A Discussions on the Existing Algorithms

In this section, we discuss some of the applications of the proposed framework. We first show that by properly choosing the two controllers and the discretization scheme, the proposed framework can be specialized to a number of popular decentralized learning algorithms. Second, we show how the proposed framework can help us identify the relationship between different algorithms, as well as facilitate development of new algorithms.

A.1 Existing Decentralized Algorithms as Discretized Multi-Rate Systems

We map some of the existing distributed algorithms into the discretized multi-rate system with specific GCFL and LCFL.

First let us begin with the DO algorithms:

DGD [3]: The update step of DGD is:

$$\mathbf{x}(k+1) = W\mathbf{x}(k) - c\nabla f(\mathbf{x}(k)),$$

where $c > 0$ is the stepsize. So it is the discretization Case III of the system with the corresponding continuous-time controllers:

$$u_{g,x} = (I - W)\mathbf{x}, \quad u_{\ell,x} = \nabla f(\mathbf{x}).$$

DLM [6]: The update step of Decentralized Linearized-ADMM (DLM) algorithm is:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{x}(k) - \eta\left(\nabla f(\mathbf{x}(k)) + c(I-W)\mathbf{x}(k) + \mathbf{v}(k)\right),\\ \mathbf{v}(k+1) &= \mathbf{v}(k) + c(I-W)\mathbf{x}(k+1).\end{aligned}$$

So it is the discretization Case III of the system with the corresponding continuous-time controllers:

$$\begin{aligned}&u_{g,x} = c(I-W)\mathbf{x} + \mathbf{v}, \quad u_{g,v} = (I-W)\mathbf{x},\\ &u_{\ell,x} = \nabla f(\mathbf{x}), \quad u_{\ell,v} = 0.\end{aligned}$$

Then we list some of the FL algorithms:

FedAvg [8]: The update step of FedAvg and Local GD is:

$$\mathbf{x}(k+1) = \begin{cases}\mathbf{x}(k) - \eta\nabla f(\mathbf{x}(k)), \ k \mod Q \neq 0,\\ R\mathbf{x}(k) - \eta\nabla f(\mathbf{x}(k)), k \mod Q = 0.\end{cases}$$

We can see that FedAvg cannot be translated into a continuous-time system as it does not have a persistent GCFL:

$$u_{g,x} = \begin{cases}0, \ t \neq k\tau_g,\\ (I-R)\mathbf{x}(t)\delta(t), t = k\tau_g = 0,\end{cases}$$

where $\delta(t)$ denotes the Dirac delta function.

FedProx [9]: By assuming the local problem of FedProx is solve by gradient descent, the update step of FedProx is:

$$\mathbf{x}(k+1) = \begin{cases}\mathbf{x}(k) - \eta_1\nabla f(\mathbf{x}(k)) - \eta_2(\mathbf{x}(k) - \mathbf{x}(k_0)), \ k \mod Q \neq 0, \ k_0 = k - (k \mod Q),\\ R\mathbf{x}(k) - \eta_1\nabla f(\mathbf{x}(k)) - \eta_2(\mathbf{x}(k) - \mathbf{x}(k_0)), k \mod Q = 0, \ k_0 = k.\end{cases}$$

So it is the discretization Case I or IV of the system with the corresponding continuous-time controllers:

$$u_{g,x} = (I - R)\mathbf{x}, \quad u_{\ell,x} = \nabla f(\mathbf{x}).$$

FedPD [11]: By assuming the local problem of FedPD is solve by gradient descent, the update step of FedPD is:

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \eta_1(\nabla f(\mathbf{x}(k)) + \mathbf{z}(k) + \eta_2(\mathbf{x}(k_0) - R\mathbf{x}(k_0))), \ k_0 = k - (k \mod Q),$$

$$\mathbf{v}(k+1) = \begin{cases}R\mathbf{x}(k), k \mod Q = 0\\ \mathbf{v}(k), k \mod Q \neq 0,\end{cases}$$

$$\mathbf{z}(k+1) = \begin{cases}\mathbf{z}(k) + \frac{1}{\eta_2}(\mathbf{x}(k) - \mathbf{v}(k)), k \mod Q = 0\\ \mathbf{z}(k), k \mod Q \neq 0,\end{cases}$$

So it is the discretization Case I or IV of the system with the corresponding continuous-time controllers:

$$\begin{aligned} u_{g,x} &= \mathbf{x} - \mathbf{v}, \quad u_{g,v} = \mathbf{v} - R\mathbf{x}, \\ u_{\ell,x} &= \nabla f(\mathbf{x}) + \mathbf{z}, \quad u_{\ell,v} = 0, \quad u_{\ell,z} = -(\mathbf{x} - \mathbf{v}). \end{aligned}$$

We can observe that $\mathbf{v}$ is tracking $R\mathbf{x}$. Replacing $\mathbf{v}$ with $R\mathbf{x}$ we have the following controllers:

$$\begin{aligned} u_{g,x} &= (I - R)\mathbf{x}, \\ u_{\ell,x} &= \nabla f(\mathbf{x}) + \mathbf{z} \quad u_{\ell,z} = -(I - R)\mathbf{x}. \end{aligned}$$

