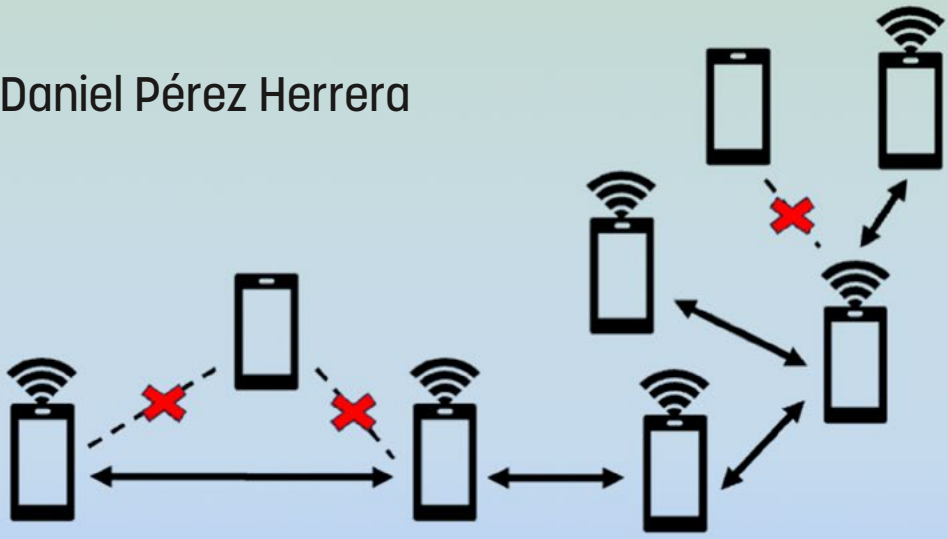
Linköping Studies in Science and Technology
Licentiate Thesis No. 2004

Communication-Efficient Scheduling Designs for Distributed Consensus and Optimization over Wireless Networks

Daniel Pérez Herrera



LINKÖPING UNIVERSITY

Linköping Studies in Science and Technology
Licentiate Thesis, No. 2004

Communication-Efficient Scheduling Designs for Distributed Consensus and Optimization over Wireless Networks

Daniel Pérez Herrera

Division of Communication Systems
Department of Electrical Engineering (ISY)
Linköping University, 581 83 Linköping, Sweden
www.commsys.isy.liu.se

Linköping 2024

This is a Swedish Licentiate Thesis.
The Licentiate degree comprises 120 ECTS credits of postgraduate studies.

Communication-Efficient Scheduling Designs for Distributed Consensus and Optimization over Wireless Networks

ISBN 978-91-8075-785-0 (print)
ISBN 978-91-8075-786-7 (PDF)
https://doi.org/10.3384/9789180757867
ISSN 0280-7971

Printed in Sweden by LiU-Tryck, Linköping 2024

Abstract

In recent years, there has been a significant surge in the development of artificial intelligence, with machine learning emerging as a fundamental aspect of its applications. Machine learning algorithms enable systems to learn from data and make predictions or decisions without explicit programming. In distributed environments, where data is often distributed across multiple nodes, decentralized learning methods have become increasingly prevalent. These methods allow for collaborative model training without using centralized data, offering benefits such as scalability, privacy, and efficiency. To ensure convergence and accuracy of the learned models, achieving consensus among distributed nodes is paramount. Consensus mechanisms enable nodes to agree on a common model despite variations in local data distributions and computational resources, forming the backbone of decentralized learning systems. Thus, the development of efficient consensus protocols is essential for realizing the potential of decentralized learning in various domains, ranging from IoT applications to large-scale data analytics.

This thesis explores strategies to minimize the communication cost in wireless multi-agents systems. It examines the potential of leveraging the broadcast nature of wireless networks, focusing on two frameworks: distributed average consensus and decentralized learning.

In distributed average consensus, wherein nodes aim to converge to the average of the initial values despite communication limitations, a novel probabilistic scheduling approach is proposed. This approach aims to streamline communication by selectively choosing a subset of nodes to broadcast information to their neighbors in each iteration. Various heuristic methods for determining node broadcast probabilities are evaluated, alongside the introduction of a pre-compensation technique to mitigate potential bias. These contributions shed light on the design of communication-efficient consensus protocols tailored to wireless environments with restricted resources.

Transitioning to decentralized learning, the thesis introduces `BASS` (Broadcast-based Subgraph Sampling) to expedite the convergence of D-SGD (decen-

tralized stochastic gradient descent) while considering the communication overhead. By generating a set of mixing matrix candidates that represent sparse subgraphs of the network topology, `BASS` facilitates the activation of collision-free subset of nodes in each iteration, optimizing communication efficiency. The optimization of sampling probabilities and the mixing matrices significantly enhances convergence speed and resource utilization compared to existing approaches. These findings underscore the inherent advantages of leveraging the broadcast capabilities of wireless channels to enhance the efficiency of decentralized optimization and learning algorithms in distributed systems.

Populärvetenskaplig Sammanfattning

Under de senaste åren har utvecklingen av artificiell intelligens ökat markant, där maskininlärning har blivit en grundläggande aspekt av dess tillämpningar. Maskininlärningsalgoritmer gör det möjligt för system att lära sig från data och göra förutsägelser eller beslut utan explicit programmering. I distribuerade miljöer, där data ofta är spridda över flera noder, har decentraliserade inlärningsmetoder blivit alltmer vanliga. Dessa metoder möjliggör samarbetsbaserad modellträning utan att använda centraliserad data, vilket erbjuder fördelar som skalbarhet, integritet och effektivitet. För att säkerställa konvergens och noggrannhet hos de inlärda modellerna är det avgörande att uppnå konsensus bland distribuerade noder. Konsensusmekanismer gör det möjligt för noder att enas om en gemensam modell trots variationer i lokala datafördelningar och beräkningsresurser, vilket utgör ryggraden i decentraliserade inlärningssystem. Därför är utvecklingen av effektiva konsensusprotokoll väsentlig för att realisera potentialen hos decentraliserad inlärning inom olika områden, från IoT-tillämpningar till storskalig dataanalys.

Denna avhandling utforskar strategier för att minimera kommunikationskostnader i trådlösa multiagentsystem. Den undersöker potentialen att utnyttja de sändningsegenskaper som trådlösa nätverk har, med fokus på två ramverk: distribuerad genomsnittskonsensus och decentraliserad inlärning.

I distribuerad genomsnittskonsensus, där noder strävar efter att konvergera till medelvärdet av de initiala värdena trots kommunikationsbegränsningar, föreslås en ny probabilistisk schemaläggningsmetod. Denna metod syftar till att effektivisera kommunikationen genom att selektivt välja en delmängd av noder som ska sända information till sina grannar vid varje iteration. Olika heuristiska metoder för att bestämma sändsannolikheter för noder utvärderas, tillsammans med införandet av en förkompensationsmetod för att motverka potentiell bias. Dessa bidrag belyser designen av kommunikationseffektiva

konsensusprotokoll anpassade till trådlösa miljöer med begränsade resurser.

Vid övergången till decentraliserad inlärning introducerar avhandlingen `BASS` (Broadcast-based Subgraph Sampling) för att påskynda konvergensen av D-SGD (decentraliserad stokastisk gradientnedstigning) samtidigt som hänsyn tas till kommunikationsöverhead. Genom att generera en uppsättning av blandningsmatriskandidater som representerar glesa subgrafer av nätverkstopologin, möjliggör `BASS` aktivering av en krockfri delmängd av noder vid varje iteration, vilket optimerar kommunikationseffektiviteten. Optimeringen av samplingssannolikheter och blandningsmatriserna förbättrar avsevärt konvergenshastigheten och resursutnyttjandet jämfört med befintliga metoder. Dessa resultat understryker de inneboende fördelarna med att utnyttja sändningsegenskaperna hos trådlösa kanaler för att förbättra effektiviteten hos decentraliserade optimerings- och inlärningsalgoritmer i distribuerade system.

