Push–Pull With Device Sampling

Yu-Guan Hsieh ⊕, Yassine Laguel ⊕, Franck Iutzeler ⊕, and Jérôme Malick ⊕

Abstract—We consider decentralized optimization problems in which a number of agents collaborate to minimize the average of their local functions by exchanging over an underlying communication graph. Specifically, we place ourselves in an asynchronous model where only a random portion of nodes perform computation at each iteration while the information exchange can be conducted between all the nodes and in an asymmetric fashion. For this setting, we propose an algorithm that combines gradient tracking with a network-level variance reduction (in contrast to variance reduction within each node). This enables the nodes to track the average of the gradients of the objective functions. Our theoretical analysis shows that the algorithm converges linearly, when the local objective functions are strongly convex, under mild connectivity conditions on the expected mixing matrices. In particular, our result does not require the mixing matrices to be doubly stochastic. In the experiments, we investigate a broadcast mechanism that transmits information from computing nodes to their neighbors and confirm the linear convergence of our method on both synthetic and real-world datasets.

Index Terms—Convex optimization, decentralized optimization, device sampling, random gossip.

I. INTRODUCTION

In this article, we focus on solving

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{M} \sum_{i=1}^{M} f_i(x)$$

(\text{P})

where each function $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is available only locally at the $i$th node of a graph. Hence, in order to reach a consensus on the minimum of (P), the $M$ nodes have to communicate using the graph’s edges.

Such decentralized optimization problems have been widely studied in the literature at least since the pioneering works of Tsitsiklis et al. [1] and Bertsekas and Tsitsiklis [2]. In terms of applications, decentralized optimization methods are popular for regression or classification problems when the communication possibilities between the nodes are scarce and cannot be handled by a central entity (e.g., for wireless sensor networks, IoT-enabled edge devices, etc.); see the recent surveys [3], [4], [5], [6]. In these applications, the workload and communications between the nodes are of primary importance.

The computation at the node level mainly depends on which optimization method serves as a basis. If the nodes are able to solve optimization subproblems, the alternating direction method of multipliers and other dual methods can be extended to the distributed setting [7], [8]. At the other end of the spectrum, stochastic gradient methods are very popular since they require minimal computation at each node [9]. Gradient-based methods offer a good compromise between these two extremes and currently know a rebirth, especially for machine learning applications; see, e.g., the recent work in [10].

In terms of exchanges, all communications between nodes have to go through the edges of the graph. If the graph is undirected (i.e., the edges are all bidirectional), the nodes can gossip to average their values. Mathematically, this corresponds to multiplying the agents’ states by a doubly-stochastic matrix; see [6, Sec. II] for details. However, if the graph is directed, such direct gossiping is no longer possible since maintaining both a consensus among the nodes and the average of their values is not possible at the same time [11]. To overcome this problem, two main types of methods have been developed. First, push–sum methods (or ratio consensus) consist in exchanging an additional “weighting”; these methods can reach an average consensus for the ratio of the two values [12], [13], [14]. However, the analysis of push–sum gradient methods is often quite involved and the algorithm can become numerically unstable due to division by very small values, see, e.g., the simulations in [15] as well as references therein. Second, push–pull methods rely on two communications steps with different mixings to maintain convergence, offering strong theoretical guarantees as well as good practical performance [15], [16].

Finally, a desirable feature in decentralized methods is the possibility to allow the nodes to randomly awake, compute, and send/receive information, which is generally called randomized gossiping [17] or asynchronous decentralized methods [5], [18]. In terms of analysis, this consists in replacing the fixed communication matrices with random ones having the support corresponding to the active links; this was actively studied for decentralized gradient methods, including push–pull gradient [15].
A. Contributions and Outline

In this article, we focus on gradient-based methods for decentralized asynchronous optimization on directed graphs. We propose and analyze an asynchronous push–pull gradient algorithm where only a fraction of the nodes are actively computing a local gradient at each iteration. This feature is inspired by the device sampling (or client selection) procedure in federated learning [19], [20]. This popular mechanism enables us to take into account the fact that all the nodes may not be available at all time and, furthermore, that querying all gradients at each iteration may be a waste of computational power if the nodes’ values only change by a little amount.

In terms of algorithm, device sampling calls for a variance reduction mechanism in order to mitigate the noise induced by the sampling of the nodes. To achieve this, we introduce a SAGA-like [21] update at the network level; see Section II-C4. This additional step thus requires an original analysis.

The remainder of the article is structured as follows. The introduction is completed by an overview on related literature. In Section II, we present our general algorithmic template [Push–Pull with Device Sampling (PPDS)], together with some specific cases of interest, connecting our method with existing methods. In Section III, we provide linear convergence results under classical convexity/smoothness assumptions on the objective functions and weak assumptions on communications. Sections IV and V are dedicated to the detailed convergence analysis and illustrative numerical simulations. Section VI concludes this article. Finally, proofs of a couple of technical intermediate results are given in the Appendix.

B. Related Works

Direct extensions of the gradient method to the decentralized setting rely on decreasing stepsizes to converge and are, thus, limited to sublinear convergence rates, even if the minimized functions are smooth and strongly convex. To overcome this situation, the gradient tracking technique was introduced; it consists in dynamically tracking the average value of the gradient and using this value instead of the local gradient. This technique enables the use of a fixed stepsize and exhibits much better rates in theory and in practice [22], [23], [24], [25]; see also the recent work in [26]. Gradient tracking can be intuitively seen as a variance reduction at the network level. The method presented in this article extends this idea of variance reduction to device sampling. Note also that in the case where the nodes’ functions are themselves a finite sum, this sum can be sampled, and variance reduction can be additionally applied at the node level [27], [28]; however, this specific form is out of the scope of this article.

AB/Push–Pull gradient methods naturally involve gradient tracking; see, e.g., [15, Rem. 1] and more generally [15], [16], [29], [30]. These algorithms share common ingredients and mainly differ in their communications models. The works that are the most closely related to the asynchronous directed setup considered in this article are [15] and [29]. These two papers study an asynchronous version of AB/push–pull, which share similarities, in the update and the communication scheme, with our proposed method (more precisely, with the special setup in Section II-C3). However, in contrast to our method, these methods require every node that is involved in the communication step to perform a local update. Moreover, the analysis in [15] only works for the more restrictive case where the nondiagonal coefficients of the mixing matrices are sufficiently small. Note finally that Zhang and You [29] do not consider a random network model but instead perform an analysis in terms of the worst case dependence on the delays. This analysis is thus complementary to our work.

C. Basic Notation and Definitions

Throughout the article, we use bold lowercase letters to denote vectors and capital letters to denote matrices. $I_k$ and $1_k$, respectively, represent the identity matrix of size $k \times k$ and the $k$-dimensional vector containing all ones. The subscript is omitted when the dimension is clear from the context. We also define $J = 1_M 1_M^\top / M$ as the projection matrix onto the consensus space and denote by $\rho(P)$ the spectral radius of a matrix $P$.

The interaction topology between the nodes is modeled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices (nodes) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, such that node $i$ can send information to $j$ only if $(i, j) \in \mathcal{E}$. The out-neighbors and in-neighbors of a node $i$ are, respectively, defined by

$$\mathcal{N}^\text{out}_i = \{ j \in \mathcal{V} : (i, j) \in \mathcal{E} \}, \mathcal{N}^\text{in}_i = \{ j \in \mathcal{V} : (j, i) \in \mathcal{E} \}.$$  

When the graph is undirected, the two sets coincide and we simply write $\mathcal{N}_i$. We say that the matrix $W = (w_{ij}) \in \mathbb{R}^{M \times M}$ is compatible with the underlying communication topology if $w_{ij} = 0$ whenever $(j, i) \notin \mathcal{E}$.

Finally, we introduce the aggregate objective function, $F(X) = \sum_{i=1}^{M} f_i(x_i)$, as a function of the variable $X = [x_1, \ldots, x_M]^\top \in \mathbb{R}^{M \times d}$. When $F$ is differentiable, we have

$$\nabla F(X) = [\nabla f_1(x_1), \ldots, \nabla f_M(x_M)]^\top.$$  

II. ALGORITHMS: EXISTING, NEW, AND EXAMPLES

In this section, we present our asynchronous push–pull gradient algorithm with device sampling. Prior to that, we recall the existing AB/push–pull method [15], [16], which inspires our algorithm. After detailing our general template, we instantiate it in several situations of interest, revealing its versatility.

A. AB/Push–Pull Method

If all the nodes are active at each iteration, the communication setup reduces to that of synchronous decentralized optimization. In this situation and assuming that the functions $f_i$ are differentiable, the AB/push–pull algorithm [15], [16] is described as follows. In addition to the decision variables $x_i$ that should minimize $f$, a variable $y_i$ is introduced to track the gradient of $f$. Then, provided $\eta > 0$ a constant stepsize and two mixing matrices $A = (a_{ij}) \in \mathbb{R}^{M \times M}$ and $B = (b_{ij}) \in \mathbb{R}^{M \times M}$,
the update of the algorithm at iteration $t$ writes
\[ x_{i}^{t+1} = \sum_{j \in V} a_{ij} x_{j}^{t} - \eta y_{i}^{t} \]
\[ y_{i}^{t+1} = \sum_{j \in V} b_{ij} y_{j}^{t} + \nabla f_{i}(x_{j}^{t+1}) - \nabla f_{i}(x_{j}^{t}) \]

It is required that $A$ and $B$ have nonnegative weights and be, respectively, row-stochastic ($A_{1} = 1$) and column-stochastic $(1^{T} B = 1^{T})$. With the notation $Y_{t} = [y_{1}^{t}, \ldots, y_{M}^{t}]^{T}$, the update can also be written, in a matrix form, as
\[ X_{t+1} = A_{t} X_{t} - \eta Y_{t} \]
\[ Y_{t+1} = B_{t} Y_{t} + \nabla F(X_{t+1}) - \nabla F(X_{t}). \]  

(PP)

Intuitively, the use of row-stochastic matrices drive $x_{i}^{t}$ to consensus while the use of column-stochastic matrices preserves the total mass, i.e., $1^{T} B = 1^{T}$ for any $\omega \in \mathbb{R}^{M}$. Moreover, if the difference $\nabla f_{i}(x_{j}^{t+1}) - \nabla f_{i}(x_{j}^{t})$ tends to zero, $y_{i}^{t}$ converges to a multiple of $\sum_{j \in V} \nabla f_{i}(x_{j}^{t})$. In fact, from the Perron–Frobenius theorem, we know that if $B$ is primitive $1$, then $\lim_{t \to +\infty} B^{t} = \pi_{B} 1^{T}$ where $\pi_{B}$ is the right eigenvector of $B$ associated with the eigenvalue $1$ such that $1^{T} \pi_{B} = 1$. Therefore, asymptotically every $x_{i}^{t}$ descends in the opposite direction to the gradient of $f$. Mathematically, it can be proven that under standard convexity assumptions AB/push–pull converges linearly with sufficiently small constant stepsize $\eta$ [32, Th. 1].

### B. Proposed PPDS

Our algorithm can be viewed as a generalization of AB/push–pull to handle the device sampling mechanism. First, in order to allow for asynchronicity, let $(A_{t})_{t \in \mathbb{N}}$ and $(B_{t})_{t \in \mathbb{N}}$ be two sequences of mixing matrices that are compatible with the underlying communication topology. Now, to handle device sampling, we denote by $\mathcal{V}_{t}$ the set of nodes that are active at time $t$. This means that node $i$ computes a local gradient at round $t$ if and only if $i \in \mathcal{V}_{t}$.

