



INSTITUT
POLYTECHNIQUE
DE PARIS



NNT : 2025IPPAT007

Fundamental Limits and Practical Algorithms for Wireless Distributed Computation and Estimation Systems

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris
en cotutelle avec Shanghai Jiao Tong University

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Réseaux, Information et Communications

Thèse présentée et soutenue à Shanghai, le 10 mars 2025, par

YUE BI

Composition du Jury :

Xin Wang Professeur, Fudan University	Président
Inbar Fijalkow Professeure, École Nationale Supérieure de l'Électronique et de ses Applications	Rapporteur
Angela Yingjun Zhang Professeure, The Chinese University of Hong Kong	Rapporteur
Eduard Jorswieck Professeur, Technische Universität Braunschweig	Rapporteur
Derya Malak Maîtresse de conférences, EURECOM	Examineur
Cunqing HUA Professeur, Shanghai Jiao Tong University	Examineur
Philippe Ciblat Professeur, Télécom Paris	Directeur de thèse (IP Paris)
Yue Wu Professeur, Shanghai Jiao Tong University	Directeur de thèse (SJTU)

Thèse de doctorat

To my beloved parents,

for being my unwavering source of strength and support. I am forever grateful for all that you have done for me. Though you may humbly underestimate your own greatness, in my heart, you are the most incredible parents one could ever ask for.

To my fiancée,

for her steadfast support and understanding throughout my studies. Your presence in my life since high school has been a constant source of joy and inspiration. The three wonderful years we shared in France have been an unforgettable and enriching chapter of our journey together.

Acknowledgements

As my joint PhD program comes to an end, I reflect on the past four and a half years—a journey filled with challenges, growth, and invaluable experiences. I am deeply grateful to the mentors, colleagues, and friends who have guided and supported me throughout this time.

First, I extend my heartfelt gratitude to my thesis supervisor, *Professor Philippe Ciblat*. His profound insights, unwavering encouragement, and meticulous attention to detail have been instrumental in shaping my research. Even in the most challenging moments, his guidance and patience never wavered, inspiring me to strive for excellence.

I am also sincerely grateful *Professor Michèle Wigger*, whose insightful advice and steadfast support have played a crucial role in both my research and my growth as a scholar. Her expertise and guidance provided a solid foundation for this thesis.

Beyond academic mentorship, they also helped me adapt to life in France, sharing their knowledge of local culture and history, which enriched my experience and deepened my connection to this land.

A special thanks goes to my co-supervisor, *Professor Yue Wu*, whose mentorship extended far beyond academia. His invaluable advice and unwavering support have guided me not only in research but also in my personal growth.

I am also indebted to the members of my jury, notably the reviewers, *Professor Inbar Fijalkow*, *Professor Angela Yingjun Zhang* and *Professor Eduard Jorswieck*, as well as examiners, *Professor Xin Wang*, *Doctor Derya Malak* and *Professor Cunqing Hua*, who dedicated their time and expertise to refining my work. Their constructive feedback and rigorous evaluations have significantly improved the quality of this thesis.

Finally, I would like to express my sincere appreciation to my fellow researchers at IP Paris LTCI and the Wireless and Mobile IoT Security Lab at SJTU, as well as everyone who has supported me along the way. Their encouragement and companionship have made this journey all the more meaningful and unforgettable.

Contents

List of Figures	ix
List of Tables	xii
List of Terms	xv
Abstract	1
Résumé (Abstract in French)	3
摘要 (Abstract in Chinese)	5
1 General Introduction	7
1.1 Context and Objectives of the Thesis	7
1.2 Related Works	9
1.3 Main Contributions and Organization	12
1.4 Notations	14
2 Degrees of Freedom of Channels with Applications to Wireless Distributed Computing (DC)	17
2.1 Introduction	17
2.2 System Models of Partially Connected Channels	21
2.2.1 Non-cooperative Partially Connected Channel (NPC)	22
2.2.2 Cooperative Partially Connected X-Channel (CPXC)	23
2.2.3 Cooperative Partially Connected Interference Channel (CPIC)	23
2.3 Main Results	24
2.4 Application to Generic Wireless DC	26
2.4.1 The MapReduce System	26
2.4.2 An Illustrative Example with $K = 6$ and $r = 2$	28
2.4.3 Results on Normalized Delivery Time (NDT)	28
2.5 Application to Wireless DC with Associative Reduce Function	31
2.5.1 An Illustrative Example with $K = 6$	31
2.5.2 General Results	32
2.6 Proof of the SDoF Bounds for NPC, CPXC and CPIX	34
2.6.1 Proof of the SDoF Lower Bound for NPC (Theorem 2.1)	34