Contents

Acknowledgements **ix**

List of Abbreviations **xi**

1 Introduction and Motivation **1**
1.1 Thesis Outline 3

2 Graph Theory Preliminaries **5**
2.1 Matrices for Graph Representations 6
2.1.1 Adjacency Matrix 6
2.1.2 Incidence Matrix 6
2.1.3 Laplacian Matrix 6
2.2 Centrality Metrics 7

3 Consensus Mechanisms **9**
3.1 Consensus Over Fixed Topologies 9
3.1.1 Most Fundamental Case 10
3.1.2 Convergence Speed 11
3.2 Consensus Over Time-Varying Topologies 12
3.2.1 Consensus Over Random Networks 13
3.2.2 Consensus with Packet Drop Communication 14

4 Decentralized Learning over Wireless Networks **15**
4.1 Variants of Gradient Descent 15
4.2 Decentralized Stochastic Gradient Descent 16
4.3 Communication Paradigms 19
4.3.1 Unicast Communication 19
4.3.2 Multicast Communication 19
4.3.3 Broadcast Communication 19
4.4 Overview of Wireless Networks 20

4.4.1 Communication Model with Packet Success / Collision 21
4.4.2 Wireless Networks Multiple Access Coordination . . . 21

5 Contributions of the Thesis 23
5.1 Included Papers . . . 23
5.2 Excluded Papers . . . 24

6 Conclusion and Future Work 27
6.1 Conclusion . . . 27
6.2 Future Work . . . 28

Bibliography 29

Included Papers 36

A Distributed Consensus in Wireless Networks with Probabilistic Broadcast Scheduling 37
1 Introduction . . . 39
2 System Model . . . 40
2.1 Graph Model . . . 40
2.2 Consensus Algorithm . . . 41
3 Partial Communication with Probabilistic Broadcast Scheduling 42
3.1 Heuristic Designs for Broadcast Probability Vector . . 43
3.2 Bias Correction . . . 44
3.3 Possible Extensions . . . 45
4 Simulation Results . . . 46
4.1 Broadcast Probability Vector Design . . . 46
4.2 Partial Communication with Bias Correction . . . 47
4.3 Potential Improvement by Using SPSA . . . 47
5 Conclusion . . . 48

B Faster Convergence with Less Communication: Broadcast-Based Subgraph Sampling for Decentralized Learning over Wireless Networks 53
1 Introduction . . . 55
1.1 Related Works . . . 57
1.2 Contributions . . . 57
2 Decentralized Optimization and Learning . . . 58
2.1 Notation . . . 58
2.2 Network Model and Graph Preliminaries . . . 58

2.3 Collaborative Machine Learning in Decentralized Setting 58
2.4 Discussion on the Properties of the Mixing Matrix . . 60
2.5 Discussions on the Assumptions 61
3 D-SGD over Wireless Networks with Broadcast Communication 61
3.1 Iterations vs. Transmission Slots 62
3.2 Full vs. Partial Communication 63
3.3 Link vs. Node Scheduling 63
4 BASS: Broadcast-Based Subgragh Sampling for Communication-Efficient D-SGD 64
4.1 Step 1: Create Collision-Free Subsets 64
4.2 Step 2: Create Subgraph Candidates 65
4.3 Step 3: Optimize Mixing Matrices and Sampling Probabilities 67
4.4 Initial Mixing Matrix Candidates 69
4.5 Comparison with Existing Approaches 72
5 Simplified Heuristic Design 73
5.1 Discussions on Complexity 78
6 Numerical Experiments 78
6.1 Performance Evaluation of BASS 79
6.2 Impact of the Communication Budget 80
6.3 Impact of Initial Graph Partition 80
6.4 Additional Experiments 81
7 Conclusions and Future Directions 83
7.1 Proof for the convexity of Problem (7) under a given set of sampling probabilities 85

Acknowledgements

First and foremost, I want to express my immense gratitude to my main supervisor, Prof. Erik G. Larsson, and my co-supervisor, Assoc. Prof. Zheng Chen, for their invaluable help and guidance throughout my journey as a Ph.D. student at Linköping University. They have taught me many things, and this thesis would not have been possible without their support.

I also want to thank all my colleagues for the time we've spent together and for creating such a great working environment. Special thanks to Deijany, my fellow Cuban and fika partner. I also want to thank Zakir for our technical and non-technical discussions.

I am deeply grateful to my family, especially my mom, Eileen (madrecita), for her unwavering support, encouragement, and for always being there for me. Even though she is across the world, she is always in my heart. Special thanks to my father, Hans, for his advice and for helping me see what truly matters during difficult times.

Finally, there are two people I cannot thank enough. One is my kaki, Gizeh, who is one of the most wonderful people in the world. The second is my wife, Rosangel, my greatest support and life partner, whose unconditional love keeps me going. Tack för att ni existerar!

Daniel Pérez Herrera
Linköping, August 2024

List of Abbreviations

ML	machine learning
FL	federated learning
DL	decentralized learning
D-SGD	decentralized stochastic gradient descent
SGD	stochastic gradient descent
i.i.d.	independently and identically distributed
GD	gradient descent
MB-SGD	mini-batch stochastic gradient descent
FDMA	frequency division multiple access
TDMA	time division multiple access

Chapter 1

Introduction and Motivation

Machine learning (ML) is a subfield of artificial intelligence [1] that learns patterns and hidden features from given data to make predictions by model training [2]. Over the last years, ML has gained an increased popularity due to proliferation of large-scale data, advancements in computational capabilities, and significant algorithmic improvements, particularly in deep learning. Additionally, the accessibility of open-source tools and frameworks, coupled with the growing integration of ML across diverse industrial applications, has further driven its adoption.

Representation learning [3] is a process in ML that enables a machine to ingest raw data and autonomously unveil the necessary representations for detection or classification. Such representations are essentially different ways of encoding or transforming the data so that its important characteristics or patterns are easier to understand and use. Deep learning methods [4], are representation learning approaches with multiple layers of representation. These layers are formed by combining non-linear modules, each responsible for transforming the representation from one level (originating from the raw input) to a slightly more abstract level. By sequentially applying these transformations, the model can learn highly intricate functions. In classification tasks, upper layers of representation emphasize crucial aspects of the input for discrimination while mitigating the impact of irrelevant variations [5]. In its most basic form, deep learning operates on a single computer (centralized ML), where a model is trained using local data for predictions. However, with the massive increase in data availability, computation efficiency, and growing privacy concerns, alternatives like federated learning (FL) [6] and decentralized learning (DL) [7, 8] have gained popularity.

In FL, the data is distributed among several users that keep it private [9],

and use it to train their local copy of an ML model located in a central server. The main objective of FL is to train the server ML model by using the local data of the users without data exchange, i.e., only the model parameters are exchanged between users and the central server. The model parameters received by the server are combined, and the resulting parameters are sent back to the users for further training with their local data. This iterative process continues until some criteria for convergence is satisfied, like a total number of iterations or a certain test accuracy level. Since the model aggregation occurs in the central server, it becomes a bottleneck in the learning process, affecting the scalability and being prone to failures. In contrast, DL does not rely on a central server for model aggregation, and each user updates its local model parameters by linearly combining them with the parameters from other users.

In its core, DL is an iterative process where each user has a copy of an ML model and a local dataset. The objective is to train and obtain a common ML model without relying on a server for model aggregation. The training process is equivalent to minimizing a global objective function that captures the accuracy of the predictions of the trained model. The most common tool used for training in DL is decentralized stochastic gradient descent (D-SGD) [10, 11]. D-SGD has two main components: stochastic gradient descent (SGD), and a consensus mechanism. The first one is an iterative method for optimizing an objective function that replaces the gradient for an estimate based on a randomly selected subset of the data. The second one is an information fusion mechanism that causes the agents to reach a state of agreement on a quantity of interest [12]. This mechanism is used in D-SGD to obtain a common model for all users.

Even if DL overcomes the bottleneck problem of FL, and removes the single point of failure by not relying on a server for model aggregation, the communication resources used per iteration could be very high. For this reason, communication-efficient D-SGD has been extensively studied in the literature, mostly focusing on reducing the amount of information transmitted by the users through model compression techniques [13–15]. Recently, another line of work focused on tuning the frequency of information exchange between the users [16, 17]. Along this line, the authors of [18] highlighted the impact of the communication topology in the runtime per iteration of D-SGD, and proposed an algorithm named `MATCHA` to improve the communication efficiency of DL. Such work was further extended by [19], that also included bandwidth and energy consumption as considerations in the communication cost to improve efficiency. In the following thesis, we explore different communication-efficient designs and their impact in the

performance of consensus problems and DL.