With the notations $A_{t} = (a_{ij}^{t})$, $B_{t} = (b_{ij}^{t})$, and $D_{t} = \text{diag}(1_{\mathcal{V}_{t}})$, i.e., $D_{t}$ is the diagonal matrix in $\mathbb{R}^{M \times M}$ whose $i$th diagonal element is $1$ if $i \in \mathcal{V}_{t}$ and $0$, otherwise, each iteration of our proposed PPDS can be stated in the compact form
\[ Y_{t+\frac{1}{2}} = Y_{t} + D_{t} (\nabla F(X_{t}) - \nabla F(Z_{t})) \]
\[ X_{t+\frac{1}{2}} = X_{t} - \eta D_{t} Y_{t+\frac{1}{2}} \]
\[ Z_{t+1} = D_{t} X_{t} + (I - D_{t}) Z_{t} \]
\[ Y_{t+1} = B_{t} Y_{t+\frac{1}{2}} \]
\[ X_{t+1} = A_{t} X_{t+\frac{1}{2}} \]  

(PPDS)

Several remarks are in order. First, we introduce auxiliary local variable $z_{i}^{t}$ for each node and write $Z_{t} = [z_{1}^{t}, \ldots, z_{M}^{t}]^{T}$. The presence of these variables indicates the nodes store their last computed gradient. This is necessary because $x_{i}^{t}$ can be modified by network communication between two successive activations of node $i$. In fact, while only the active nodes perform local updates at each iteration, the inactive nodes can be involved in the communication process. This flexibility allows us to take into account a wider class of algorithms, as illustrated in the forthcoming examples. Finally, as in AB/push–pull, we only require the matrices $(A_{t})_{t \in \mathbb{N}}$ and $(B_{t})_{t \in \mathbb{N}}$ to be, respectively, row- and column-stochastic. This means that we allow for one-way communication and, in particular, inactive nodes may passively receive information without sending it back to their local states.

In terms of implementation, our method (PPDS) gives Algorithm 1 for asynchronous optimization on directed graphs. In the next section, we discuss special cases and show that we recover existing algorithms.

### C. Special Cases

1) **AB/Push–Pull**: We first demonstrate that the original AB/push–pull algorithm [15], [16], indeed, falls within the PPDS framework. For this, we fix $A_{t} = A$, $B_{t} = B$, and $\mathcal{V}_{t} = \mathcal{V}$. Then, after rearranging, the (PPDS) update can be written as
\[ X_{t+1} = A (X_{t} - \eta Y_{t+\frac{1}{2}}) \]
\[ Y_{t+\frac{1}{2}} = B Y_{t+\frac{1}{2}} + \nabla F(X_{t+1}) - \nabla F(X_{t}) \]

This is exactly the adapt-then-combine variant of AB/push–pull as presented in [15].

2) **Communication Between Active Nodes**: We can imagine a situation where only active agents participate in the communication. Then, these active agents may communicate with each other using mixing matrices $A(\mathcal{V}_{t})$, $B(\mathcal{V}_{t})$ that are compatible with the induced subgraph $G[\mathcal{V}_{t}]$, defined by the vertex set $\mathcal{V}_{t}$ and the edges of $\mathcal{E}$ that connect two vertices of $\mathcal{V}_{t}$. For example, if the graph is symmetric and $A(\mathcal{V}_{t}) = B(\mathcal{V}_{t})$ is the Metropolis matrix of $G[\mathcal{V}_{t}]$, we have $A_{t} = B_{t}$ and
\[ a_{ij}^{t} = \begin{cases} \frac{1}{\text{deg}_{G}(i)} \text{deg}_{G}(j) & \text{if } i \neq j \text{ and } \{i,j\} \in \mathcal{E} \cap \mathcal{O}_{\mathcal{V}_{t}} \\ 0 & \text{if } i = j \end{cases} \]

where $\text{deg}_{G}(i) = \text{card}(\mathcal{N}(i) \cap \mathcal{V}_{t})$ is the degree of $i \in \mathcal{V}_{t}$ in the induced graph $G[\mathcal{V}_{t}]$.

3) **Broadcast-Type Update**: As mentioned previously, our algorithm allows inactive nodes to passively receive information

---

1A square nonnegative matrix $W$ is called primitive if there exists a power $k \geq 1$ such that $W^{k} > 0$; see [31, Th. 8.5.2].

---

**Algorithm 1: PPDS (at each node $i$).**

1: Initialize: $y_{i}^{1} = \nabla f_{i}(x_{i}^{1})$; $z_{i}^{1} = x_{i}^{1}$
2: for $t = 1, 2, \ldots$ do
3: Local update
4: if $i \in \mathcal{V}_{t}$ then
5: $y_{i}^{t+\frac{1}{2}} \leftarrow y_{i}^{t} + \nabla f_{i}(x_{i}^{t}) - \nabla f_{i}(z_{i}^{t})$
6: $x_{i}^{t+\frac{1}{2}} \leftarrow x_{i}^{t} - \eta y_{i}^{t+\frac{1}{2}}$
7: Set $z_{i}^{t+1} \leftarrow x_{i}^{t}$ and store $\nabla f_{i}(z_{i}^{t+1})$
8: else
9: $y_{i}^{t+\frac{1}{2}} \leftarrow y_{i}^{t+\frac{1}{2}} + \nabla f_{i}(x_{i}^{t+\frac{1}{2}}) - \nabla f_{i}(z_{i}^{t+1})$
10: end if
11: Communication
12: $x_{i}^{t+1} = \sum_{j \in \mathcal{V}} a_{ij}^{t} x_{j}^{t+\frac{1}{2}}$ \(\triangleright A_{t}\) is row-stochastic
13: $y_{i}^{t+1} = \sum_{j \in \mathcal{V}} b_{ij}^{t} y_{j}^{t+\frac{1}{2}}$ \(\triangleright B_{t}\) is column-stochastic
14: end for
from active nodes. Therefore, the active nodes can simply broadcast their local variables to their neighbors, no matter whether these neighbors are active or not. To ensure the row-stochasticity of \( A_t \), the received \( x_i^t \)'s are averaged out. On the other hand, to guarantee the column-stochasticity of \( B_t \), an active node divides its \( y_i^t \) by the number of nodes it sends the information to, as usually done in a push-sum scheme.

For concreteness, let us denote by \( N_{j,t}^{out} \) the set of neighbors that active worker \( j \in \mathcal{V}_t \) transmit information to (including itself) in round \( t \) and by \( \mathcal{N}_{i,t}^{in} = \{ j \in \mathcal{V}_t : i \in N_{j,t}^{out} \} \) the set of active workers that send information to \( i \) in this same round. The mixing matrices \( A_t \) and \( B_t \) are then defined by

\[
\begin{align*}
a_{ij}^t &= \frac{1}{\text{card}(N_{i,t}^{in}\cup\{i\})} \quad \text{if} \ j \in N_{i,t}^{in}\cup\{i\} \\quad \text{0 otherwise}, \\
b_{ij}^t &= \frac{1}{\text{card}(N_{j,t}^{out}\cup\{j\})} \quad \text{if} \ j \in N_{j,t}^{out}\cup\{j\} \\quad 1 \quad \text{if} \ j = i \quad \text{0 otherwise}.
\end{align*}
\]

In this example, we see that our method offers an additional degree of freedom compared to G-push–pull [15] since in that algorithm \( a_{ij}^t > 0 \) only if \( i, j \in \mathcal{V}_t \); this is not necessarily the case in our approach.

4) SAGA: SAGA [21] is a well-known (centralized) variance reduction methods that replace the stochastic gradient \( \nabla f_i(x_i) \) with an unbiased gradient estimator with diminishing variance. For this, we store a table of gradients \( (\nabla f_i(x_i^t))_{i=1}^M \), where, similar to PPDS, \( x_i^t \) is the iterate at which \( \nabla f_i \) was last evaluated. Let \( i_t \) be sampled from the index set \( \{1, \ldots, M\} \). The update of SAGA is then given by

\[
g_t = \nabla f_{i_t}(x_t) - \nabla f_{i_t}(x_t) + \frac{1}{M} \sum_{i=1}^M \nabla f_i(x_i) \\
x_{t+1} = x_t - \eta g_t.
\]

To recover SAGA from PPDS, we set \( A_t = B_t = J \). This ensures \( y_i^t = (1/M) \sum_{i=1}^M \nabla f_i(x_i) \), and thus, \( y_i^{t+\frac{1}{2}} \) is exactly updated as in (1) when \( i \) is active. Specifically, if exactly one node is sampled at each iteration, (PPDS) with stepsize \( \eta \) and \( A_t = B_t = J \) is equivalent to SAGA with stepsize \( \eta/M \). If multiple workers are active in the same time slot, we get a minibatch version of SAGA.

III. LINEAR CONVERGENCE OF PPDS

In this section, we present convergence guarantees of PPDS for strongly convex functions over a random network model. Concretely, we make the following standard convexity/smoothness assumption on the objective functions:

**Assumption 1:** All the individual \( f_i^t \)'s are \( L \)-smooth and convex; the global function \( f \) is \( \mu \)-strongly convex.

Thanks to the strong convexity of \( f \), we know there exists a unique solution of (P), which we will denote by \( x_\star \). Moreover, we model \( (\mathcal{V}_t)_{t\in\mathbb{N}}, \ (A_t)_{t\in\mathbb{N}}, \) and \( (B_t)_{t\in\mathbb{N}} \) as random variables satisfying the following.

**Assumption 2:** The random variables \( (\mathcal{V}_t, A_t, B_t)_{t\in\mathbb{N}} \) are temporally independent and identically distributed.

Assumption 2 is actually only needed to provide the contractions of Lemmas 3 and 6. Hence, it could be weakened accordingly. We chose to keep it as such for ease of reading and for consistency with the literature.

A. General Case

First, we present our linear convergence result under rather weak assumptions on communications (essentially that the information can flow all over the network) and device sampling (each node is sampled with positive probability).

**Assumption 3:** The mixing matrices \( (A_t)_{t\in\mathbb{N}} \) and \( (B_t)_{t\in\mathbb{N}} \) have the following properties.

a) For all \( t \in \mathbb{N} \), \( A_t \) is row-stochastic and \( B_t \) is column-stochastic.

b) Both \( A := \mathbb{E}[A_t] \) and \( B := \mathbb{E}[B_t] \) are primitive.

c) There exists \( \nu > 0 \) such that \( a_{ii}^t \geq \nu \) and \( b_{ii}^t \geq \nu \) for all \( i \in \mathcal{V}, t \in \mathbb{N} \).

**Assumption 4:** Every node is sampled with positive probability, i.e., \( p_t := \mathbb{P}(i \in \mathcal{V}_t) > 0 \) for all \( i \in \mathcal{V} \).

It is straightforward to verify that Assumptions 2–4 are fulfilled in all the aforementioned examples. In particular, in Section II-C3, the primitivity of matrices \( A \) and \( B \) are ensured by the strong connectivity of the underlying graph \( G \) since \( A_{ij} > 0 \) if and only if \( (j, i) \in E \) if and only if \( B_{ij} > 0 \). On the other hand, Assumption 3 c posits that at each iteration, each node maintains a fraction of its previous states. This rules out the counterexamples in which the states of the active nodes are always overwritten by those of the inactive states. Under these fairly weak assumptions, we manage to prove the convergence of PPDS as stated in the following theorem.

**Theorem 1:** Let Assumptions 1–4 hold. If (PPDS) is run with a sufficiently small stepsize \( \eta > 0 \), then the following holds.

a) \( x^t \) converges almost surely to the solution \( x_\star \).

b) The expected squared distance between the iterate and the solution \( \mathbb{E}||x^t - x_\star||^2 \) vanishes geometrically.

Theorem 1 shows that the nice properties of gradient tracking and variance-reduced methods are also preserved by our algorithm: It converges with constant stepsize and enjoys a linear convergence rate as centralized gradient descent. Therefore, our method effectively reduces the variances of the noises induced by both sampling and communication.

We note that the assumptions for this result are similar to the ones for G-push–pull in [15], except that 1) we do not put additional restrictions on the coefficients of the gossip matrix (unlike [15, eq. (24a)]); and 2) we allow for a device sampling strategy that can be independent or correlated with the gossiping step.

B. Case of Doubly Stochastic Matrices.

Due to the generality of the result, Theorem 1 only describes the qualitative behavior of the algorithm. To derive a convergence rate with explicit dependence on the problem parameters, we focus on the specific situation where the mixing matrices are...
doubly stochastic and the active devices are sampled uniformly at random. Formally, we make the following assumptions.