Finally, we give an example of the rate-optimal algorithms:

Scaffold [10]: The update step of Scaffold is:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) - \eta_1(\nabla f(\mathbf{x}(k)) - \mathbf{z}(k) + \mathbf{v}_2(k_0)),\ k_0 = k - (k \mod Q), \\ \mathbf{v}_1(k+1) &= \begin{cases} \mathbf{v}_1(k) + \eta_2 R(\mathbf{x}(k) - \mathbf{v}_1(k)), & k \mod Q = 0 \\ \mathbf{v}_1(k), & k \mod Q \neq 0, \end{cases} \\ \mathbf{v}_2(k+1) &= \begin{cases} \mathbf{v}_2(k) - R(\mathbf{v}_2(k) - \frac{1}{Q\eta_1}(\mathbf{v}_1(k) - \mathbf{x}(k))), & k \mod Q = 0 \\ \mathbf{v}_2(k), & k \mod Q \neq 0, \end{cases} \\ \mathbf{z}(k+1) &= \mathbf{z}(k) - \frac{1}{Q}\mathbf{v}_2(k) - \frac{1}{Q\eta_1}(\mathbf{x}(k+1) - \mathbf{x}(k)), \end{aligned}$$

So it is the discretization Case IV of the system with the corresponding continuous-time controllers:

$$\begin{aligned} u_{g,x} &= \mathbf{v}_2, \quad u_{g,v} = [R(\mathbf{v}_1 - \mathbf{x}); R\mathbf{v}_2 - \frac{1}{\eta_1}R(\mathbf{v}_1 - \mathbf{x})], \\ u_{\ell,x} &= \nabla f(\mathbf{x}) - \mathbf{z}, \quad u_{\ell,v} = 0, \quad u_{\ell,z} = \mathbf{v}_2 + \frac{1}{\eta_1}\dot{\mathbf{x}}. \end{aligned}$$

xFilter [12]: The update step of xFilter is:

$$\begin{aligned} \mathbf{x}(k+1) &= \eta_1((1-\eta_2)I - \eta_2(I - W))\mathbf{x}(k) + (1-\eta_1)\mathbf{x}(k-1) + \eta_2\eta_1\mathbf{v}(k_0), k_0 = k - (k \mod K) \\ \mathbf{v}(k+1) &= \begin{cases} \mathbf{v}(k) + (\mathbf{z}_1(k) - \mathbf{z}_2(k)) - (I - W)\mathbf{x}(k), & k \mod K = 0 \\ \mathbf{v}(k), & k \mod K \neq 0, \end{cases} \\ \mathbf{z}_1(k+1) &= \begin{cases} \mathbf{x}(k) - \eta_3\nabla f(\mathbf{x}(k)), & k \mod K = 0 \\ \mathbf{z}_1(k), & k \mod K \neq 0, \end{cases} \\ \mathbf{z}_2(k+1) &= \begin{cases} \mathbf{z}_1(k), & k \mod K = 0 \\ \mathbf{z}_2(k), & k \mod K \neq 0, \end{cases} \end{aligned}$$

So it is the discretization Case V of the system with the corresponding continuous-time controllers:

$$\begin{aligned} u_{g,x} &= \frac{\eta_2}{2 - \eta_2}W\mathbf{x}, \quad u_{g,v} = W\mathbf{x}, \\ u_{\ell,x} &= \frac{\eta_2}{2 - \eta_2}\mathbf{v}, \quad u_{\ell,v} = (\mathbf{z}_1 - \mathbf{z}_2), \quad u_{\ell,z} = [\eta_3\nabla f(\mathbf{x}); -(\mathbf{z}_1 - \mathbf{z}_2)]. \end{aligned}$$

A.2 Existing Algorithms Connections

From the previous section, we have identified the controllers used by each algorithm in their continuous-time counterparts.

We can summarize the above interpretation of the existing algorithm into Table 2. From the table, we can see that some of the algorithms corresponds to the same continuous-time dynamic system and the only difference in the way we discretize the system. For examples, FedPD and DLM have the same continuous-time dynamics while DLM corresponds to Case III that GCFL and LCFL have the same sampling intervals, while FedPD correspond to Case I,IV that $\tau_g = Q\tau_\ell$; Scaffold and xFilter also have the same continuous-time dynamics.

Algorithm	Global Consensus	Local Computation	FL	RO	DO
DGD	$(I-W)\mathbf{y}$	$\nabla f(\mathbf{x})$	FedProx	–	DGD
DLM	$c(I-W)\mathbf{x}+\mathbf{v}$	$\nabla f(\mathbf{x})$	FedPD	–	DLM
xFilter	$(I-W)\mathbf{x}+\mathbf{v}$	$u_{\ell,v} = -\dot{\nabla} f(\mathbf{x})$	Scaffold	xFilter	–

Table 2: The summary of the controllers used in different algorithms. In GCFL and LCFL we abstract the most important steps of the controller.