1.1 Thesis Outline

In Chapter 2, we provide some preliminary results and definitions of graph theory. In Chapter 3, we introduce the consensus mechanism, and study its convergence conditions and properties in different types of topologies. In Chapter 4, we focus on DL over wireless networks. First, we study the different variants of gradient descent. Then, we focus on D-SGD, and its typical assumptions for convergence. After that, we study the different communication paradigms and provide an overview on wireless networks and communication coordination. In Chapter 5, the included/excluded papers are listed together with the thesis contribution. In Chapter 6, we present the conclusion and mention possible future research directions. Finally, we present the included papers.

Chapter 2

Graph Theory Preliminaries

Graph theory is a branch of mathematics that delves into the study of relationships and connections within diverse systems [20]. A graph, composed of vertices and edges, serves as a fundamental model for representing the intricate web of interactions between entities. Vertices, representing individual elements, are connected by edges, symbolizing relationships that can be either directional or bidirectional. This versatile framework allows us to explore a wide range of phenomena, such as modeling communication systems, which is the focus of this thesis. The applications of graph theory span across various disciplines, making it an indispensable tool for problem-solving and understanding the interconnected nature of systems in both theoretical and practical contexts.

A graph can be defined as an ordered pair $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of vertices (also referred to as nodes or users), and $\mathcal{E} \subseteq \{(i,j) \in \mathcal{V} \times \mathcal{V} | i \neq j\}$ is a set of edges or links. A link is considered undirected if it connects two nodes i and j bidirectionally, and directed if it points from one node i to another node j. In a directed link, the starting point is the node where the link points from and the ending point is the node being pointed to by the link. Based on the classification of all the links in a graph, the graph can also be classified as directed or undirected.

In an undirected graph, the set of nodes sharing a link with a node i is defined as the set of neighbors of node i, denoted as $\mathcal{N}_i = \{j | (i,j) \in \mathcal{E}\}$. The cardinality of this set represents the degree of node i, i.e., $d_i = |\mathcal{N}_i|$. In a directed graph however, each node i has out-degree (the number of links with starting point in node i) and in-degree (the number of links pointing to node i).

2.1 Matrices for Graph Representations

2.1.1 Adjacency Matrix

One way of representing a graph is through matrices. One example is the adjacency matrix, which is a square matrix where the rows and columns correspond to the nodes of the graph.

Consider a graph with N nodes. The adjacency matrix is denoted by $\mathbf{A}$, and has dimension $N \times N$. The elements of $\mathbf{A}$ are defined as follows:

$$A_{ij} = \begin{cases} 1, & \text{if there is a link from node } j \text{ to node } i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

For an undirected graph, the adjacency matrix is symmetric, as the relationship between nodes is mutual. In the case of a directed graph, the matrix need not be symmetric, reflecting the directed nature of the edges. The elements of the adjacency matrix are binary, making it particularly efficient for storage and manipulation.

2.1.2 Incidence Matrix

Another matrix representation of a graph is through the incidence matrix [21], denoted as $\mathbf{B}$. In an incidence matrix, rows correspond to nodes, columns to edges, and the entries indicate whether an edge is incident upon a node. The dimension of $\mathbf{B}$ is $N \times M$, where $M = |\mathcal{E}|$ is the total number of edges in the graph.

If we enumerate the edges in a graph as $e_1, ..., e_M$, then for a directed graph, the incidence matrix is defined as follows:

$$B_{ij} = \begin{cases} 1, & \text{if } s(e_j) = i \\ -1, & \text{if } t(e_j) = i \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $s(e_j)$ and $t(e_j)$ are the starting and ending nodes respectively of the edge e_j. In the case of an undirected graph, the incidence matrix is created using a random orientation on the edges.

2.1.3 Laplacian Matrix

The Laplacian matrix is another matrix representation used in graph theory. It provides insights into the structure and properties of a graph, particularly

focusing on its connectivity. The degree matrix of a graph is a diagonal matrix $\mathbf{D} = \text{diag}(d_1, ..., d_N)$ whose diagonal elements are the nodes degree $d_1, ..., d_N$. For an undirected graph, using the degree and adjacency matrices of a graph, the Laplacian matrix is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{3}$$

The Laplacian matrix of an undirected graph can be obtained also from the incidence matrix (using a random orientation) as follows:

$$\mathbf{L} = \mathbf{B}\mathbf{B}^{\top}. \tag{4}$$

Note that the orientation does not alter the resulting $\mathbf{L}$.

The Laplacian matrix has some important properties, like:

- The second smallest eigenvalue of $\mathbf{L}$ (also known as Fiedler value) is used to study how well connected a graph is (the larger the better connected the graph is) [22].
- The number of zero eigenvalues in the Laplacian corresponds to the number of connected components in the graph [23].

2.2 Centrality Metrics

Centrality metrics are a set of measures used to quantify the importance or influence of nodes within a network. Networks can represent a variety of systems, such as social interactions, transportation systems, or biological processes, and the centrality metrics help identify key nodes that play crucial roles in the network structure. An appropriate centrality metric is subjective, and it will depend on the task at hand, so there are many metrics that can be found in the literature. We denote the centrality of node i as ζ_i. Some commonly used centrality metrics are [24]:

- Degree centrality: this metric is just the degree of each node, i.e., $\zeta_i = d_i$. Basically, it interprets that the more connected a node is, the more important it is, regardless of the importance of the nodes connected to it.
- PageRank centrality: this metric, originally developed by Google for ranking web pages, measures the importance of a node based on the importance of the nodes pointing to it. The centrality of the nodes can be computed either iteratively or using a closed-form solution that is more expensive computationally.

- Betweenness centrality: it measures the extent to which a node lies on the shortest paths between other nodes. It identifies nodes that act as bridges or intermediaries in the network. The betweenness centrality of a node i is defined as the fraction of shortest paths between pairs of other nodes that pass through node i.

Chapter 3

Consensus Mechanisms

Distributed consensus is an iterative algorithm used to reach a state of agreement among a set of users (also referred to as agents) on a quantity of interest [25]. As a special case, *average* consensus mechanisms have as main objective that all users agree on the *average* of their initial values [26, 27]. Distributed and average consensus have become very popular over the last years, and there is an extensive literature on the subject [12, 28–30].

In general, consensus problems are modeled using a graph (directed or undirected), where the nodes represent the users, and the links, their connections. Depending on the application and the specific problem, these connections are kept fixed [31] or may vary over the iterations due to transmission failures or communication designs [32, 33]. In any case, the main goal is to achieve consensus.

3.1 Consensus Over Fixed Topologies

The simplest example is when the topology is fixed, and there are no communication failures. In this case, the information exchange among the users can be represented by a fixed matrix $\mathbf{W}$ known as the weight [29], averaging [34] or mixing matrix [35] in the literature.

Suppose there are N users. The vector $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_N(t)]^\top$, contains the value $x_i(t)$ of each user i at iteration t. $\mathbf{x}(t)$ is updated according to the following:

$$\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t) \tag{5}$$

In such case $\mathbf{W}$ has dimension $N \times N$. As a consequence of (5):

$$\mathbf{x}(t) = \mathbf{W}^t\mathbf{x}(0) \tag{6}$$

From (6), we can observe that to achieve consensus (all users agree on a common value), the mixing matrix needs to satisfy the following:

$$\lim_{t\to\infty} \mathbf{W}^t = \mathbb{1}\mathbf{v}^\top \tag{7}$$

where $\mathbb{1}$ is the column vector of all ones and $\mathbf{v}$ is any vector.

3.1.1 Most Fundamental Case

To facilitate the analysis, a commonly made assumption in the literature is that the mixing matrix is non-negative, i.e., $W_{ij} \geq 0$ for all i, j [33]. Some important definitions are:

Definition 1. A matrix $\mathbf{W}$ is associated to a directed graph $\mathcal{G}_{\mathbf{W}}$, in the sense that there is an edge from node j to node i when $W_{ij} \neq 0$. The matrix $\mathbf{W}$ is *irreducible* if and only if $\mathcal{G}_{\mathbf{W}}$ is strongly connected.