**Assumption 3’:** For all \( t \in \mathbb{N} \), both \( A_t \) and \( B_t \) are doubly stochastic. Moreover, we have the inequality

\[
\lambda := \max(\rho(E[A_t(1 - J)A_t]), \rho(E[B_t(1 - J)B_t])) < 1.
\]

**Assumption 4’:** \( \mathcal{V}_t \) is of fixed size \( S \) and is sampled uniformly from all the subsets of this size.

The bistochastic assumption is for example verified when the communication matrices are the Metropolis matrices of the subgraphs of the active nodes (covered in Section II-C2). However, it is still possible to have communications between active and inactive nodes under this assumption, as demonstrated by the SAGA example (see Section II-C4). On the technical side, the bistochasticity of the matrices and the condition \( \lambda < 1 \) allow us to derive a per-step contraction of the variance of the nodes’ variables (see Lemma 3). Using uniform sampling further facilitates the analysis and makes the final expression much more concise. Building upon these, the next theorem states the stepsize condition and the convergence rate of PPDS when these assumptions are fulfilled.

**Theorem 2:** Let Assumptions 1, 2, 3’, and 4’ hold. If (PPDS) is run with stepsize

\[
\eta \leq \min \left( \frac{(1 - \lambda)^2}{4A L S}, \frac{(1 - \lambda)^2}{2304L S^2}, \frac{1}{576L} \sqrt{\frac{M}{S}} \right)
\]

then the expected squared distance \( \mathbb{E}[||x_t^i - x_t||^2] \) vanishes geometrically in \( O(\gamma^t) \) with \( \gamma = \max(1 - \frac{\mu}{2M}, 1 - \frac{\lambda^2}{4A M}) \). In particular, it takes

\[
O\left( \left( \frac{L}{\mu} \sqrt{\frac{M}{S}} \left( \frac{1}{(1 - \lambda)^2} + \frac{1}{\lambda^2} \right) \right) \log \left( \frac{1}{\xi} \right) \right)
\]

iterations to achieve \( \xi \) accuracy when \( \eta \) is suitably tuned.

Theorem 2 indicates that a larger stepsizes can (and should) be taken for smaller sample size, if all the other parameters are fixed. Intuitively, this is because at each iteration, fewer gradients enter the network, and thus, these gradients can be used with a larger weight.

In terms of dependence on problem parameters, the linear dependence on the condition number \( L/\mu \) matches that of standard gradient descent, whereas the \( 1/(1 - \lambda)^2 \) dependence on the mixing parameter is common in the literature of gradient tracking [24] but has been further improved recently by Koloskova et al. [33] in the case of the single fixed mixing matrix. Regarding the effect of device sampling, although the complexity in terms of iterations is degraded by \( \sqrt{M/S} \) compared to asynchronous push–pull without device sampling (i.e., \( S = M \)), the complexity in a number of computed gradients is actually improved. To see this, we multiply (3) by \( S \) and verify that the resulting quantity indeed decreases when \( S \) gets smaller.

Nonetheless, device sampling may also affect the connectivity of the network, and thus, \( \lambda \) if the communication matrices are chosen according to the sampled devices \( \mathcal{V}_t \) (for instance, in Section II-C3). Therefore, unlike in the centralized case, sampling with variance reduction is not always guaranteed to converge faster here. Rather, there is a communication–computation tradeoff that involves both the choice of the sampling size \( S \) and the mixing matrices \( A_t, B_t \).

**IV. CONVERGENCE ANALYSIS**

In this section, we outline the proofs of Theorems 1 and 2. To begin, let us define

\[
g_t^i = y_t^i + \nabla f_i(x_t^i) - \nabla f_i(z_t^i)
\]

as the gradient estimator of node \( i \) at iteration \( t \) so that \( y_t^{i+1} = g_t^i \) if and only if \( i \in \mathcal{V}_t \), and \( y_t^{i+1} = y_t^i \) otherwise. With the mass preservation property of column-stochastic matrices and the definition of \( z_t^i \), we have immediately the following lemma.

**Lemma 1:** Suppose that the matrices \( (B_t)_{t \in \mathbb{N}} \) are column-stochastic. It holds that

\[
\sum_{i=1}^M y_t^i = \sum_{i=1}^M \nabla f_i(z_t^i), \quad \sum_{i=1}^M g_t^i = \sum_{i=1}^M \nabla f_i(x_t^i).
\]

Therefore, if the iterates move in the direction \( -\sum_{i=1}^M g_t^i \), we can expect convergence of the algorithm. This idea is crucial for our proof.

Another important step in the analysis is to establish that the nodes’ decisions variables converge to a consensus. For this, let us write \( x_t = 1^T X_t/M \) for the average of these variables. Similarly, we also use the notation \( y_t = 1^T Y_t/M \).

Finally, we would like to highlight that the expectation \( \mathbb{E} \) is taken over the randomness induced by both sampling and communication. We define \( (F_t)_{t \in \mathbb{N}} \) as the natural filtration associated to the sequence \( (X_t)_{t \in \mathbb{N}} \) so that \( (\mathbb{V}_t,s,A_t,b_s)\}_{1 \leq s \leq t-1} \) is \( F_t \) measurable while \( (\mathbb{V}_t,A_t,B_t) \) is not. For simplicity, we write \( \mathbb{E}_t \) for the expectation conditioned on the history up to time \( t \), i.e., \( \mathbb{E}_t[.] = \mathbb{E}[.|F_t] = \mathbb{E}[|(\mathbb{V}_s,A_s,B_s)\}_{1 \leq s \leq t-1}] \).

**A. Analysis With Doubly Stochastic Matrices**

As a warm-up, we first establish the convergence of the algorithm in the simpler case where both Assumption 3’ and Assumption 4’ hold. This allows us to highlight our proof strategy without having to deal with the additional difficulties caused by the fact of having asymmetric communications. Following previous works that analyze gradient tracking and variance-reduced methods, the essential idea of our proof is to derive a system of inequalities for the following quantities:

\[
d_t = \mathbb{E}[||x_t - x_t||^2], \quad e_t = \mathbb{E}[f(x_t) - f(x_t)]
\]

\[
\rho_t = \mathbb{E}[||x_t - 1x_t||^2], \quad \zeta_t = \mathbb{E}[||y_t - 1y_t||^2]
\]

\[
\psi_t = \sum_{i=1}^M \mathbb{E}[||\nabla f_i(x_t^i) - \nabla f_i(x_t^i)||^2].
\]

Here, \( d_t \) and \( e_t \) measure the performance of the averaged iterate; \( \rho_t \) and \( \zeta_t \) measure the variances of the two variables of the agents; and \( \psi_t \) measures the quality of the control variates and is standard in the analysis of variance-reduced algorithms [34].
[35]. The following proposition bounds these quantities by a linear combination of their previous values.

**Proposition 1:** Let \( r_t = [d_t, \rho_t, \zeta_t, \psi_t]^\top \). Under Assumptions 1, 2, 3', and 4', we have
\[
    r_{t+1} \leq Qr_t + e_t h
\]
where the entries of \( Q \) and \( h \) are given by
\[
    Q = \begin{bmatrix}
        1 - \frac{\eta S}{2M} & \frac{\eta L S}{M(1-\lambda)} + \frac{10\eta^2 L^2 S^2}{M} & \frac{2\eta^2 S^2}{M} & \frac{4\eta^2 S^2}{M} \\
        0 & \frac{20\eta^2 L^2 S}{M(1-\lambda)} & \frac{4\eta^2 S^2}{M} & \frac{8\eta^2 S^2}{M} \\
        0 & \frac{M(1-\lambda)}{M(1-\lambda)} & \frac{1}{1-\lambda} & \frac{1}{1-\lambda} \\
        0 & \frac{2L^2 S^2}{M} & 0 & 1 - \frac{S}{M}
    \end{bmatrix}
\]
\[
h = \left[ -\frac{\eta S}{M} + \frac{20\eta^2 L^2 S^2}{M^2}, \frac{4\eta^2 L^2 S}{1-\lambda}, \frac{16LS}{1-\lambda}, \frac{4LS}{1-\lambda} \right]^\top.
\]

To prove Proposition 1, we start by presenting a series of technical lemmas that are useful for this purpose. First, in order to deal with device sampling, we observe that \( G_t = [g_1^t, \ldots, g_M^t]^\top \) plays an important role since \( D_t Y_{t+1/2} = D_t G_t \). With the uniform sampling of Assumption 4', we obtain the following lemma.

**Lemma 2:** Let Assumption 2 and 4' hold. Then, the following holds.

\[
a) \quad \mathbb{E}_t[\|D_t Y_{t+1/2}\|^2] = \frac{S}{M} \sum_{i=1}^M \|g_i^t\|^2.
\]

**Proof:** a) Note that \( D_t Y_{t+1/2} = \sum_{i \in V_t} y_i^t = \sum_{i \in V_t} g_i^t. \)

Therefore
\[
\mathbb{E}_t[\|D_t Y_{t+1/2}\|^2] = \mathbb{E}_t \left[ \sum_{i \in V_t} g_i^t \right] = \mathbb{E}_t \left[ \sum_{i \in V_t} g_i^t \right] = \sum_{i=1}^M g_i^t \mathbb{E}_t[1_{i \in V_t}] = \frac{S}{M} \sum_{i=1}^M g_i^t.
\]
We can put \( g_i^t \) outside the expectation since it is \( F_t \)-measurable.

b) Similarly, we have
\[
\mathbb{E}_t[\|Y_{t+1/2}\|^2] = \mathbb{E}_t \left[ \sum_{i \in V_t} g_i^t \right] \leq \mathbb{E}_t \left[ \sum_{i \in V_t} g_i^t \right] = \frac{S}{M} \sum_{i=1}^M g_i^t.
\]

To control the distance to consensus, we use the following lemma that shows a contraction property of the mixing matrices.

**Lemma 3:** Let Assumption 2 and 3' hold. Then, the following holds.

\[
a) \quad \mathbb{E}_t[\|A_t X_t - 1\xi_t^t\|^2] \leq \lambda \|X_t - 1\xi_t^t\|^2.
\]

**Proof:** Since \( A_t \) is doubly stochastic, we can write
\[
\mathbb{E}_t[\|A_t X_t - 1\xi_t^t\|^2]
\]
\[
= \mathbb{E}_t[\|(I-J)A_t(X_t - 1\xi_t^t)\|^2]
\]
\[
= \mathbb{E}_t[\tr((X_t - 1\xi_t^t)^\top (I-J)A_t^\top (X_t - 1\xi_t^t))]
\]
\[
= \tr((X_t - 1\xi_t^t)^\top \mathbb{E}_t[A_t^\top (I-J)A_t](X_t - 1\xi_t^t))
\]
\[
\leq \rho(\mathbb{E}_t[1\xi_t^\top (I-J)A_t])\|X_t - 1\xi_t^t\|^2.
\]

Under Assumption 2, we have \( \mathbb{E}_t[1\xi_t^\top (I-J)A_t] = \mathbb{E}[1\xi_1^\top (I-J)A_1] \) and \( a \) follows immediately given that \( \rho(\mathbb{E}_t[1\xi_t^\top (I-J)A_t]) \leq \lambda. \) Property \( b \) is proved in the same way.

Finally, we can use the smoothness of the objective functions and the optimality conditions to bound the expected squared norm of \( G_t \) and gradients differences by the quantities introduced in (4).

**Lemma 4:** Let Assumption 1 hold and \( (B_t)_{t \in \mathbb{N}} \) be column-stochastic. We have the following.

a) \( \mathbb{E}[\|\nabla F(X_t) - \nabla F(1\times_s)^t\|^2] \leq 2\rho_t \| \theta_t + 4ML\epsilon_t \|.
\]

b) \( \mathbb{E}[\|\nabla F(X_t) - \nabla F(Z_t)^t\|^2] \leq 4\rho_t \| \theta_t + 8ML\epsilon_t + 2\psi_t \|.
\]

c) \( \mathbb{E}[\|G_t\|^2] \leq 10\rho_t \| \theta_t + 20ML\epsilon_t + 4\psi_t + 2\xi_t \|
\]

**Proof:** See Appendix A.

We are now ready to prove Proposition 1 by leveraging the above lemmas.

