Stepsize

In this section, we present a result when it is useful to pull back the last iterates of local passes. In particular, we show that one can reduce the variance of FedAvg with uniform partial participation.

Theorem 20 *Assume that all losses $f_{m,i}$ are L-smooth and μ-strongly convex. Define $\alpha = \frac{\eta}{\gamma n}$. Let $\gamma \le \frac{1}{L}$ and $0 \le \alpha < 1$. Then for iterates x_t generated by Algorithm 1 we have*

$$\mathbb{E}\,\|x_T - x_*\|^2 \le (1-\alpha+\alpha(1-\gamma\mu)^n)^T\,\|x_0 - x_*\|^2 + \frac{\alpha}{(1-\alpha)\,(1-(1-\gamma\mu)^n)}\gamma^2\frac{\sigma_*^2}{C} + 2\gamma^3\sigma_{\text{rad}}^2\frac{1}{1-(1-\gamma\mu)^n}\sum_{i=0}^{n-1}(1-\gamma\mu)^i.$$

Proof Let us denote $f_{S_t} = \frac{1}{|S_t|}\sum_{m\in S_t} f_m$. We start by rewriting the distance to the optimum in the following way:

$$\begin{aligned} x_{t+1} - x_* &= (1-\alpha)x_t + \alpha x_t^n - x_* \\ &= (1-\alpha)x_t + \alpha x_t^n - (1-\alpha)(x_* + \frac{\alpha}{1-\alpha}\gamma n\nabla f_{S_t}(x_*)) - \alpha(x_* - \gamma n\nabla f_{S_t}(x_*)). \end{aligned}$$

Therefore, by convexity of the squared norm,

$$\|x_{t+1} - x_*\|^2 \le (1-\alpha)\|x_t - (x_* + \frac{\alpha}{1-\alpha}\gamma n\nabla f_{S_t}(x_*))\|^2 + \alpha\|x_t^n - (x_* - \gamma n\nabla f_{S_t}(x_*))\|^2.$$

We bound the two terms in the right-hand side separately. For the first term, it suffices to take expectation over the sampling of client cohort S_t,

$$\begin{aligned} {}_{S_t}\|x_t - (x_* + \frac{\alpha}{1-\alpha}\gamma n\nabla f_{S_t}(x_*))\|^2 &\overset{(18)}{=} \|x_t - x_*\|^2 + \frac{\alpha^2}{(1-\alpha)^2}\gamma^2 n^2_{S_t}\|\nabla f_{S_t}(x_*)\|^2 \\ &= \|x_t - x_*\|^2 + \frac{\alpha^2}{(1-\alpha)^2}\gamma^2 n^2\frac{\sigma_*^2}{|S_t|}. \end{aligned}$$

For the second term, we use the results of prior work on convergence of RR that gives

$$\|x_t^n - (x_* - \gamma\nabla f_{S_t}(x_*))\|^2 \le (1-\gamma\mu)^n\|x_t - x_*\|^2 + 2\gamma^3\sigma_{\text{rad}}^2\sum_{i=0}^{n-1}(1-\gamma\mu)^i,$$

where, as shown by [20], $\sigma_{\text{rad}} \ge 0$ is some constant satisfying

$$\sigma_{\text{rad}}^2 \le L\sum_{m=1}^{M}(n^2\|\nabla f_m(x_*)\|^2 + \frac{n}{4}\sigma_{*,m}^2).$$

Notice that the upper bound depends on α in a nonlinear way, so the optimal value of α would often lie somewhere in the interval $(0,1)$. Recurrence $a_{t+1} \le (1-\rho)a_t + c$ implies by induction $a_t \le (1-\rho)^t a_0 + \frac{c}{\rho}$, so by propagating the bound above to x_0, we obtain

$$\begin{aligned} \|x_t - x_*\|^2 \le{}& (1-\alpha+\alpha(1-\gamma\mu)^n)^t\|x_0 - x_*\|^2 + \frac{\alpha}{(1-\alpha)(1-(1-\gamma\mu)^n)}\gamma^2\frac{\sigma_*^2}{|S_t|} \\ &+ 2\gamma^3\sigma_{\text{rad}}^2\frac{1}{1-(1-\gamma\mu)^n}\sum_{i=0}^{n-1}(1-\gamma\mu)^i. \end{aligned}$$

Notice that the last term does not change with α, so its optimal value is completely determined by the first two terms. ■