Definition 2. A non-negative matrix is *primitive* if it is irreducible and has only one non-zero eigenvalue of maximum modulus.

Definition 3. The *Perron eigenvalue* r of a matrix $\mathbf{W}$ is a positive real eigenvalue such that any other eigenvalue λ satisfies $|\lambda| < r$.

We have the following result known as the Perron-Frobenius theorem for non-negative and primitive matrices [36]:

Theorem 1. Let $\mathbf{W}$ be a non-negative and primitive matrix, with $\mathbf{w}$ and $\mathbf{v}$ being respectively the right and left eigenvectors (both positive) associated to the Perron eigenvalue r of $\mathbf{W}$, then:

$$\lim_{t\to\infty} \frac{\mathbf{W}^t}{r^t} = \frac{\mathbf{w}\mathbf{v}^\top}{\mathbf{v}^\top\mathbf{w}} \tag{8}$$

Note that since $\mathbf{w}$ and $\mathbf{v}$ are positive, $\mathbf{v}^\top\mathbf{w} \neq 0$. From (7) and (8), we observe that we want $\mathbf{w} = \mathbb{1}$, and $r = 1$. We can conclude that a non-negative matrix $\mathbf{W}$, associated to a strongly connected graph $\mathcal{G}_{\mathbf{W}}$, achieves consensus (satisfies (7)) if $\mathbf{W}\mathbb{1} = \mathbb{1}$, and $\lambda_2(\mathbf{W}) < 1$, where $\lambda_2(\mathbf{W})$ is the second largest eigenvalue, in modulus, of $\mathbf{W}$.

If we impose symmetry in $\mathbf{W}$, i.e., $\mathbf{W}^\top = \mathbf{W}$, then all its eigenvalues are real values, and $\mathbf{v} = \mathbb{1}$. As a consequence, average consensus is reached, since:

$$\lim_{t\to\infty} \mathbf{W}^t = \frac{\mathbb{1}\mathbb{1}^\top}{\mathbb{1}^\top\mathbb{1}} = \frac{1}{N}\mathbb{1}\mathbb{1}^\top. \tag{9}$$

3.1.2 Convergence Speed

The number of iterations needed to achieve consensus is desired to be as small as possible. The relation between $\mathbf{W}$ and the convergence speed is as follows. In every iteration t of the consensus algorithm, the distance to the current average can be defined as [37]:

$$\mathbf{d}(t) = (\mathbf{I} - \mathbf{J})\mathbf{x}(t) \tag{10}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{J} = \frac{1}{N}\mathbb{1}\mathbb{1}^\top$. If we assume that $\mathbf{W}\mathbb{1} = \mathbb{1}$, and $\mathbf{W}^\top = \mathbf{W}$, we have:

$$\mathbf{d}(t+1) = (\mathbf{I} - \mathbf{J})\mathbf{W}\mathbf{x}(t) = (\mathbf{I} - \mathbf{J})\mathbf{W}\mathbf{d}(t), \tag{11}$$

where for the last equality we used that $\mathbf{I} - \mathbf{J} = (\mathbf{I} - \mathbf{J})(\mathbf{I} - \mathbf{J})$, and that $(\mathbf{I} - \mathbf{J})\mathbf{W} = \mathbf{W}(\mathbf{I} - \mathbf{J})$. Taking the Euclidean norm-square of the error:

$$\|\mathbf{d}(t+1)\|^2 = \mathbf{d}^\top(t)\mathcal{W}\mathbf{d}(t) \leq \lambda_1(\mathcal{W})\,\|\mathbf{d}(t)\|^2\,, \tag{12}$$

where $\mathcal{W} = \mathbf{W}(\mathbf{I} - \mathbf{J})\mathbf{W}$, and $\lambda_1(\mathcal{W})$ is the largest eigenvalue (in modulus) of $\mathcal{W}$. The last inequality in (12) comes from the Rayleigh quotient. Consequently,

$$\|\mathbf{d}(t)\|^2 \leq \lambda_1^t(\mathcal{W})\,\|\mathbf{d}(0)\|^2 \tag{13}$$

From (13) we can conclude that if $\lambda_1(\mathcal{W}) < 1$, then consensus is reached. We can also see that the smaller $\lambda_1(\mathcal{W})$, the faster decreases the norm of the distance to the current average per iteration, leading to faster consensus.

Since $\mathcal{W} = (\mathbf{W} - \mathbf{J})^2$, we have that $\lambda_1(\mathcal{W}) = \lambda_1((\mathbf{W} - \mathbf{J})^2) = \|\mathbf{W} - \mathbf{J}\|_2^2$, where the last equality comes from the definition of spectral norm. Minimizing $\lambda_1(\mathcal{W}) = \|\mathbf{W} - \mathbf{J}\|_2^2$ is equivalent to minimizing $\|\mathbf{W} - \mathbf{J}\|_2 = \lambda_1(\mathbf{W} - \mathbf{J})$. Before we continue with the analysis, let us introduce the Wielandt Deflation Theorem [38]:

Theorem 2. Suppose $\lambda_1, \lambda_2, ..., \lambda_N$ are eigenvalues of a matrix $\mathbf{A}$ with associated eigenvectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N$ and that λ_1 has multiplicity 1. If $\mathbf{w}$ is a vector such that $\mathbf{w}^\top\mathbf{v}_1 = 1$, then

$$\mathbf{B} = \mathbf{A} - \lambda_1\mathbf{v}\mathbf{w}^\top$$

has eigenvalues $0, \lambda_2, ..., \lambda_N$.

Using Theorem 2, the matrix $\mathbf{W} - \mathbf{J} = \mathbf{W} - \frac{1}{N}\mathbb{1}\mathbb{1}^\top$ (that using the notation of the theorem would be $\mathbf{v} = \mathbb{1}$, $\mathbf{w} = \frac{1}{N}\mathbb{1}$, and $\lambda_1 = 1$) has as largest

eigenvalue (in modulus) the second largest eigenvalue (in modulus) of $\mathbf{W}$, i.e., $\lambda_1(\mathbf{W}-\mathbf{J}) = \lambda_2(\mathbf{W})$. From this we conclude that the convergence speed of distributed consensus is determined by the second largest eigenvalue of the mixing matrix. It is also clear that if $\lambda_2(\mathbf{W}) < 1$, convergence is guaranteed, as we concluded before.

3.2 Consensus Over Time-Varying Topologies

For time varying topologies, the mixing matrix $\mathbf{W}(t)$ varies over the iterations t. There could be different reasons why the topology varies over time, for example:

- in every iteration, a new graph is generated at random.
- the base topology is fixed, but links might fail due to packet drops, transmission failures or scheduling decisions, leading to varying effective topologies per iteration.

In this thesis we focus on the second reason, specifically the case of a fixed base topology with varying effective topology per iteration due to probabilistic node scheduling. However, we will also explore the first scenario in the section 3.2.1.

If we assume that the matrices $\{\mathbf{W}(t)\}_{t=1}^{\infty}$ are independently and identically distributed (i.i.d.), and like before, we assume that $\mathbf{W}(t)\mathbb{1} = \mathbb{1}$, and $\mathbf{W}^\top(t) = \mathbf{W}(t)$ for all t, then the analysis for the convergence speed remains similar, with:

$$\mathbf{d}(t+1) = (\mathbf{I}-\mathbf{J})\mathbf{W}(t)\mathbf{d}(t). \tag{14}$$

Taking the conditional expectation of $\mathbf{d}(t+1)$ given $\mathbf{d}(t)$:

$$\mathbb{E}[\|\mathbf{d}(t+1)\|^2 \,|\mathbf{d}(t)] = \mathbf{d}^\top(t)\bar{\mathcal{W}}\mathbf{d}(t) \leq \lambda_1(\bar{\mathcal{W}})\,\|\mathbf{d}(t)\|^2\,, \tag{15}$$

where $\bar{\mathcal{W}} = \mathbb{E}[\mathbf{W}^\top(t)\mathbf{W}(t)] - \mathbf{J}$. Repeteadly conditioning and using (15), we obtain the bound:

$$\mathbb{E}[\|\mathbf{d}(t)\|^2] \leq \lambda_1^t(\bar{\mathcal{W}})\,\|\mathbf{d}(0)\|^2\,. \tag{16}$$

Therefore, following a similar analysis as before, $\left\|\mathbb{E}[\mathbf{W}^\top(t)\mathbf{W}(t)] - \mathbf{J}\right\|_2 < 1$ guarantees convergence in the mean-square sense, and its minimization reduces the convergence time of the distributed consensus algorithm.