**Proof of Proposition 1:** In the following, we bound the four quantities in question respectively.

**Bounding \( d_{t+1} \):** We develop
\[
\mathbb{E}[\|\x_t - x_s\|^2] = \mathbb{E}[\|\x_t - \frac{\eta}{M}\|D_t Y_{t+1/2} - x_s\|^2]
\]
\[
= \mathbb{E}[\|\x_t - x_s\|^2 - \frac{2\eta}{M}\langle \x_t - x_s, 1\|D_t Y_{t+1/2}\| \rangle]
\]
\[
+ \frac{\eta^2}{M^2}\|1\|D_t Y_{t+1/2}\|^2.
\]

Using Lemmas 1 and 2 and Assumption 1, we get
\[
\mathbb{E}[\|\x_t - y_s, 1\|D_t Y_{t+1/2}\|]
\]
\[
= \langle \x_t - x_s, \frac{S}{M} \sum_{i=1}^M \nabla f_i(x_i) \rangle
\]
\[
= \sum_{i=1}^M \left( f_i(\x_t) - f_i(x_s) - \frac{L}{2}\|\x_t - x_s\|^2 + f_i(x_s) - f_i(x_s) \right)
\]
\[
= S(f(\x_t) - f(x_s)) - \frac{LS}{2M}\|\x_t - 1\xi_t^t\|^2
\]
\[
\geq \frac{S}{2}(f(\x_t) - f(x_s)) + \frac{\mu S}{4}\|\x_t - x_s\|^2 - \frac{LS}{2M}\|\x_t - 1\xi_t^t\|^2.
\]

In the last line, we have used the fact that \( f(x) - f(x_s) \geq (\mu/2)\|x - x_s\|^2 \) for every \( x \in \mathbb{R}^d \) since \( f \) is strongly convex. As for the last term of (6), we resort to Lemmas 2b and 4c. This
gives
\[\mathbb{E}[\|\mathbf{1}^T D_t Y_{t+\frac{1}{2}}\|^2] \leq \frac{S^2}{M} (10L^2 \rho_t + 20M \rho_t + 4 \psi_t + 2 \zeta_t).\]
Combining the above inequalities, we get
\[
d_{t+1} \leq \left(1 - \frac{\eta \mu S}{2M}\right) d_t + \frac{\eta LS}{M^2} + \frac{10n^2 L^2 S^2}{M^3} \rho_t + \frac{2n^2 S^2}{M^2} \zeta_t + \frac{4n^2 S^2}{M^3} \psi_t - \left(\frac{\eta S}{M} - \frac{20n^2 L^2 S^2}{M^2}\right) \epsilon_t.
\]
Bounding \(\rho_{t+1,}\) in the inequality \(\|a + b\|^2 \leq (1 + \delta)\|a\|^2 + (1 + 1/\delta)\|b\|^2\), choosing \(\delta = (1 - \lambda)/2\) gives
\[
\|a + b\|^2 \leq \frac{1 + \lambda}{2\lambda} \|a\|^2 + \frac{1 + \lambda}{1 - \lambda} \|b\|^2.
\]
Since \(A_t\) is doubly stochastic and hence column-stochastic, it holds \(J A_t = J\). We then have
\[
\mathbb{E}_t[\|X_t - 1 \mathbf{1}^T Y_{t+\frac{1}{2}}\|^2] = \mathbb{E}_t[\|A_t X_t - \eta A_t D_t Y_{t+\frac{1}{2}} - (1 - \eta J D_t Y_{t+\frac{1}{2}})\|^2] \leq \frac{1 + \lambda}{2\lambda} \mathbb{E}_t[\|A_t X_t - 1 \mathbf{1}^T Y_{t+\frac{1}{2}}\|^2] + \frac{1 + \lambda}{1 - \lambda} \mathbb{E}_t[\|D_t Y_{t+\frac{1}{2}}\|^2].
\]
Using Lemma 3a, the first term can be bounded by \((1 + \lambda)\|X_t - 1 \mathbf{1}^T Y_{t+\frac{1}{2}}\|^2/2\). The same does not apply to the second term as \(A_t\) and \(D_t\) are not independent. Nonetheless, with the bistochasticity of \(A_t\), we can still write
\[
\mathbb{E}_t[\|A_t D_t Y_{t+\frac{1}{2}} - J D_t Y_{t+\frac{1}{2}}\|^2] \leq \mathbb{E}_t[\|D_t Y_{t+\frac{1}{2}}\|^2] = \frac{S}{M} \sum_{i=1}^{M} \|g_i^t\|^2.
\]
With Lemma 4c, taking total expectation in (9) then gives
\[
\rho_{t+1} \leq \left(1 + \frac{\lambda}{2}\right) \rho_t + \frac{1 + \lambda}{1 - \lambda} \frac{\eta^2 S}{M(1 - \lambda)} (10L^2 \rho_t + 20M \rho_t + 4 \psi_t + 2 \zeta_t) \leq \left(1 + \frac{\lambda}{2}\right) \rho_t + \frac{20n^2 L^2 S^2}{M^2} \frac{1 + \lambda}{1 - \lambda} \frac{\eta^2 S}{M(1 - \lambda)} \frac{\rho_t}{\epsilon_t} + \frac{8n^2 S^2}{M^2} \frac{1 + \lambda}{1 - \lambda} \psi_t + \frac{4n^2 S^2}{M} \frac{1 + \lambda}{1 - \lambda} \frac{\eta^2 S}{M} \epsilon_t.
\]
Bounding \(\zeta_{t+1}\). Similar to the above, using (8) and the bistochasticity of \(B_t\), we obtain
\[
\|Y_{t+1} - 1 \mathbf{1}^T Y_{t+1}\|^2 = \|B_t Y_t - B_t D_t (\nabla F(X_t) - \nabla F(Z_t)) - (1 - \eta J D_t (\nabla F(X_t) - \nabla F(Z_t)))\|^2 \leq \frac{1 + \lambda}{2\lambda} \|B_t Y_t - 1 \mathbf{1}^T Y_t\|^2 + \frac{1 + \lambda}{1 - \lambda} \|D_t (\nabla F(X_t) - \nabla F(Z_t))\|^2.
\]
Taking expectation in (10) and applying Lemmas 3b and 4b yields
\[
\zeta_{t+1} \leq \left(1 + \frac{\lambda}{2}\right) \zeta_t + \frac{1 + \lambda}{1 - \lambda} \frac{S}{M} \left(4L^2 \rho_t + 8M \rho_t + 2 \psi_t\right) \leq \left(1 + \frac{\lambda}{2}\right) \zeta_t + \frac{8L^2 S}{M(1 - \lambda)} \rho_t + \frac{4S}{M(1 - \lambda)} \psi_t + \frac{16LS}{1 - \lambda} \epsilon_t.
\]
Bounding \(\psi_{t+1}\). Note that by the update rule of \(z_i^t\), we have
\[
\mathbb{E}_t[\|\nabla f_i(x_i^{t+1}) - \nabla f_i(x_i)\|^2] = \left(1 - \frac{S}{M}\right) \|\nabla f_i(x_i^t) - \nabla f_i(x_i)\|^2 + \frac{S}{M} \|\nabla f_i(x_i^{t+1}) - \nabla f_i(x_i)\|^2.
\]
Summing from \(i = 1\) to \(M\), applying Lemma 4a and taking total expectation, we get
\[
\psi_{t+1} \leq \left(1 - \frac{S}{M}\right) \psi_t + \frac{S}{M} (2L^2 \rho_t + 4M \rho_t) \leq \left(1 - \frac{S}{M}\right) \psi_t + \frac{2L^2 S}{M} \rho_t + 4LS \epsilon_t.
\]

**Conclusion.** Putting all together, we get exactly (5).

From the linear system of inequalities (5), there are multiple ways to derive the linear convergence of the algorithm. To obtain the explicit convergence rate and stepsize condition presented in Theorem 2, we construct a suitable Lyapunov function, which is a linear combination of \(d_t, \rho_t, \zeta_t, \) and \(\psi_t\) with positive coefficients, and prove that this function decreases geometrically at each iteration.

**Proof of Theorem 2:** Let us consider the vector
\[
\omega = \begin{bmatrix} 1 & \sqrt{5(1 - \lambda)} & \eta(1 - \lambda) & \eta \end{bmatrix}^T
\]
and \(\gamma\) as defined in Theorem 2, it can be verified that Proposition 1 implies
\[
\omega^T r_{t+1} \leq \gamma \omega^T r_t
\]
whenever stepsize condition (2) is satisfied. This means \(\omega^T r_t\) converges geometrically in \(O(\gamma^t)\). To conclude, we use the inequality
\[
\mathbb{E}[\|x_i^t - x_i\|^2] \leq \mathbb{E}[\|x_i^t - x_i\|^2] + 2\mathbb{E}[\|x_i - x_i\|^2] \leq 2 \rho_t + 2d_t.
\]
Detailed computations for proving (11) are provided in Appendix B.

**B. Analysis for the General Case**

Under our weakest set of assumptions (Assumptions 3 and 4), the mixing matrices do not provide a contraction toward a consensus at each iteration. Nevertheless, the primitivity of the

[^1]: Without loss of generality, we assume \(\lambda > 0\). Otherwise, the first term in the inequalities are always 0 and we can simply take \(\delta = 1\). The same remark applies to the analysis in Section IV-B.
mixing matrices in expectation enables us to show that after a number of gossip steps \(l\) (implicitly defined), some sort of contraction happens for both matrices sequences but with respect to a time-varying weighted average instead of a uniform one.

This has direct consequences on our proof technique since the linear system of equations developed previously has to be modified and in particular extended to track \(l\) successive iterations. With this augmentation, the proof techniques developed before do not hold anymore and we resort to analyzing the spectral radius of the recurrence matrix by perturbation theory arguments when the stepsize is small.

These two points significantly complicate the convergence proof of the method and constitute the main technical contributions of the article.

1) **Multistep Contraction:** To establish the multistep contraction brought by the mixing matrices, we first leverage the primitivity assumption on \(A = \mathbb{E}[A_1]\) and \(B = \mathbb{E}[B_1]\) to show that inequalities similar to the one in Assumption 3' hold when we consider the product of successive matrices, which we abbreviate as

\[
A_{t:s} = A_t A_{t-1} \ldots A_s, \quad B_{t:s} = B_t B_{t-1} \ldots B_s.
\]

The following lemma generalizes Assumption 3' and is useful for deriving inequalities in the form of Lemma 3.

**Lemma 5:** Let Assumptions 2 and 3 hold. Then, there exists an integer \(l\) such that

\[
\rho(\mathbb{E}[A_{1:t+1}^\top (I - J) A_{1:t+1}]) < 1
\]

\[
\rho(\mathbb{E}[(I - J)^\top B_{1:t+1} B_{1:t+1} (I - J)]) < 1.
\]

**Proof:** We will write \(\|W\|\) and \(\|W\|_F\), respectively, for the spectral norm and the Frobenius norm of a matrix \(W\). Lemma 5 is an immediate result of [13, Prop. 2], which states that \(\mathbb{E}[\|I - J\| A_{1:t+1}]^2\|_F\) converges to 0 at a geometric rate. We can thus set \(l\) sufficiently large so that \(\mathbb{E}[\|I - J\| A_{1:t+1}]^2\|_F < 1\), and the first inequality then follows from that

\[
\rho(\mathbb{E}[A_{1:t+1}^\top (I - J) A_{1:t+1}]) \leq \rho(\mathbb{E}[A_{1:t+1}^\top (I - J) A_{1:t+1}])
\]

\[
= \mathbb{E}[\|I - J\| A_{1:t+1}]^2\|_F
\]

\[
\leq \mathbb{E}[\|I - J\| A_{1:t+1}]^2\|_F^2
\]

where we have used the convexity of the spectral radius function \(\rho\) and the fact that the spectral norm of a matrix is bounded from above by its Frobenius norm.

For the second inequality, we observe that the matrices \((B_{t}')_{t \in \mathbb{N}}\) have the same assumptions as \((A_t)_{t \in \mathbb{N}}\). Moreover

\[
\rho(\mathbb{E}[(I - J)^\top B_{1:t+1}^\top B_{1:t+1} (I - J)])
\]

\[
\leq \mathbb{E}[\rho((I - J)^\top B_{1:t+1}^\top B_{1:t+1} (I - J))]
\]

\[
= \mathbb{E}[(B_{1:t+1}^\top (I - J)]^2\|_F
\]

\[
= \mathbb{E}[(I - J) B_{1:t+1}^\top (I - J)]^2\|_F
\]

Hence, the same argument applies.