From the table we can see that there are many missing blanks. Each of these blanks represents a new algorithm. Also, we cam combine different GCFL and LCFL to create new algorithms that are not in this table. In the next section, we use DGT as an example to show how it can be extended to Case I,IV and to a different GCFL.

B Example: Analysis and Extensions of Gradient Tracking

In this section, we use the well-known gradient tracking algorithm as an example to illustrate how our proposed framework can be used in practice to analyze algorithm behavior, and to facilitate the development of new algorithms.

B.1 The Gradient Tracking Algorithm

The iteration of the original gradient tracking algorithm is given below:

$$\begin{aligned} &\mathbf{x}(k+1) = W\mathbf{x}(k) - c\mathbf{v}(k),\\ &\mathbf{v}(k+1) = W\mathbf{v}(k) + \nabla f(\mathbf{x}(k+1)) - \nabla f(\mathbf{x}(k)), \end{aligned} \tag{8}$$

where $c > 0$ is some stepsize. Note that the algorithm only has one auxiliary consensus state $\mathbf{v}$. Under the assumption that a) W is symmetric and doubly stochastic; b) f_i's has Lipschitz gradients and non-convex; c) $\sum_i f_i$ is lower bounded, this algorithm converges to the stationary point of the problem at a rate of $\mathcal{O}(1/T)$ [31, 32].

Our approach is to first analyze the corresponding continuous-time double-feedback system, and apply appropriate discretization schemes and utilize the corresponding convergence results.

B.2 Continuous-time Analysis

We begin by analyzing the continuous-time counterpart of the gradient tracking algorithm. First, notice that the gradient tracking algorithm falls into the case that $\tau_g = \tau_\ell$, because communication and computation happen at the same time-scale. By letting $\tau_g = \tau_\ell \to 0$, we obtain the following continuous-time dynamic:

$$\begin{aligned} &\dot{\mathbf{x}}(t) = -\eta_g(t)(I-W)\mathbf{x}(t) - \eta_\ell(t)(c\mathbf{v}(t)),\\ &\dot{\mathbf{v}}(t) = -\eta_g(t)(I-W)\mathbf{v}(t) + \eta_\ell(t)(\nabla \dot{f}(\mathbf{x})). \end{aligned} \tag{9}$$

where $\eta_g(t) = 1, \eta_\ell(t) = 1, \forall\, t$. In the above system, the global controllers given by:

$$u_{g,x} = (I-W)\mathbf{x}(t), \quad u_{g,v} = (I-W)\mathbf{v}(t),$$

and local update controllers given by:

$$u_{\ell,x} = \eta\mathbf{v}(t), \quad u_{\ell,v} = -\nabla\dot{f}(\mathbf{x}(t)),$$

where $\nabla\dot{f}(\mathbf{x}(t)) := \left\langle \nabla^2 f(\mathbf{x}(t)), \dot{\mathbf{x}}(t)\right\rangle$.

Next, let us verify the assumptions A1-A4. First, it is straightforward to prove A2 with the definition of u_g. For A1, we know that W is doubly-stochastic and symmetric and the communication graph is connected and $\mathbb{1}$ is an eigenvector of $I-W$. Therefore $\mathbf{y}$ is an averaging system with linear convergence rate C_g equals to the eigenvalue of $I-W$ with the second smallest magnitude.

For A3, we can verify it by the following steps:

$$\begin{aligned} \|G_{\ell,x}(x,v;f_i) - G_{\ell,x}(x',v';f_i)\| &= \eta\,\|v-v'\|\\ \|G_{\ell,v}(x,v;f_i) - G_{\ell,x}(x',v';f_i)\| &= \left\|\left\langle\nabla^2 f_i(x), \dot{x}\right\rangle - \left\langle\nabla^2 f_i(x'), \dot{x}'\right\rangle\right\|\\ &\le \left(\left\|\nabla^2 f_i(x)\right\| + \left\|\nabla^2 f_i(x')\right\|\right)\|\dot{x}-\dot{x}'\|\\ &\le 2cL_f\,\|v-v'\|, \end{aligned}$$

where L_f is the constant of the Lipschitz gradient. So the smoothness constant of the local controller g_ℓ can be expressed as $L = c \max\{2L_f, 1\}$.

Finally, to check whether the LCFL satisfies A4, let us initialize $\mathbf{v}(t) = \nabla f(\mathbf{x}(t))$, and assume that $\eta_g(t) = 0$ in (9), that is, the GCFL is inactive. Then we have:

$$\mathbf{v}(t+\tau) = \nabla f(\mathbf{x}(t+\tau)) \tag{10}$$

$$\dot{\mathbf{x}}(t+\tau) = -c\mathbf{v}(t+\tau) = -c\nabla f(\mathbf{x}(t+\tau)). \tag{11}$$

The algorithm becomes the gradient flow algorithm that satisfies A4 [14].