3.2.1 Consensus Over Random Networks

Before analyzing the convergence of consensus mechanisms over random networks, let us first introduce some relevant concepts:

Definition 4. A matrix $\mathbf{A}$ is said to be *row stochastic* or simply *stochastic* if $\mathbf{A}$ is non-negative and $\mathbf{A}\mathbb{1} = \mathbb{1}$.

Similarly, a *column stochastic* matrix can be defined. If a matrix is row and column stochastic, it is said to be *doubly stochastic*.

Definition 5. A sequence of matrices $\{\mathbf{A}(t)\}_{t=1}^{\infty}$ is said to be *weakly ergodic* if the rows of the product matrix converge to each other as the number of terms in the product grows, i.e., $\lim_{T\to\infty} \prod_{t=1}^{T} \mathbf{A}(t) = \mathbb{1}\mathbf{v}^{\top}$ for some vector $\mathbf{v}$.

Consider the case where in every iteration we have a random network. Suppose we have a random sequence of i.i.d. stochastic mixing matrices $\{\mathbf{W}(t)\}_{t=1}^{\infty}$, each representing the connectivity of its corresponding random graph. If all matrices have positive diagonals, then the following statements are equivalent [28]:

- The random sequence $\{\mathbf{W}(t)\}_{t=1}^{\infty}$ is weakly ergodic almost surely.
- $|\lambda_2(\mathbb{E}[\mathbf{W}(t)])| < 1$.

The first point, according to the definition of weakly ergodic and almost sure convergence, can be written as:

$$\mathbb{P}\left(\lim_{T\to\infty} \prod_{t=1}^{T} \mathbf{W}(t) = \mathbb{1}\mathbf{v}^{\top}\right) = 1 \tag{17}$$

for some vector $\mathbf{v}$. This result tells us that for a given i.i.d. sequence of stochastic mixing matrices with positive diagonals $\{\mathbf{W}(t)\}_{t=1}^{\infty}$, if $|\lambda_2(\mathbb{E}[\mathbf{W}(t)])| < 1$, then $\mathbf{x}(t+1) = \mathbf{W}(t)\mathbf{x}(t)$ reaches consensus almost surely (with probability one).

Note that, since the mixing matrices are only row stochastic, the convergence of the consensus algorithm will not be to the average of the initial values in general. Also, the main diagonal is restricted to be positive for all mixing matrices.

3.2.2 Consensus with Packet Drop Communication

In some practical applications, even if the topology is fixed, the information exchange among the users might fail due to, for example, communication failures. Another example is when using user or device scheduling, and only a fraction of the total users transmit their values in every iteration. In such examples the missing information translates into disconnected links; therefore, time-varying mixing matrices are used to represent the communication among the users in every consensus step.

Two methods proposed to deal with missing information are the *balanced compensation method* and the *biased compensation method* [33]. In both methods, a fixed mixing matrix $\mathbf{W}$ is used to represent the fixed topology, and the difference comes in how the users manage the weights of the missing information. With the balanced compensation method, each user distributes the weights of the missing information equally among the weights of the received ones. On the other hand, with the biased compensation method, each user adds the weights of the missing information to its own weight.

Consider the problem of distributed consensus over a fixed topology with probabilistic node scheduling, i.e., each node has a certain probability of being scheduled for transmission in every iteration, and they are scheduled independently. In this case, since the topology is fixed, we can define $\mathbf{W}$ for a consensus mechanism, but due to the random node scheduling, $\mathbf{W}$ must be modified in every iteration. This leads to $\mathbf{W}(t)$, that can be constructed using the biased compensation method.

The biased compensation method guarantees convergence of the consensus algorithm almost surely and in the mean square sense if $\mathbf{W}$ is such that $W_{ii} > 0$ for all i, and satisfies (7) [33]. However, this convergence is not to the average of the initial values in general.

Chapter 4

Decentralized Learning over Wireless Networks

In general, ML requires the following:

- a training dataset S,
- an ML model with parameter vector $\mathbf{x}$,
- an objective function $F(\mathbf{x}, S)$ that measures the error between the model output and the desired value,
- an iterative method to update the model parameters vector $\mathbf{x}$ such that $F(\mathbf{x}, S)$ is minimized.

The ML model is chosen based on the nature of the problem. The choice of the model depends on factors like the type of task (classification [39], regression [40], etc.) and the characteristics of the data. During training, the model adjusts its internal parameters iteratively to minimize the value of the objective function (the difference between its predicted outputs and the actual labels in the training data). This process is typically done using optimization algorithms, such as gradient descent (GD) or some of its variants [41].

4.1 Variants of Gradient Descent

GD is an optimization algorithm commonly used in deep learning to minimize the cost or loss function during the training of a model. The goal of training a model is to find the optimal set of parameters (weights and biases) that

minimizes the difference between the predicted output and the actual target values.

The term "gradient" refers to the partial derivatives of the loss function with respect to each parameter. The negative gradient points in the direction of the steepest decrease in the loss, so updating the parameters in that direction helps to reach the minimum of the loss function [42]. The iterative process (with iteration index t) for training an ML model with GD is:

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta g(\mathbf{x}(t)) \tag{18}$$

where $g(\mathbf{x}(t))$ denotes the gradient of the objective function, i.e., $g(\mathbf{x}(t)) = \nabla F(\mathbf{x}(t), S)$, and η is a scalar known as learning rate that reduces or increases the effect of the gradient in the update of $\mathbf{x}$.

In an ML problem, the objective function depends on the parameters and the training data. The GD algorithm computes the gradient of the objective function using the entire training dataset for each iteration. This can be computationally expensive, especially when dealing with large datasets. For this reason, other variants of GD, like SGD, uses a single randomly selected sample $\delta \in S$ at each iteration to compute the gradient, i.e., $g(\mathbf{x}(t)) = \nabla F(\mathbf{x}, \delta)$. Another variant of GD is mini-batch stochastic gradient descent (MB-SGD), that uses a small, randomly selected subset $\Delta \subset S$ of the training data at each iteration to obtain the gradient, i.e., $g(\mathbf{x}(t)) = \nabla F(\mathbf{x}, \Delta)$. While SGD is computationally cheap, the fluctuating gradients slow down convergence; therefore, MB-SGD is a better approach in general, since is not as expensive as GD nor so fluctuating as SGD. In Figure 1 we show a comparison between these variants.

4.2 Decentralized Stochastic Gradient Descent

Decentralized Learning (DL) is a type of ML where the data is distributed among different users or nodes in a network. Each user has its own ML model and objective function, and the distributed optimization problem can be defined as:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left(F(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} F_i(\mathbf{x}) \right), \tag{19}$$

where each $F_i : \mathbb{R}^d \to \mathbb{R}$ defines the local objective function of user i, and N is the total number of users. For model training, the local objective F_i can

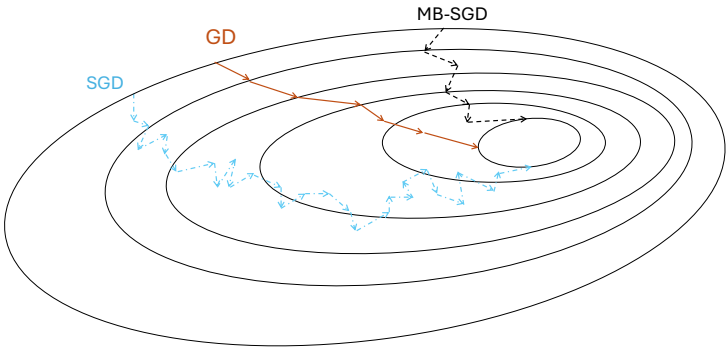


Figure 1: Progression of the error/loss function across iterations t illustrated on the contour $F(\mathbf{x}(t)) = c$, with a decreasing c corresponding to level curves closer to the center.

be defined as the empirical local risk/loss function

$$F_i(\mathbf{x}) := \frac{1}{|\mathcal{D}_i|} \sum_{s \in \mathcal{D}_i} f(\mathbf{x}; s), \tag{20}$$

where $\mathcal{D}_i$ refers to the local data of user i, and $f(\mathbf{x}; s)$ is the loss function evaluated on the model parameters $\mathbf{x}$ and data sample s.