Another important challenge toward proving a result in the spirit of Lemma 3 is that the matrices \((A_t)_{t \in \mathbb{N}}\) (resp. \((B_t)_{t \in \mathbb{N}}\)) do not have a fixed left (resp. right) Perron vector, and as a consequence, there are not predetermined values that the variables should converge to after the mixing matrices are applied. To overcome this difficulty, we introduce two sequences of random vectors \((\mathbf{v}_t)_{t \in \mathbb{N}}\) and \((\mathbf{u}_t)_{t \leq T}\).

The sequence \((\mathbf{u}_t)_{t \leq T}\) is defined in a time-reversed manner and mimics the absolute probability sequence [36], [37] that can be defined for \((A_t)_{t \in \mathbb{N}}\). However, the above construction gives an explicit expression of \(\mathbf{u}_t\), which turns out to be useful for our proof. Also, notice that the value of \(\mathbf{u}_t\) is dependent on the choice of \(T\) though this is implicit from the notation.

Since \((B_t)_{t \in \mathbb{N}}\) are column-stochastic and \((A_t)_{t \in \mathbb{N}}\) are row-stochastic, one deduces immediately that both \((\mathbf{v}_t)_{t \in \mathbb{N}}\) and \((\mathbf{u}_t)_{t \leq T}\) are sequences of probability vectors. Moreover, under Assumption 2, we have \(\mathbb{E}[\mathbf{u}_t] = \mathbb{E}[\mathbf{u}_{t+1}] A_t = \mathbb{E}[\mathbf{u}_{t+1}] A_t\). By induction, we then get

\[
\mathbb{E}[\mathbf{u}_t] = \pi_A \forall t \in \{1, \ldots, T\}.
\]

In the remainder of the section, we will take \(l \geq 0\) such that the inequalities of Lemma 5 are satisfied and define

\[
\lambda = \max(\rho(\mathbb{E}[A_{1:t+1}^\top (I - J) A_{1:t+1}]),)
\]

\[
\rho(\mathbb{E}[(I - J)^\top B_{1:t+1}^\top B_{1:t+1} (I - J)])
\]

so that \(\lambda < 1\). The multistep contraction property is stated as follows.

**Lemma 6:** Let Assumptions 2 and 3 hold. Take \(l\) as in Lemma 5 and \(\lambda\) from (13). Then, the following holds.

\[a) \quad \mathbb{E}[\|I - J\| A_{1:t+1} X_t^2] \leq \lambda \|X_t - I X_t\|^2.\]

\[b) \quad \mathbb{E}[\|B_{1:t+1}^\top (I - \mathbf{v}_t 1^\top) Y_t\|^2] \leq \lambda \|(I - \mathbf{v}_t 1^\top) Y_t\|^2.\]

**Proof:** The lemma is proved exactly in the same way as Lemma 3. Just notice that

\[
(I - J) A_{1:t+1} X_t = (I - J) A_{1:t+1} X_t
\]

\[
= (I - J) A_{1:t+1} X_t - \mathbf{1} X_t^2
\]

since \(A_{1:t+1}\) is row-stochastic. On the other hand

\[
B_{1:t+1}^\top (I - \mathbf{v}_t 1^\top) Y_t = B_{1:t+1}^\top (I - J) (I - \mathbf{v}_t 1^\top) Y_t,
\]

since \(\mathbf{v}_t\) is a probability vector.

2) **Linear System of Inequalities:** As in Section IV-A, the proof for Theorem 1 also relies on the derivation of a linear system of inequalities. Nevertheless, since there is a contraction only every \(l + 1\) steps, we need to take into account the values of relevant quantities for \(l + 1\) consecutive iterations and the system becomes \(l + 1\) times larger. Given that the mixing matrices are no longer doubly stochastic, the variables that come into play also need to be modified accordingly. We consider the following quantities:

\[
d_t = \mathbb{E}[(\mathbf{u}_t^\top X_t - \mathbf{x}_t)^2], \quad e_t = \mathbb{E}[f(\mathbf{x}_t) - f(\mathbf{x}_s)]
\]

\[
p_t = \mathbb{E}[(\mathbf{u}_t^\top X_t - \mathbf{1} X_t)^2], \quad \zeta_t = \mathbb{E}[\|Y_t - \mathbf{v}_t 1^\top Y_t\|^2]
\]

\[
\psi_t = \mathbb{E}[\|\nabla F(X_t) - \nabla F(Z_t)\|^2].
\]
Compared to (4), we define $d'_t$ because we no longer have $\mathbb{1}^\top A_t X_t = \mathbb{1}^\top X_t$, while it holds $u_{t+1}^\top A_t X_t = u_t X_t$. The definition of $c'_t$ is consistent with Lemma 6 b. Finally, we also replace $\psi_t$ by $\psi'_t$ for technical reasons. Note that the value of $d'_t$ depends on $T$ since its definition involves $u_t$.

The following two lemmas collect several inequalities that will be useful for our proof.

**Lemma 7:** It holds the following.

a) $\|D_t\| \leq 1$.

b) $\|u_t^\top X_t - \bar{x}_t\|^2 \leq \|X_t - \mathbb{1}\bar{x}_t\|^2$.

c) The spectral norm of a row- or column-stochastic matrix of size $M \times M$ is not larger than $\sqrt{M}$.

**Proof:** a) is trivial and c) can be proven by using the fact that the spectral norm of a matrix is bounded by its Frobenius norm. As for b), since $u_t$ is a probability vector

$$\|u_t^\top X_t - \bar{x}_t\|^2 = \sum_{i=1}^M u_t^i (x_t^i - \bar{x}_t)^2 \leq \sum_{i=1}^M u_t^i (x_t^i - \bar{x}_t)^2 \leq \sum_{i=1}^M |x_t^i - \bar{x}_t|^2 = \|X_t - \mathbb{1}\bar{x}_t\|^2.$$

In the above, we have used the notation $u_t = (u_t^i)_{i \in V}$. □

**Lemma 8:** Let Assumption 1 hold and $(B_t)_{t \in \mathbb{N}}$ be column-stochastic. We have the following.

a) $\|\mathbb{1}^\top X_t - \mathbb{1}^\top \bar{x}_t\|^2 \leq 2c'_t + (4M + 4)\psi'_t$.

b) $\|E_t^i\|^2 \leq 4M L^2 \rho_t + (8M + 8)\psi'_t + 4c'_t + 8M^2 L\rho_t$.

**Proof:** See Appendix C. □

Since the sampling is not uniform, Lemma 2 does not hold anymore and we need to approximate $G_t$ by $\mathbb{1}^\top G_t$. When deriving the descent inequality. Given the definition of $d'_t$ and the fact that the nodes are sampled, we say that the effective stepsize at time $t$ is $\eta \alpha_t$ with

$$\alpha_t = u_t^\top D_t^\top V_t.$$

The following lemma controls $\mathbb{E}[\alpha_t \chi_t]$ for any real-valued nonnegative random variable $\chi_t$ that is $F_t$-measurable.

**Lemma 9:** Let Assumptions 2–4 hold. We define $p = \min_{v \in V} \rho_t$, $\pi_A = \min_{v \in V} [\pi_A]$, and $\omega = \pi_A \nu_t$. Then, $\omega > 0$, and for any $F_t$-measurable real-valued nonnegative random variable $\chi_t$, we have

$$\omega \mathbb{E}[\chi_t] \leq \mathbb{E}[\alpha_t \chi_t] \leq \mathbb{E}[\chi_t].$$

**Proof:** See Appendix D. □

We are now ready to state and prove the linear system of inequalities in question. We denote by $P \otimes Q$ the Kronecker product of two matrices $P$ and $Q$, and write $E_t^{ij} = E_t^{i,j}$ for the matrix of size $k \times k$ that has a single nonzero entry with value 1 at position $(i, j)$.

**Proposition 2:** For $T > l$ and $t \in \{1, ..., T - l\}$, let $\alpha$ be defined as in Lemma 9 and $r'_t \in \mathbb{R}^{(l+1)\times (l+1)}$ be defined by

$$r'_t = [d'_{t+l} \ldots d'_t \rho_{t+l} \ldots \rho_t \psi'_t \psi'_t \psi'_t \psi'_t \psi'_t \psi'_t \psi'_t \psi'_t \psi'_t \psi'_t].$$

We also define $W_1, W_2 \in \mathbb{R}^{(l+1)\times (l+1)}$ as

$$W_1 = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 1 & \cdots & \cdots & \cdots & 1 \\ 0 & \cdots & 0 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

Then, under Assumptions 1–4, if PPDS is run with $\eta \leq \omega/(16ML)$, we have

$$r_{t+1}' \leq (Q_0 + \eta Q_e) r'_t$$

where

$$Q_0 = I_4 \otimes W_1 + c_1 E_{4,3}^{1} \otimes W_2 + \frac{1+\lambda}{2} (E_{1,1}^{4} + E_{3,3}^{4}) \otimes E_{1,1}^{l+1}$$

and

$$Q_e = \begin{bmatrix} -c_1 & c_2 & c_3 & c_4 \\ 0 & 0 & 0 & 0 \\ c_9 & 0 & c_{11} & c_{12} \\ 0 & 0 & 0 & 0 \end{bmatrix} \otimes E_{1,1}^{l+1} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \otimes W_2$$

are defined with positive constants $(c_k)_{1 \leq k \leq 13}$ that are entirely determined by $\mu, L, M, \lambda, l, p$, and $\omega$.

**Proof:** We will make use of the inequality

$$e_t \leq L^2 (d'_t + \rho_t).$$

This comes from the simple fact that

$$f(\bar{x}_t) - f(x_t) \leq \frac{L^2}{2} \|\bar{x}_t - x_t\|^2 \leq L^2 (\|u_t^\top X_t - \bar{x}_t\|^2 + \|u_t^\top X_t - x_t\|^2) \leq L^2 (\|X_t - \mathbb{1}\bar{x}_t\|^2 + \|u_t^\top X_t - x_t\|^2).$$

Also, notice that $\|D_t Y_t \frac{l}{2}\| = \|D_t G_t\| \leq \|D_t\| \|G_t\| \leq \|G_t\|$. □

Now, let us fix $t \in \{l + 1, \ldots, T - l\}$. We bound $\rho_{t+1}$, $\psi'_t$, $\psi'_t$, and $d'_t$ in terms of the previous values of these same variables.