To solve the problem in (19) we can use the first-order optimization algorithm D-SGD. It comprises two algorithms introduced before: stochastic gradient descent (or MB-SGD), and a consensus algorithm. The first part is responsible for using the local data of each user to train its ML model, and the second one forces all users to agree on a common ML model. Another alternative to solve (19) is using decentralized gradient descent, whose convergence was studied in [43]. However, it has a high computational cost. In this thesis, we focus on D-SGD with MB-SGD.

With D-SGD, the model parameter vector $\mathbf{x}_i(t)$ of each user i in every iteration t is updated in two steps. First the stochastic gradient update:

$$\mathbf{x}_i\left(t + \frac{1}{2}\right) = \mathbf{x}_i(t) - \eta \mathbf{g}_i(t), \tag{21}$$

where $\mathbf{g}_i(t)$ represents the stochastic gradient of node i computed based on a mini-batch of samples drawn from its local dataset, i.e., $\mathbf{g}_i(t) = \frac{1}{|\xi_i(t)|} \sum_{s \in \xi_i(t)} \nabla f(\mathbf{x}_i(t); s)$ with $\xi_i(t) \subseteq \mathcal{D}_i$ drawn at random. The second

step is the consensus update:

$$\mathbf{x}_i(t+1) = \sum_{j=1}^{N} W_{ij}(t)\mathbf{x}_j\left(t+\frac{1}{2}\right). \tag{22}$$

Typical assumptions made in the non-convex decentralized optimization literature on the local objective functions are the following [11, 15, 44, 45]:

Assumption 1. Each local objective function $F_i(\mathbf{x})$ is differentiable and its gradient is l-Lipschitz, i.e., $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq l\|\mathbf{x}-\mathbf{y}\|, \forall i \in \{1,2,...,N\}$.

Assumption 2. Stochastic gradients at each node are unbiased estimates of the true local gradient of the local objectives, i.e., $\mathbb{E}[\mathbf{g}_i] = \nabla F_i(\mathbf{x}_i)$.

Assumption 3. The variance of the stochastic gradient at each node is uniformly bounded, i.e., $\mathbb{E}[\|\nabla f(\mathbf{x},s) - \nabla F_i(\mathbf{x})\|^2] \leq \sigma^2, \forall i \in \{1,2,...,N\}$.

Assumption 4. The deviation between the local gradients and the global gradient is bounded by a non-negative constant, i.e., $\frac{1}{N}\sum_{i=1}^{N}\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \zeta^2$.

Assumptions on the mixing matrix are also made. For example, in the case of fixed topologies, most works assume that the mixing matrix $\mathbf{W}$ is symmetric, doubly stochastic, and such that $|\lambda_2(\mathbf{W})| < 1$ [45–47]. These assumptions, as shown in Chapter 3, are sufficient conditions to achieve average consensus.

In this thesis we focus on the particular case of time-varying topologies with the mixing matrix randomly chosen in every iteration from a family of possible mixing matrices. In this framework we find the following assumptions [19]:

Assumption 5. For a total number of iterations T, the mixing matrices $\{\mathbf{W}(t)\}_{t=1}^{T}$ are independently and identically distributed, with $\left\|\mathbb{E}[\mathbf{W}^\top(t)\mathbf{W}(t)] - \mathbf{J}\right\|_2 < 1$.

Assumption 6. Every mixing matrix $\mathbf{W}(t)$ is symmetric with each row/column summing up to one, i.e., $\mathbf{W}^\top(t) = \mathbf{W}(t), \mathbf{W}(t)\mathbb{1} = \mathbb{1}, \forall t$.

Note that Assumption 6 does not require each $\mathbf{W}(t)$ to have non-negative entries. Since the objective function $F(\mathbf{x})$ can be non-convex, we cannot guar-

antee convergence to the global optimum. Instead, an upper bound on the averaged gradient norm $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla F(\bar{\mathbf{x}}(t))\right\|^2\right]$, where $\bar{\mathbf{x}}(t)=\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i(t)$, was provided in [18,19]. This bound depends on $\left\|\mathbb{E}[\mathbf{W}^\top(t)\mathbf{W}(t)]-\mathbf{J}\right\|_2$, and its minimization is key in reducing the convergence time of D-SGD. When $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla F(\bar{\mathbf{x}}(t))\right\|^2\right]$ approaches zero, the D-SGD algorithm converges to a stationary point.

4.3 Communication Paradigms

In general, communication among users can be classified into unicast, where the communication is from one-to-one, multicast, with one-to-many communication, and broadcast as a special case of multicast [48]. Each type has its own characteristics and applications.

4.3.1 Unicast Communication

Unicast communication is a one-to-one type of communication where data is sent from one sender to one receiver. One consequence of this is that to obtain bidirectional exchange of information, two unicast communications are required. Unicast communication is the most common type of communication in computer networks and it is used for tasks such as web browsing, email, and file transfer. A visual example is provided in Figure 2(b). The main limitation is its scalability, since in a large network requiring users to communicate with multiple others constantly, the communication could become time-consuming.

4.3.2 Multicast Communication

Multicast communication involves sending data from one sender to multiple, but not necessarily all, receivers in the network. It can be seen as an extension of unicast communication. Note that, like before, transmission occurs in a single direction, and bi-directional communication requires multiple transmissions with different senders. Multicast communication is used for applications such as streaming media, online gaming, and video conferencing. See Figure 2(c) for an example.

4.3.3 Broadcast Communication

Broadcast communication is a special case of multicast communication, where the transmitted message is received by all users in the network. However, in this work we assume that due to distance, or communication blockage,

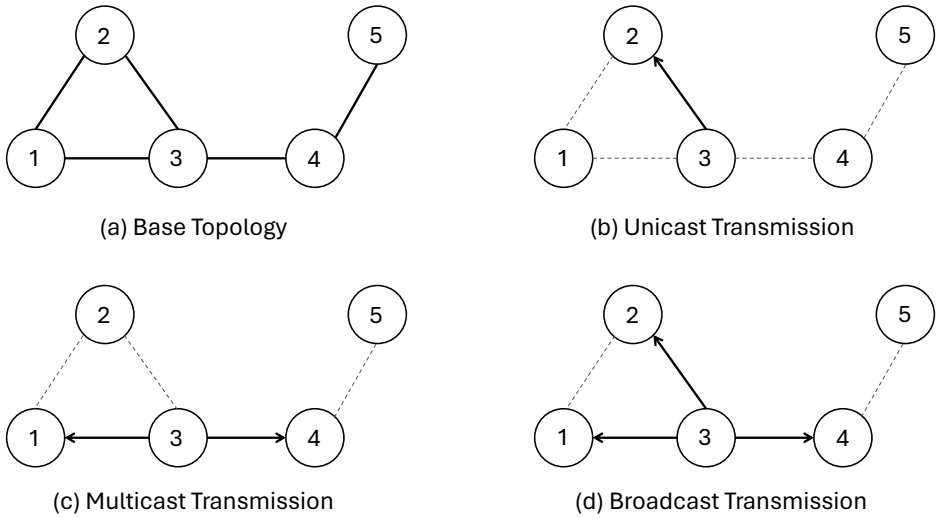


Figure 2: Comparison between different types of communication in the topology in (a) with node 3 as transmitter. The dashed lines represent inactive links.

some users in the network are unable to decode the received signal, and we consider it as not received. Two nodes that are not connected in the graph representation of the network cannot communicate to each other, meaning that if either transmits a signal, it will not be successfully decoded by the other. An example is provided in Figure 2(d), where node 3 transmits, and all its neighbors receive the information. Broadcast communication can be useful for tasks such as distributing updates or announcements.

4.4 Overview of Wireless Networks

A wireless network is a type of network where devices communicate with each other without physical cables. Instead, communication occurs over the air using radio waves. Wireless networks are an example of multiaccess communication, where multiple devices can access the channel simultaneously [49]. One of its distinctive features is that the communication channels are inherently broadcast channels, and when a device accesses the channel and transmits, all the devices in its transmission range are able to receive the transmitted information. Such transmissions are known as broadcast transmissions. The transmission range varies depending on the transmission power of the devices and the channel condition. Our focus is on the commu-

nication coordination, and not on the physical characteristics of the wireless medium.

4.4.1 Communication Model with Packet Success / Collision

The topology of a wireless network can be represented by a directed graph, where there is an edge from node i to node j if j is in the transmission range of node i. In this thesis, we make the following assumption: if node i is in the transmission range of node j, then j is also in the transmission range of node i. This allows us to represent the topology of the network using an undirected graph.

We make the following assumption on the communication in this multiaccess medium. If a device/node i transmits, node j will receive its information if:

- node j is in the transmission range of node i, i.e., there is a link between nodes i and j,
- no other node k such that there is a link between nodes j and k is transmitting, and
- node j is not transmitting at the same time as node i.

Since if all nodes broadcast at the same time using the same frequency band, the information reception will fail due to packet collisions, coordination is necessary to determine when and how one user can access the channel and transmit (receive) information to (from) another user.

4.4.2 Wireless Networks Multiple Access Coordination

Two approaches for coordinating multiple access communication in wireless networks are random access [50], which is prone to collisions, and orthogonal division of resources. In this thesis we focus on the second approach, that involves dividing the available spectrum into non-overlapping and orthogonal channels. By leveraging orthogonality, devices can access these channels concurrently without causing interference, thereby enhancing overall system performance. Two common techniques for orthogonal division of resources are:

- Frequency-division multiple access (FDMA), where the frequency spectrum is divided into distinct frequency bands or channels. Each channel serves as an independent resource, allowing devices to communicate

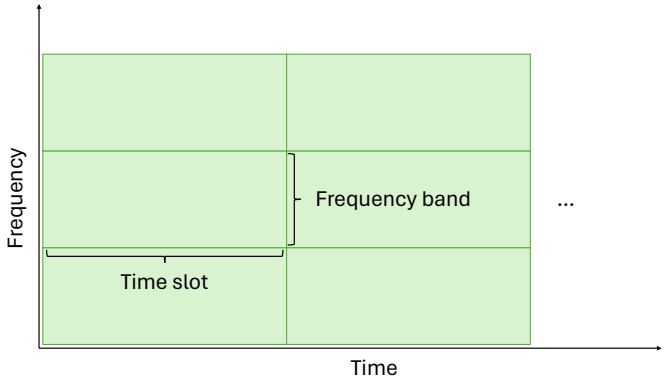


Figure 3: Orthogonal division of resources

simultaneously without interference as long as they operate on different channels.

- Time-division multiple access (TDMA), where different time slots are allocated within a shared frequency band to various users. Each device transmits in its designated time slot, ensuring temporal orthogonality.

An illustration is provided in Figure 3.

Note that with TDMA, one node at the time accesses the channel, i.e., one node per time slot. However, this approach is not as efficient as if we reuse resources spatially by allowing users/nodes that are geographically far away access the channel simultaneously (at the same time and using the same frequency). For example, if we consider the topology in Figure 2(a), nodes 2 and 5 could access the channel simultaneously with no collisions. Nodes 3 and 5, however, should not access the channel simultaneously, since there would be a collision at node 4.

In this thesis, we consider the case of using a single frequency band, and reusing resources spatially (time slots). We use a perfect scheduling strategy to assign to each group of users their corresponding time slot in every iteration. We explore the impact of partial communication in the convergence speed of D-SGD. Partial communication refers to the case where, in every iteration of the training process, not all users, but only a fraction of them, exchange their parameter vector with their neighbors. We measure the convergence speed as the number of time slots (also referred to as transmission slots) incurred in the training process.

Chapter 5

Contributions of the Thesis

This thesis focuses on the impact of partial communication on the convergence speed of consensus and decentralized learning problems. In Paper A, we show the advantage of using partial communication with centrality metrics-based probabilistic nodes scheduling as compared to full communication in a distributed consensus problem over wireless networks. We also propose a pre-compensation method to remove the bias of the final state of the consensus problem. In Paper B, we propose two methods for partial communication over wireless networks for a decentralized learning problem. We also optimized the entries of the mixing matrices and their sampling probabilities for a given communication budget.

5.1 Included Papers

Paper A: Distributed Consensus in Wireless Networks With Probabilistic Broadcast Scheduling

Authored by: Daniel Pérez Herrera , Zheng Chen , and Erik G. Larsson

Published in IEEE Signal Processing Letters, vol. 30, pp. 41–45, 2023.

Abstract: We consider distributed average consensus in a wireless network with partial communication to reduce the number of transmissions in every iteration/round. Considering the broadcast nature of wireless channels, we propose a probabilistic approach that schedules a subset of nodes for broadcasting information to their neighbors in every round.We compare several heuristic methods for assigning the node broadcast probabilities under a fixed number of transmissions per round. Furthermore, we introduce a pre-compensation method to correct the bias between the consensus value and the

average of the initial values, and suggest possible extensions for our design. Our results are particularly relevant for developing communication-efficient consensus protocols in a wireless environment with limited frequency/time resources.

Paper B: Faster Convergence with Less Communication: Broadcast-Based Subgraph Sampling for Decentralized Learning over Wireless Networks

Authored by: Daniel Pérez Herrera, Zheng Chen, and Erik G. Larsson

Submitted to IEEE Transactions on Communications, February 2024.

Abstract: Consensus-based decentralized stochastic gradient descent (D-SGD) is a widely adopted algorithm for decentralized training of machine learning models across networked agents. A crucial part of D-SGD is the consensus-based model averaging, which heavily relies on information exchange and fusion among the nodes. For consensus averaging over wireless networks, due to the broadcast nature of wireless channels, simultaneous transmissions from multiple nodes may cause packet collisions if they share a common receiver. Therefore, communication coordination is necessary to determine when and how a node can transmit (or receive) information to (or from) its neighbors. In this work, we propose `BASS`, a broadcast-based subgraph sampling method designed to accelerate the convergence of D-SGD while considering the actual communication cost per iteration. `BASS` creates a set of mixing matrix candidates that represent sparser subgraphs of the base topology. In each consensus iteration, one mixing matrix is sampled, leading to a specific scheduling decision that activates multiple collision-free subsets of nodes. The sampling occurs in a probabilistic manner, and the elements of the mixing matrices, along with their sampling probabilities, are jointly optimized. Simulation results demonstrate that `BASS` enables faster convergence with fewer transmission slots compared to existing link-based scheduling methods. In conclusion, the inherent broadcasting nature of wireless channels offers intrinsic advantages in accelerating the convergence of decentralized optimization and learning.

5.2 Excluded Papers

The following paper is excluded from the thesis due to it being superfluous.

Decentralized Learning over Wireless Networks with Broadcast-Based Subgraph Sampling

Authored by: Daniel Pérez Herrera , Zheng Chen , and Erik G. Larsson

Accepted for publication in the IEEE International Conference on Communications, 2024.

This paper contains preliminary results of Paper B.

Abstract: This work focuses on the communication perspective of decentralized learning over wireless networks, using consensus-based decentralized stochastic gradient descent (D-SGD). Considering the actual communication cost or delay caused by in-network information exchange in every iteration, our goal is to achieve fast convergence of the algorithm measured by improvement per transmission slot. We propose BASS, an efficient communication framework for D-SGD over wireless networks with broadcast-based subgraph sampling. More explicitly, in every iteration, we activate multiple subsets of non-interfering nodes to broadcast model updates to their neighbors. These subsets are activated randomly over time with some probabilities under a given communication cost (e.g., number of transmission slots per iteration). During the consensus update step, only bi-directional links are effectively considered to preserve the communication symmetry. As compared to existing link-based scheduling methods, the broadcasting nature of wireless channels provides inherent advantages in speeding up convergence of decentralized learning by creating more communicated links under the same number of transmission slots.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Combining the insights obtained from the analysis of distributed average consensus and decentralized stochastic gradient descent over wireless networks, this thesis contributes to the advancement of efficient communication protocols in distributed systems.