**Bounding $d'_{t+1}$:** We decompose

$$\|u_{t+1}^\top X_{t+1} - x_t\|^2$$

$$\|u_t^\top A_t X_t - \eta u_t^\top A_t D_t Y_t \frac{l}{2} - x_t\|^2$$

$$= \|u_t^\top X_t - x_t\|^2 + \eta^2 \|u_t^\top D_t Y_t \frac{l}{2}\|^2$$

$$- 2\eta (u_t^\top X_t - x_t, u_t^\top D_t Y_t \frac{l}{2}).$$
With (17), the second term can be easily bounded using
\[
\|u_t^T D_t Y_{t+\frac{1}{2}}\| \leq \|u_t\| \|D_t Y_{t+\frac{1}{2}}\| \leq \|G_t\|.
\]
As for the third term, it can be further decomposed as
\[
(u_t^T X_t - x_t^*, u_t^T D_t G_t) = (u_t^T X_t - x_t^*, u_t^T D_t G_t)
+ (x_t - x_t^*, u_t^T D_t (G_t - v_t 1^T G_t))
+ (\xi_t - x_t, u_t^T D_t (v_t 1^T G_t)).
\] (18)
where we used again \(D_t Y_{t+\frac{1}{2}} = D_t G_t\). Let us bound the three terms separately. Using Lemma 8b, for any \(\delta_1 > 0\), we have
\[
E[-2\eta\langle u_t^T X_t - x_t, u_t^T D_t G_t \rangle] \leq E[\eta \delta_1 \|u_t^T X_t - x_t\|^2 + \frac{\eta}{\delta_1} \|u_t^T D_t G_t\|^2] \leq \eta \delta_1 \|X_t - 1\xi_t\|^2 + \frac{\eta}{\delta_1} E[\|G_t\|^2].
\]
\[
\leq \eta \delta_1 \rho_t + \frac{\eta}{\delta_1} (4M^2 L^2 \rho_t + (8M + 8) \psi_t^4 + 4\zeta_t^4 + 8M^2 L e_t).
\]
With Lemmas 7b and 8a, we can bound the second term of (18) for any \(\delta_2 > 0\) as
\[
E[-2\eta\langle \xi_t - x_t^*, u_t^T D_t (G_t - v_t 1^T G_t) \rangle] \leq E[\eta \delta_2 \|\xi_t - x_t^*\|^2 + \frac{\eta}{\delta_2} \|u_t^T D_t (G_t - v_t 1^T G_t)\|^2] \leq 2\eta \delta_2 E[\|\xi_t - u_t^T X_t\|^2] + 2\eta \delta_2 E[\|u_t^T X_t - x_t\|^2] + \frac{\eta}{\delta_2} E[\|G_t - v_t 1^T G_t\|^2] \leq 2\eta \delta_2 \rho_t + 2\eta \delta_2 d_t' + \frac{\eta}{\delta_2} (2\zeta_t^4 + (4M + 4) \psi_t^4). (19)
\]
To bound the last term of (18), we use \(1^T G_t = \sum_{i=1}^{M} \nabla f_i(x_t^i)\). Following (7), we then get
\[
\langle \xi_t - x_t^*, 1^T G_t \rangle \geq \frac{M}{2} (f(\xi_t) - f(x_t^*)) + \frac{\mu M}{4} \|\xi_t - x_t\|^2 - \frac{L}{2} \|X_t - 1\xi_t\|^2 \geq \frac{M}{2} (f(\xi_t) - f(x_t^*)) + \frac{\mu M}{8} \|u_t^T X_t - x_t\|^2 - \frac{\mu M}{4} \|\xi_t - u_t^T X_t\|^2 - \frac{L}{2} \|X_t - 1\xi_t\|^2.
\]
Applying Lemmas 9 and 7b gives
\[
E[-2\eta \alpha_t \langle \xi_t - x_t^*, 1^T G_t \rangle] \leq -\eta \alpha_t M e_t - \frac{\eta}{\delta_1} d_t' + \frac{\eta}{\eta} \left( L + \frac{\mu M}{2} \right) \rho_t.
\]
We recall that \(\alpha_t = v_t D_t u_t\). Putting all together and choosing \(\delta_1 = 16ML^2 / 2\) and \(\delta_2 = \alpha_t \mu M / 16\), we obtain
\[
d_{t+1}' \leq \left( 1 - \frac{\eta}{\delta_1} \frac{\mu M}{4} \right) d_t' - \eta \alpha_t M e_t + \eta \left( L + \frac{\mu M}{2} \right) \rho_t + \frac{16\eta M L}{\alpha} \rho_t + \frac{16\eta M L}{\eta^2} \rho_t + \frac{16\eta M L}{\eta^2} \rho_t.
\]
\[
\cdot (4ML^2 \rho_t + (8M + 8) \psi_t^4 + 4\zeta_t^4 + 8M^2 L e_t) + \frac{\eta \alpha_t M}{8} \rho_t + \frac{\eta \alpha_t M}{8} d_t' + \frac{16\eta M}{\eta^2} (2\zeta_t^4 + (4M + 4) \psi_t^4).
\]
The coefficient of \(e_t\) is \(-\eta M (\frac{1}{2} - 8\eta M L)\). Since \(\eta \leq \alpha_t / (16ML)\), this is nonpositive and we have indeed
\[
d_{t+1}' \leq (1 - c_t) d_t' + c_t \rho_t + c_t \eta \psi_t^4 + c_t \eta \zeta_t^4
\]
for some positive constants \((c_t)_{1 \leq k \leq 4}\).

**Bounding \(d_t\).** Let \(s \in \{1, \ldots, t\}\). As the matrices \((A_t)_{t \in \mathbb{N}}\) are row-stochastic, it holds
\[
(I - J) A_{t+1} (I - J) A_s = (I - J) A_{t+1} A_s = (I - J) A_{t+1} A_s (I - J).
\]
Hence, for any \(\delta > 0\), we can write
\[
\|((I - J) A_{t+1} (I - J) X_{s+1})^2 = \|(I - J) A_{t+1} (I - J) X_s\|^2 + (1 + \delta) \|\|A_{t+1} (I - J) X_s\|^2 + \frac{1}{(1 + \delta)} \|\|I - J\|\| A_{t+1} (I - J) X_s\|^2 + \frac{1}{(1 + \delta)} \|\|I - J\|\| A_{t+1} (I - J) X_{s+1}\|^2 + \frac{1}{(1 + \delta)} \|\|I - J\|\| A_{t+1} (I - J) X_{s+1}\|^2 + \frac{1}{(1 + \delta)} \|\|I - J\|\| A_{t+1} (I - J) X_{s+1}\|^2 + \frac{1}{(1 + \delta)} \|\|I - J\|\| A_{t+1} (I - J) X_{s+1}\|^2.
\] (20)

In the last line, we have used the fact that \(A_{t+1}\) is row-stochastic so that \(\|A_{t+1}\| \leq \sqrt{M}\) and the inequality \(\|D_s Y_{s+\frac{1}{2}}\| \leq \|G_s\|\).

Since \(X_{t+1} - 1^T \xi_t^* = (I - J) A_{t+1} (I - J) X_{t+1}\), applying (20) repeatedly then gives
\[
\|X_{t+1} - 1^T \xi_t^*\|^2 \leq (1 + \delta)^{t+1} \|(I - J) A_{t+1} (I - J) X_s\|^2 + \frac{1}{(1 + \delta)} \|\|I - J\|\| A_{t+1} (I - J) X_{s+1}\|^2.
\]
Let \(\delta = \frac{1}{t+1} \log \frac{t+2}{t+1} > 0\) so that for all \(0 \leq s \leq l + 1\), we have \((1 + \delta)^s \leq \frac{2}{t+1} \lambda < 1\). Taking expectation in the above inequality and invoking Lemmas 6a and 8b leads to
\[
\rho_{t+1} \leq \frac{1 + \lambda_1}{2} \rho_t + \frac{\eta^2 M}{\lambda} \left( \frac{l + 1}{2} \right)^{\sum_{s=0}^{l} \Delta_{l-s}} + \frac{1}{\rho_t} \rho_t + \frac{1}{\rho_t} \sum_{s=0}^{l} (c_5 d_{t-s} c_6 \rho_{t-s} + c_7 \psi_{t-s} + c_8 \zeta_{t-s}).
\]

**Bounding \(\psi_{t+1}\).** By Young’s inequality
\[
\|
\n\|
\]
\[
\|
\]
\[
\|
\]
\[
\text{The update rule of } z_{t+1}^i \text{ implies that}
\]
\[
\frac{1}{\rho_t} \rho_t + \frac{1}{\rho_t} \sum_{s=0}^{l} (c_5 d_{t-s} c_6 \rho_{t-s} + c_7 \psi_{t-s} + c_8 \zeta_{t-s}).
\]

**Bounding \(\psi_{t+1}\).** By Young’s inequality
\[
\|
\]
\[
\|
\]
\[
\|
\]
\[
\text{The update rule of } z_{t+1}^i \text{ implies that}
\]
\[
\|\|f_i(x_{t+1}^i) - f_i(x_t^i)\|\|^2
\]
\[
\|\|f_i(x_{t+1}^i) - f_i(x_t^i)\|\|^2
\]
\[
= (1 - p_i) \| \nabla f_i(x_i') - \nabla f_i(x_i) \|^2 + p_i \| \nabla f_i(x_i') - \nabla f_i(x_i) \|^2
\]
\[
= (1 - p_i) \| \nabla f_i(z_i') - \nabla f_i(x_i) \|^2.
\]

With \( p = \min_{i \in V} p_i \) as defined in Lemma 9, we then have
\[
E_i[[\nabla f_i(x_i' + 1) - \nabla f_i(x_i')]^2] \leq \left(1 - \frac{p_i}{2}\right) \| \nabla f_i(z_i') - \nabla f_i(x_i) \|^2
\]
\[
+ 2 \frac{\| \nabla f_i(x_i') - \nabla f_i(x_i) \|^2}{p_i}.
\]

Taking total expectation and summing from \( i = 1 \) to \( M \) gives
\[
\psi_1' \leq \left(1 - \frac{p}{2}\right) \psi_1' + 2 \frac{L}{\lambda} \| \nabla f_i(x_i) - \nabla f(X_{t+1}) \|^2.
\]

By Lipschitz-continuity of the gradients, it holds \( \| \nabla f_i(x_i) - \nabla f(X_{t+1}) \|^2 \leq L \| x_i - X_{t+1} \|^2 \). We then develop
\[
X_{t+1} - X_t = A_t(X_t - \eta D_t Y_{t+\frac{1}{2}}) - X_t
\]
\[
= (A_t - I)(I - J) X_t - \eta A_t D_t Y_{t+\frac{1}{2}}.
\]

With \( \| A_t - I \|^2 \leq 2\| A_t \|^2 + 2\| I \|^2 \leq 2M + 2 \), we obtain that
\[
\| \nabla f(X_t) - \nabla f(X_{t+1}) \|^2 \leq L^2((4M + 4)\| x_t - \bar{x}_t \|^2 + 2\eta^2 \| G_t \|^2).
\]

Combining the above and applying Lemma 8b leads to
\[
\psi_1' \leq \left(1 - \frac{p}{2}\right) \psi_1' + 2 \frac{L^2}{\lambda} (4M + 4)\| x_t - \bar{x}_t \|^2
\]
\[
+ 2\eta^2 M(4M L^2 \rho_t + (8M + 8)\|\psi_t' + 4\|\psi_t' + 8M^2 L e(t))
\]
\[
\text{Using (16) and } \eta \leq \alpha/(16ML) \text{ we deduce the existence of positive constants } (c_1, c_2, c_3) \text{ such that}
\]
\[
\psi_1' \leq c_1 \eta \theta_t + c_1 \eta \rho_t + (1 - \frac{p}{2} + c_1 \eta \lambda) + c_1 \eta \lambda.
\]

Bounding \( \zeta_{t+1} \). Let \( s \in \{1, \ldots, t\} \). Using the column-stochasticity of \( B_s \) and the definition \( v_{s+1} = B_s v_s \), we get
\[
(I - v_{s+1}1^T)B_s = B_s - v_{s+1}1^T = B_s - B_s v_s 1^T.
\]

Hence, for any \( \delta > 0 \), it holds that
\[
\|B_{s+1}(I - v_{s+1}1^T)Y_{t+1}\|^2
\]
\[
= \|B_{s+1} - B_{s+1}v_s 1^T\|^2 \|D_s(\nabla F(X_s) - \nabla F(Z_s))\|^2
\]
\[
\leq (1 + \delta)\|B_{s+1}(I - v_s 1^T)Y_{t+1}\|^2
\]
\[
+ \left(1 + \frac{1}{\delta}\right)\|B_{s+1} - B_{s+1}v_s 1^T\|^2 \|D_s\|^2 \|\nabla F(X_s) - \nabla F(Z_s)\|^2
\]
\[
\leq (1 + \delta)\|B_{s+1}(I - v_s 1^T)Y_{t+1}\|^2
\]
\[
+ 4M \left(1 + \frac{1}{\delta}\right)\|\nabla F(X_s) - \nabla F(Z_s)\|^2.
\]

The last inequality holds because both \( B_{t_s} \) and \( B_{t_s} v_s 1^T \) are column-stochastic. Since \( Y_{t+1} - v_{t+1}1^T Y_{t+1} = B_{t+1}(I - v_{t+1}1^T)Y_{t+1} \), applying (21) repeatedly then gives
\[
\|Y_{t+1} - v_{t+1}1^T Y_{t+1}\|^2
\]
\[
\leq (1 + \delta)\|B_{t+1}(I - v_{t+1}1^T)Y_{t+1}\|^2
\]
\[
+ 4M \left(1 + \frac{1}{\delta}\right) \sum_{s=0}^{t} (1 + \delta)^s \|\nabla F(X_{t-s}) - \nabla F(Z_{t-s})\|^2.
\]

Let us take \( \delta = \frac{1}{1 + \log(1 + \delta)} > 0 \) as before. Taking total expectation in the above inequality and invoking Lemma 6b leads to
\[
\zeta_{t+1} \leq 1 + \frac{\theta}{\lambda} \zeta_t + 4M \left(1 + \frac{1}{\delta}\right) \sum_{s=0}^{t} \psi_{t-s},
\]
\[
\text{We set } c_{13} = \frac{4M}{\lambda} (1 + \frac{1}{\delta}).
\]

**Conclude.** Putting all together we get exactly (15). □

**3) Geometric Convergence of PPDS:** From Lemma 2, we are now in a position to prove the geometric convergence of PPDS by showing that the spectral radius of \( Q_0 + \eta Q_e \) is smaller than 1 for \( \eta > 0 \) sufficiently small.

**Proof of Theorem 1:** In the following, we analyze the eigenvalues of \( Q_0 + \eta Q_e \) with help of matrix perturbation theory. We first notice that \( Q_0 \) is a block-triangular matrix. Its characteristic polynomial can be easily computed and is given by
\[
\rho(Q_0) = \rho(2(\nu - 1) \left(1 - \frac{p}{2}\right)) \left(1 - \frac{1}{2}\right)^2.
\]

This shows that the spectral radius of \( Q_0 \) is 1 and 1 is also the unique eigenvalue of the largest modulus of the matrix.

Let us denote by \( \theta_1 = 1, \theta_2, \ldots, \theta_{4t+1} \) the eigenvalues of \( Q_0 \) so that \( |\theta_k| < 1 \) for all \( k \in \{2, \ldots, 4(t+1)\} \). By continuity of the eigenvalues, for any \( \varepsilon > 0 \), there exists \( \delta > 0 \) such that if \( \eta < \delta \), for any \( \theta_k \) of multiplicity \( m \) the matrix \( Q_0 + \eta Q_e \) has exactly \( m \) eigenvalues (counting multiplicity) in \( \mathbb{B}(\theta_k, \varepsilon) \), the open disk centered at \( \theta_k \) with radius \( \varepsilon \); see [38, Ch. 5.1]. Let us take \( \varepsilon \) small enough such that all the eigenvalues of \( Q_0 + \eta Q_e \) are smaller than \( 1 - \varepsilon \) in modulus except the greatest one. For \( \eta < \delta \), we can then define \( \theta_1(\eta) \) as the unique eigenvalue of \( Q_0 + \eta Q_e \) that is in \( \mathbb{B}(1, \varepsilon) \). We will now show that \( |\theta_1(\eta)| < 1 \) for \( \eta \) sufficiently small. For this, let
\[
u = [10 \ldots 0], \quad v = [1 \ldots 10 \ldots 0]^T.
\]
\[
\text{be, respectively, the left and the right eigenvectors of } Q_0 \text{ associated with the eigenvalue 1. By [39, Th. 6.3.12] (see also [40, Th. 1]), we have}
\]
\[
\theta'_1(0) = \nu^T Q_0 \nu / \nu^T v = -c_1 < 0.
\]

As a consequence, \( |\theta_1(\eta)| < 1 \) for \( \eta \) sufficiently small and subsequently \( \theta(Q_0, \eta Q_e) < 1 \).

In order to conclude, we need to get rid of the dependence on \( T \) which plays a role in the definition of the vectors \( (u_t)_{1 \leq t \leq T} \) and the quantities \( (d_t')_{1 \leq t \leq T} \). We recall that \( d_t = E[\|x_t - \bar{x}_t\|^2] \). As in (19), we have both \( d_t \leq 2 \delta_t' + 2 \rho \) and \( d_t' \leq 2 \delta_t + 2 \rho \). Let us define \( d_t'' \) by replacing \( (d_t')_{1 \leq t \leq t+1} \) by \( (d_s)_{s \leq s \leq t+1} \). The
above inequalities can then be translated into $r_i^t \leq Wr_i^t$ and $r_i^t \leq Wr_i^t$ for a nonnegative matrix $W$ properly defined. Note that neither $W$ nor $Q_0 + \eta Q_c$ depend on $t$ or $T$. Therefore, the inequality

$$r_i^t \leq W(Q_0 + \eta Q_c)^{t-1} Wr_i^t$$

which holds for all $t \in \mathbb{N}$ guarantees the geometric convergence of $r_i^t$ and subsequently of all the relevant quantities when $\eta$ is small enough.

Finally, from the geometric convergence of $\mathbb{E}[\|x_i^t - x_i\|^2]$, we deduce that $x_i^t$ converges to $x_i$, almost surely by using Markov’s inequality and the Borel–Cantelli lemma.

\[\Box\]

V. SIMULATIONS

In this section, we illustrate the interest of PPDS for asynchronous decentralized optimization on a synthetic ridge regression problem and a logistic regression problem on a real dataset.\(^4\)

A. Dataset, Tasks, and Models

For both problems, we minimize an objective of the form

$$f(x) = \frac{1}{M} \sum_{i=1}^{M} \left( \sum_{j=1}^{n_{\text{local}}} f_{i,j}(x) + \lambda_i \|x\|^2 \right)$$

that is, each worker $i$ has a local dataset of $n_{\text{local}}$ examples and a Tikhonov regularization term with parameter $\lambda_i = 1/n_{\text{local}}$. Since the local objectives are convex, this regularization makes the problem strongly convex.

We form the communication networks by generating random geometric graphs (RGG) using the library `networkx` [41] with a different number of nodes $M$ and radius $r$ for each experiment. We consider the broadcast setting illustrated in Section II-C3. Precisely, all activated nodes broadcast their models to one (randomly chosen) neighbor during a communication step. We illustrate the effect of device sampling by comparing algorithms with full-device participation and with random device sampling (20 nodes for the synthetic dataset and 10 nodes for EMNIST). The relevant parameters are reported in Table I.

---

**A. Ridge regression on synthetic dataset**: For this problem, the local losses are defined as

$$f_{i,j}(x) = (b_{i,j} - x^\top a_{i,j})^2$$

where $(a_{i,j}, b_{i,j}) \in \mathbb{R}^d \times \mathbb{R}$ are data points generated using the procedure `make_regression` from scikit-learn [42]. Different seeds are used for different nodes, yielding statistically heterogeneous distributions between the nodes.

**B. Logistic regression on EMNIST**: The EMNIST dataset [43] is comprised of images of handwritten digits and letters from several authors. We consider the problem of finding which character is written from its image. For this, we consider local losses of the form

$$f_{i,j}(x) = -b_{i,j} \log(\text{softmax}(x^\top a_{i,j}))$$

where the $(a_{i,j}, b_{i,j})$ are, respectively, $d = 28 \times 28$ grayscale images of handwritten digits character and their associated one-hot label $b_{i,j} \in \{0 - 9, a - z, A - Z\}$ totaling 62 classes. Each worker’s local dataset comes from images from the same author.

B. Algorithms, Hyperparameters, and Evaluation Metrics

We compare the proposed algorithm PPDS with several baselines: decentralized gradient descent (DGD) with and without sampling, push–pull, and G-push–pull. The same broadcast communication scheme is applied to all the methods, and the same uniform sampling strategy is adopted whenever device sampling is involved.

For each method, the stepsize is taken fixed, tuned with a coarse-to-fine strategy: We first select the stepsize $\eta$ within $\{10^{-k}, 2 \leq k \leq 5\}$ yielding the best global training loss; then, a second search is performed over $\{\eta 2^k, -2 \leq k \leq 2\}$.

We use the functional suboptimality $(1/M) \sum_{i=1}^{M} f(x_i) - f_*$ as evaluation metric. To compute $f_*$, we obtain the optimal solution by solving a linear system in the case of synthetic dataset and we set it as the best solution in terms of final training losses found by the implemented algorithms in the case of real dataset.

For each experiment, we report the metric in terms of three different measures:

1. number of iterations;
2. communication cost, i.e., the cumulative number of activated communication links;
3. number of local updates.

These cover different aspects that influence the efficiency of distributed optimization.

C. Numerical Results

First, we observe Figs. 1 and 2 that the proposed method PPDS converges linearly, as expected from Theorem 1. Furthermore, looking at the right-hand plots, we see that PPDS outperforms all the other methods when it comes to measuring the functional optimality with respect to the number of local updates. This illustrates that PPDS, indeed, saves computational resources by an efficient interplay between computation and communications. Concerning the communication complexity (middle plots), PPDS is at least as competitive as G-push–pull, which was shown in [15] to beat other baselines. Finally, we observe

---

\[\footnotemark[4]\] The code to reproduce the experiments can be found at https://github.com/yassine-laguel/ppds.
that push–pull, as a synchronous gradient-tracking method, naturally achieves the best performance when measuring in terms of the number of iterations (left-hand plots), but tends to be less efficient when we consider the actual communication and computational costs.

**VI. CONCLUSION**

In this article, we showed how device sampling can be incorporated into asynchronous decentralized gradient descent, by extending the push–pull method. We proved linear convergence of the method on strongly convex functions and validate our approach on problems with synthetic and real data. This work also opens several research directions. This goes from the theoretical analysis of our method in nonstrongly-convex, nonconvex, or the stochastic setup (see, e.g., [33] for analysis of gradient tracking in these setups) to the investigation of how our approach can be combined with other existing techniques such as local gradients computation and gradient compression.
**APPENDIX A**

**PROOF OF LEMMA 4**

Proof: 
a) Since $f_i$ is $L$-smooth, it holds for all $x, x' \in \mathbb{R}^d$ that
\[
\|\nabla f_i(x) - \nabla f_i(x')\|^2 \leq 2L(f_i(x) - f_i(x') - \langle x - x', \nabla f_i(x') \rangle).
\]
Subsequently
\[
\sum_{i=1}^{M} \|\nabla f_i(x_i^t) - \nabla f_i(x_*)\|^2 
\leq 2 \sum_{i=1}^{M} \|\nabla f_i(x_i^t) - \nabla f_i(x_*)\|^2 
+ 2 \sum_{i=1}^{M} \|\nabla f_i(x_*) - \nabla f_i(x_*)\|^2 
\leq 2L \sum_{i=1}^{M} \|x_i^t - x_*\|^2 
+ 4L \sum_{i=1}^{M} (f_i(x_*) - f_i(x_*) - \langle x_i^t - x_*, \nabla f_i(x_*) \rangle) 
= 2L^2 \|x_i^t - 1x_*\|^2 
+ 4ML(f(x_*) - f(x_*) - \langle x_i^t - x_*, \nabla f(x_*) \rangle) 
= 2L^2 \|x_i^t - 1x_*\|^2 + 4ML(f(x_*) - f(x_*)).
\]
In the last line, we used that $\nabla f(x_*) = 0$. Taking expectations gives the desired inequality.

b) The inequality is straightforward from a) and the following decomposition:
\[
\|\nabla F(X_i) - \nabla F(Z_i)\|^2 
\leq 2\|\nabla F(X_i) - \nabla F(1^T x_*)\|^2 
+ 2\|\nabla F(1^T x_*) - \nabla F(Z_i)\|^2.
\]

C) We first decompose
\[
\|G_t\|^2 = \|(I - J)G_t + JG_t\|^2 = \|(I - J)G_t\|^2 + \|JG_t\|^2.
\]
For the first term, we simply use the definition of $G_t$ to obtain
\[
\|(I - J)G_t\|^2 
\leq 2\|(I - J)Y\|^2 + 2\|(I - J)(\nabla F(X_t) - \nabla F(Z_i))\|^2 
\leq 2\|Y_t - 1y_*\|^2 + 2\|\nabla F(X_t) - \nabla F(Z_i)\|^2.
\]
From Lemma 1b, we know that $\sum_{i=1}^{M} g_i = \sum_{i=1}^{M} \nabla f_i(x_i^t)$ or equivalently $JG_t = J\nabla F(X_t)$. Thus, for the second term, we use again $\nabla f(x_*) = 0$ to get
\[
\|JG_t\|^2 = M \left\| \frac{1}{M} \sum_{i=1}^{M} \nabla f_i(x_i^t) - \frac{1}{M} \sum_{i=1}^{M} \nabla f_i(x_*) \right\|^2 
\leq \sum_{i=1}^{M} \|\nabla f_i(x_i^t) - \nabla f_i(x_*)\|^2.
\]
Combining the above, taking expectation, and using a) and b) gives the desired result.