In the study of distributed average consensus, a novel approach employing partial communication through probabilistic broadcast scheduling was proposed. Through simulations, a trade-off between convergence speed and consensus value bias was revealed, highlighting the efficacy of leveraging partial communication to achieve consensus with reduced communication costs. Various heuristic methods for assigning broadcast probabilities and a pre-compensation mechanism for bias correction were introduced, paving the way for further investigation into optimal scheduling strategies.

Shifting focus to decentralized learning, the thesis presented `BASS`, a framework designed to accelerate the convergence of D-SGD by customizing communication dynamics and resource utilization. By sampling subgraphs of the network topology and leveraging broadcast transmission, `BASS` outperforms existing scheduling policies, achieving faster convergence with fewer transmission slots. These findings underscore the potential of exploiting network connectivity and data heterogeneity in node scheduling design, suggesting avenues for future exploration in optimizing communication budget allocation and adapting to dynamic network conditions.

6.2 Future Work

Some potential directions for future work are the following:

- Enhancing scheduling decisions by considering factors beyond graph connectivity in partial communication, such as data heterogeneity among users.
- Optimize the communication budget allocation over time, using an adaptive or event-triggered design.
- Extend to the case where different users work on different tasks, i.e., different learning models. In this case the communication among users must consider this difference.

In future research, building on these ideas can lead to even more efficient communication protocols, enhancing the overall performance of DL.

Bibliography

[1] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 2018, pp. 210–215.

[2] M. Kubat and J. Kubat, *An introduction to machine learning*. Springer, 2017, vol. 2.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[4] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and trends® in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[8] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[9] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2021.

[10] B. Swenson, R. Murray, S. Kar, and H. V. Poor, "Distributed stochastic gradient descent: Nonconvexity, nonsmoothness, and convergence to local minima," *arXiv preprint arXiv:2003.02818*, 2020.

[11] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[12] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[13] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *Proceedings of the 8th International Conference on Learning Representations*, 2019.

[14] Y. Lu and C. De Sa, "Moniqua: Modulo quantized communication in decentralized sgd," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6415–6425.

[15] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[16] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 212–229, 2019.

[17] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE conference on computer communications*, 2019, pp. 1387–1395.

[18] J. Wang, A. K. Sahu, G. Joshi, and S. Kar, "MATCHA: A matching-based link scheduling strategy to speed up distributed optimization," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5208–5221, 2022.

[19] C.-C. Chiu, X. Zhang, T. He, S. Wang, and A. Swami, "Laplacian matrix sampling for communication-efficient decentralized learning," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 887–901, 2023.

[20] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[21] J. W. Grossman, D. M. Kulkarni, and I. E. Schochetman, "Algebraic graph theory without orientation," *Linear Algebra and its Applications*, vol. 212, pp. 289–307, 1994.

[22] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[23] A. E. Brouwer and W. H. Haemers, *Spectra of graphs*. Springer Science & Business Media, 2011.

[24] V. Latora, V. Nicosia, and G. Russo, *Complex networks: principles, methods and applications*. Cambridge University Press, 2017.

[25] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical association*, vol. 69, no. 345, pp. 118–121, 1974.

[26] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM journal on control and optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[27] H. Wang, X. Liao, and T. Huang, "Average consensus in sensor networks via broadcast multi-gossip algorithms," *Neurocomputing*, vol. 117, pp. 150–160, 2013.

[28] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 791–795, 2008.

[29] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, 2009.

[30] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 634–649, 2008.

[31] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[32] Y. Chen, R. Tron, A. Terzis, and R. Vidal, "Corrective consensus: Converging to the exact average," in *49th Conference on Decision and Control*. IEEE, 2010, pp. 1221–1228.

[33] F. Fagnani and S. Zampieri, "Average consensus with packet drop communication," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 102–133, 2009.

[34] T. C. Aysal, A. D. Sarwate, and A. G. Dimakis, "Reaching consensus in wireless networks with probabilistic broadcast," in *47th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2009, pp. 732–739.

[35] C. Asensio-Marco and B. Beferull-Lozano, "Accelerating consensus gossip algorithms: Sparsifying networks can be good for you," in *International Conference on Communications*. IEEE, 2010, pp. 1–5.

[36] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[37] S. S. Pereira and A. Pages-Zamora, "Consensus in correlated random wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6279–6284, 2011.

[38] A. Björck and G. Dahlquist, "Numerical methods in scientific computing volume ii," *Society for Industrial and Applied Mathematics, Philadelphia, PA*, 2008.

[39] A. A. Soofi and A. Awan, "Classification techniques in machine learning: applications and issues," *J. Basic Appl. Sci*, vol. 13, no. 1, pp. 459–465, 2017.

[40] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 140–147, 2020.

[41] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[42] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[43] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[44] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE journal of selected topics in signal processing*, vol. 16, no. 3, pp. 487–500, 2022.

[45] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3478–3487.

[46] J. Wang and G. Joshi, "Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms," *Journal of Machine Learning Research*, vol. 22, no. 213, pp. 1–50, 2021.

[47] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3723–3741, 2021.

[48] R. Wittmann and M. Zitterbart, *Multicast communication: Protocols, programming, & applications*. Elsevier, 2000.

[49] D. Bertsekas and R. Gallager, *Data networks*. Athena Scientific, 2021.

[50] Z. Chen, M. Dahl, and E. G. Larsson, "Decentralized learning over wireless networks: The effect of broadcast with random access," in *24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2023, pp. 316–320.

Included Papers

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

https://doi.org/10.3384/9789180757867

Other Recently Published Theses From

The Division of Communication Systems
Department of Electrical Engineering (ISY)
Linköping University, Sweden

Oksana Moryakova, *Contributions to Efficient Design and Implementation of Variable Digital Filters*, Linköping Studies in Science and Technology. Licentiate Thesis, No. 2002, 2024.

Unnikrishnan Kunnath Ganesan, *Beyond Boundaries: Evolving Connectivity with Massive MIMO*, Linköping Studies in Science and Technology. Dissertations, No. 2395, 2024.

Olle Abrahamsson, *On Aggregation and Dynamics of Opinions in Complex Networks*, Linköping Studies in Science and Technology. Licentiate Thesis, No. 1990, 2024.

Ahmet Kaplan, *Signal Processing Aspects of Bistatic Backscatter Communication*, Linköping Studies in Science and Technology. Licentiate Thesis, No. 1989, 2024.

Chung-Hsuan Hu, *Communication-Efficient Resource Allocation for Wireless Federated Learning Systems*, Linköping Studies in Science and Technology. Licentiate Thesis, No. 1969, 2023.

Ziya Gülgün, *GNSS and Massive MIMO: Spoofing, Jamming and Robust Receiver Designfor Impulsive Noise*, Linköping Studies in Science and Technology. Dissertations, No. 2310, 2023.

Ema Becirovic, *Signal Processing Aspects of Massive MIMO*, Linköping Studies in Science and Technology. Dissertations, No. 2251, 2022.

Zakir Hussain Shaik, *Cell-Free Massive MIMO: Distributed Signal Processing and Energy Efficiency*, Linköping Studies in Science and Technology. Licentiate Thesis, No. 1924, 2022.

Unnikrishnan Kunnath Ganesan, *Distributed Massive MIMO: Random Access, Extreme Multiplexing and Synchronization*, Linköping Studies in Science and Technology. Licentiate Thesis, No. 1923, 2022.

Özgecan Özdogan, *Signal Processing Aspects of Massive MIMO and IRS-Aided Communications*, Linköping Studies in Science and Technology. Dissertations, No. 2199, 2022.

Amin Ghazanfari, *Multi-Cell Massive MIMO: Power Control and Channel Estimation*, Linköping Studies in Science and Technology. Dissertations, No. 2142, 2021.

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology,
Licentiate Thesis No. 2004, 2024

Division of Communication Systems
Department of Electrical Engineering

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se