**APPENDIX B**

**PROOF OF (11)**

Proof: Let $Q_k$ denotes the $k$th column of $Q$ and $e_k$ denote the $k$th canonical vector of $\mathbb{R}^4$. First, $\omega^\top Q_1 = 1 - \frac{\eta_S}{2M} = (1 - \frac{\eta_S}{2M}) \omega^\top e_1$.

Since $\eta \leq \frac{1 - \lambda^2}{144L^2 \sqrt{M} \lambda^2}$, it holds
\[
\omega^\top Q_2 = \frac{\sqrt{S}(1 - \lambda)}{M^\frac{3}{2}} \left( \frac{\eta L}{1 - \lambda} \sqrt{\frac{S}{M}} + \frac{10\eta^2 L^2}{1 - \lambda} \left( \frac{S}{M} \right)^{\frac{3}{2}} \right) 
+ \frac{1 + \lambda}{2} + \frac{20\eta^2 L^2 S}{M(1 - \lambda)} + \frac{\eta L}{4(1 - \lambda)} \sqrt{\frac{S}{M}} \leq \frac{\sqrt{S}(1 - \lambda)}{M^\frac{3}{2}} \left( \frac{5\eta L}{4(1 - \lambda)} \sqrt{\frac{S}{M}} + \frac{30\eta^2 L^2 S}{M(1 - \lambda)} + \frac{1 + \lambda}{2} \right) 
\leq \frac{\sqrt{S}(1 - \lambda)}{M^\frac{3}{2}} \frac{3 + \lambda}{4} = \frac{3 + \lambda}{4} \omega^\top e_2.
\]
With $\eta \leq \frac{1 - \lambda^2}{1296L^2 \sqrt{M} \lambda^2}$, we have
\[
\omega^\top Q_3 = \frac{\eta (1 - \lambda)}{96ML} \left( \frac{192\eta L}{1 - \lambda} \sqrt{\frac{S}{M}} \right)^2 
+ \frac{384\eta L}{1 - \lambda} \left( \frac{S}{M} \right)^{\frac{3}{2}} + \frac{1 + \lambda}{2} \leq \frac{\eta (1 - \lambda)}{96ML} \left( \frac{576\eta L}{1 - \lambda} \sqrt{\frac{S}{M}} \right)^{\frac{3}{2}} + \frac{1 + \lambda}{2} 
\leq \frac{\eta (1 - \lambda)}{96ML} \frac{3 + \lambda}{4} = \frac{3 + \lambda}{4} \omega^\top e_3.
\]
Similarly, using $\eta \leq \frac{1}{576L} \sqrt{\frac{M}{S}}$, we get
\[
\omega^\top Q_4 = \frac{\eta}{12ML} \left( \frac{48\eta L}{2M} \right)^2 + 96\eta L \left( \frac{S}{M} \right)^{\frac{3}{2}} 
+ \frac{S}{2M} + 1 - \frac{S}{M} \leq \frac{\eta}{12ML} \left( 1 - \frac{S}{2M} + 144\eta L \left( \frac{S}{M} \right)^{\frac{3}{2}} \right) 
\leq \frac{\eta}{12ML} \left( 1 - \frac{S}{4M} \right) = \left( 1 - \frac{S}{4M} \right) \omega^\top e_4.
\]
As for $e_t$, we note that $\eta \leq \frac{(1 - \lambda^2)^2}{2304L^2 \sqrt{M} \lambda^2}$ and thus
\[
\omega^\top h = -\eta S \frac{M}{2M} + 20\eta^2 L S^2 \frac{M}{M^2} + 40\eta^2 L \left( \frac{S}{M} \right)^{\frac{3}{2}} + \frac{\eta S}{6M} + \frac{\eta S}{3M} 
\leq -\frac{\eta S}{2M} + 60\eta^2 L \left( \frac{S}{M} \right)^{\frac{3}{2}} \leq 0.
\]
As $\eta \leq \frac{(1-\lambda)^2}{4\gamma^2} \sqrt{\frac{M}{2}}$ implies that $1 - \frac{\mu_S}{2M} \geq \frac{3+\lambda}{4}$, we have
$$\omega^T Q \leq \gamma \omega^T I$$
where the inequality is elementwise and $\gamma = \max(1 - \frac{\mu_S}{2M}, 1 - \frac{S}{2M})$. Since all the involved terms are nonnegative, combining with the above inequalities gives
$$\omega^T r_{t+1} = \omega^T Q r_t + e_t \omega^T h \leq \gamma \omega^T r_t.$$

**APPENDIX C**

**PROOF OF LEMMA 8**

**Proof:**

a) From the definition of $g_t$, we can write
$$\|G_t - v_t^1 G_t\|^2 = \|Y_t - v_t^1 Y_t + (I - v_t^1 I)(\nabla F(X_t) - \nabla F(Z_t))\|^2 \leq 2\|Y_t - v_t^1 Y_t\|^2 + 2\|I - v_t^1 I\|^2 \|\nabla F(X_t) - \nabla F(Z_t)\|^2.$$ We conclude by using
$$\|I - v_t^1 I\|^2 \leq 2\|I\|^2 + 2\|v_t^1 I\|^2 \leq 2M + 2$$
taking expectation over the above inequalities.

b) By Young’s inequality
$$\|G_t\|^2 \leq 2\|G_t - v_t^1 G_t\|^2 + 2\|v_t^1 G_t\|^2.$$ Using $1^T G_t = 1^T \nabla F(X_t)$, $\nabla f(x_t) = 0$, and the fact that $v_t$ is a probability vector, we deduce that
$$\|v_t^1 G_t\|^2 \leq \left(\sum_{i=1}^M \nabla f_i(x_t^i)\right)^2 = \left(\sum_{i=1}^M \nabla f_i(x_t^i) - \sum_{i=1}^M \nabla f_i(x_t^i)\right)^2 \leq M \sum_{i=1}^M \|\nabla f_i(x_t^i)\|^2.$$ Combining the above two inequalities with a) and Lemma 4a, we get the desired result.

**APPENDIX D**

**PROOF OF LEMMA 9**

**Proof:** Using the independence assumption, we can write
$$\mathbb{E}[\alpha_t \chi_t] = \mathbb{E}[u_{t+1}^T A_t D_t v_t \chi_t] = \mathbb{E}[u_{t+1}^T] \mathbb{E}[A_t D_t] \mathbb{E}[v_t \chi_t].$$

From Assumption 3c, we deduce that
$$\mathbb{E}[A_t D_t] \geq \mathbb{E}[\text{diag}(\nu 1) D_t] = \nu \text{diag}(\mathbb{P})$$
where $\mathbb{P} = (p_t)_{t \in T}$. We have shown in (12) that $\mathbb{E}[u_{t+1}] = \pi_A$. Notice that all the elements of $\pi_A$ are positive according to the Perron–Frobenius theorem. Thus, $\pi_A > 0$ and we have
$$\mathbb{E}[u_{t+1}^T] \mathbb{E}[A_t D_t] \geq \pi_A^T \nu \text{diag}(\mathbb{P}) \geq \alpha 1^T$$
where Assumption 4 ensures that $p > 0$ and thus $\alpha > 0$.

On the other hand, $u_t$ being a probability vector, we can always upper bound $u_t^T D_t$ by $1^T$. Using the nonnegativity of $v_t$ and $\chi_t$, we then obtain
$$\mathbb{E}[v_t \chi_t] \leq \mathbb{E}[\alpha \chi_t] \leq \mathbb{E}[1^T v_t \chi_t].$$

This is exactly (14) since $v_t$ is a probability vector.

**REFERENCES**

[1] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Trans. Autom. Control*, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.

[2] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Sci., 2015.

[3] A. Nedić, A. Olshevsky, and M. G. Rabbat, “Network topology and communication-computation tradeoffs in decentralized optimization,” *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.

[4] T. Yang et al., “A survey of distributed optimization,” *Ann. Rev. Control*, vol. 47, pp. 278–305, 2019.

[5] M. Assran, A. Aytekin, H. R. Feyzmahdavian, M. Johansson, and M. G. Rabbat, “Advances in asynchronous parallel and distributed optimization,” *Proc. IEEE*, vol. 108, no. 11, pp. 2013–2031, Nov. 2020.

[6] A. Nedić, “Distributed gradient methods for convex machine learning problems in networks: Distributed optimization,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 92–101, May 2020.

[7] J. C. Duchi, A. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: Convergence analysis and network scaling,” *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[8] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Boston, MA, USA: Now Publishers Inc., 2011.

[9] A. Nedić and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[10] D. Kovalev, A. Salim, and P. Richtärík, “Optimal and practical algorithms for smooth and strongly convex decentralized optimization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 18342–18352.

[11] J. M. Hendrickx and J. N. Tsitsiklis, “Fundamental limitations for anonymous distributed systems with broadcast communications,” in *Proc. 53rd Ann. Allerton Conf. Commun., Control, Comput.*, 2015, pp. 9–16.

[12] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information,” in *Proc. 44th Ann. IEEE Symp. Found. Comput. Sci.*, 2003, pp. 482–491.

[13] F. Fuzeter, P. Ciblat, and W. Hachem, “Analysis of sum-weight-like algorithms for averaging in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2802–2814, Jun. 2013.

[14] A. Nedić and A. Olshevsky, “Stochastic gradient-push for strongly convex functions on time-varying directed graphs,” *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.

[15] S. Pu, W. Shi, J. Xu, and A. Nedić, “Push–pull gradient methods for distributed optimization in networks,” *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, Jan. 2020.

[16] R. Xin and U. A. Khan, “A linear algorithm for optimization over directed graphs with geometric convergence,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 315–320, Jul. 2018.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[18] F. Fuzeter, P. Bianchi, P. Ciblat, and W. Hachem, “Asynchronous distributed optimization using a randomized alternating direction method of multipliers,” in *Proc. IEEE 52nd Conf. Decis. Control*, 2013, pp. 3671–3676.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Conf. Artif. Intel. Statist.*, 2017, pp. 1273–1282.

[20] P. Kairouz et al., “Advances and open problems in federated learning,” *Proc. Nat. Acad. Sci.*, vol. 114, no. 12, pp. 3061–3070, Apr. 2017.

[21] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1646–1654.

[22] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, “Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant step-sizes,” in *Proc. IEEE 54th Conf. Decis. Control*, 2015, pp. 2085–2090.
Was born in Besançon, France, received the Ph.D. degree in mathematics with Université Grenoble Alpes, Grenoble, France, in 2021.

His research interests include distributed optimization, online learning, and the study of learning-in-game dynamics.

**Franck Iutzeler** was born in Besançon, France, in 1987. He received the engineering degree from Telecom Paris, Paris, France, in 2010, the M.Sc. degree from Sorbonne Université, Paris, in 2010, and the Ph.D. degree from Telecom Paris in 2013, all in applied mathematics.

During 2014–2015, he was a Postdoctoral Associate, first with Supelec, Gif-sur-Yvette, France, and then with Université Catholique de Louvain, Louvain-la-Neuve, Belgium. Since 2015, he has been an Assistant Professor with Université Grenoble Alpes. His research interests resolve around optimization methods and theory for data science, in particular, focusing on machine learning problems, distributed optimization, robust optimization, and multiagent systems.

**Jérôme Malick** received the Ph.D. degree in mathematics from Inria, Paris, France, in 2005.

He is currently a Senior Researcher with CNRS and has been leading the DAO team of Lab. Jean Kuntzmann, Université Grenoble Alpes, Grenoble, France, since 2017. He was a Postdoc with Cornell University in 2006. His research interests include mathematical optimization and its interactions, especially on distributed optimization, multiagent learning, and optimization under uncertainty. Dr. Malick was the recipient of the Robert Faure award for the Most Outstanding Young Researcher in Optim/OR in France in 2009.