2.6.2	Proof of the SDoF Lower Bound for CPXC (Theorem 2.2)	37
2.6.3	Proof of the SDoF Upper Bounds for CPXC and CPIC (Theorem 2.2 and 2.3)	42
2.7	Conclusion	45
3	Bounds on NDT of Wireless DC	47
3.1	Introduction	47
3.2	Wireless MapReduce Systems	49
3.2.1	Sufficiency of Symmetric File Assignments	49
3.2.2	Relation to the Network's SDoF with r -fold Cooperation	50
3.3	Main Results	51
3.3.1	Comparison to Previous Upper Bounds	52
3.4	Examples of our IA Scheme without Zero-forcing (ZF)	55
3.4.1	Example 1: $K \geq 3, r = 1$	55
3.4.2	Example 2: $K = 4, r = 2$	58
3.5	The General IA-Scheme without ZF	63
3.6	A Scheme with IA and ZF for K odd and $r = (K - 1)/2$	69
3.7	Proof of the NDT Lower Bound in Theorem 3.1	75
3.8	Conclusion	79
4	Multi-bit Quantizer Optimization for Distributed Estimation	81
4.1	Introduction	81
4.2	Preliminaries	82
4.2.1	Geometric Programming (GP)	82
4.2.2	Signomial Geometric Programming (SGP)	83
4.3	System Model	86
4.4	Non-asymptotic Cramer-Rao Bound	87
4.5	Asymptotic Cramer-Rao Bound	89
4.5.1	Optimized Quantizer with SGP	89
4.5.2	Integral-preserving Discretization	92
4.6	Numerical Results	96
4.7	Conclusion	99
5	Stochastic Activation Broadcast Sum-weight for Distributed Estimation	101
5.1	Introduction	101
5.2	Preliminaries	102
5.3	System Model	103
5.3.1	Graph-based Network	103
5.3.2	Node Activation	104
5.3.3	Communication Failure	104
5.3.4	Sensor Observation and Estimation	105
5.4	Broadcast Sum-weight Framework with Stochastic Activation	106
5.4.1	Average Conservation and Consensus	106
5.4.2	Framework Design	107

5.5	Convergence Rate Analysis	109
5.6	Numerical Results	112
5.7	Conclusion	115
6	Conclusions and Perspectives	119
	Appendix	122
A	Supplementary Proof of Theorem 2.2: Full-Rank Property for Submatrices of $\hat{\mathbf{D}}_p$	125
B	Supplementary Proofs of Theorem 3.1	129
B.1	Proof of Monotonicity and Convexity of values $C_i^{(t)}$	129
B.2	Proof of Structure of Minimizer	129
	Bibliography	131
	Publications	138

List of Figures

2.1	An example of PC model for $K = 6$, $r = 2$ and the matrices \mathbf{M} , \mathbf{N} in (2.1). The solid arrow represent the message passing, while the dash line indicates that the Tx group cause interference to the Rx.	18
2.2	CPXC model for $K = 6$, $r = 2$	19
2.3	The cooperative interference channel model for $K = 6$, $r = 2$	19
2.4	Illustrative Example with $K = 6$ and $r = 2$	28
2.5	Upper bounds on $\Delta^*(r)$ for the TDMA scheme, the one-shot scheme in [26], the classic IA scheme and our new IA scheme when $K = 24$	31
2.6	Illustrative Example with $K = 6$	32
3.1	Bounds on $\Delta^*(r)$ from Theorem 3.1 compared to the optimal zero-forcing and interference cancellation scheme in [26] and to the upper bound obtained by the IA scheme in Theorem 2.4 Theorem 2.4 for $K = 11$	54
3.2	Bounds on $\Delta^*(r)$ from Theorem 3.1 compared to the optimal NDT achieved by the optimal zero-forcing and interference cancellation scheme in [26] and to the upper bound obtained by the IA scheme in Theorem 2.4 for $K = 20$	55
4.1	Comparison between the intuitive quantization and our method of quanti- zation with $L = 3$ and $L = 4$	95
4.2	Optimized distribution $g(\tau)$ obtained by polynomial fitting based on discrete solution of Problem 4.11.	96
4.3	Average CRB vs K (SNR=8dB).	97
4.4	CRB versus SNR ($K = 2000$).	100
5.1	Example graph	108
5.2	Actual message flow graph	109
5.3	Connectivity and edge density with different r and K	113
5.4	Slope versus ω for different γ and α with $K = 10$	115
5.5	Average slope, ω and empirical ω over graphs versus γ for different α with $K = 10$	116
5.6	Simulation results of Broadcast Sum-Weight on MSE'_t with different α , γ with $K = 10$	116
5.7	Simulation results of Broadcast Sum-Weight on MSE_t for different α , γ with $K = 10$	117

5.8 The average of the optimal γ versus the communication range r 117

List of Tables

2.1	Table of bounds for three channels.	26
3.1	Table showing the precoding matrix used to send each message M_k^j	56
3.2	Messages $M_{k,\mathcal{T}}^j$ precoded by the three precoding matrices $\mathbf{U}_{\{2,3\}}$, $\mathbf{U}_{\{2,4\}}$, and $\mathbf{U}_{\{3,4\}}$	60
3.3	Let $r = 2$. The table illustrates the codesymbols $\mathbf{b}_{k,\mathcal{T}}^j$ that are premultiplied by the precoding matrix $\mathbf{U}_{\{2,3\}}$. Entries for sets \mathcal{T} either equal to $\{2, 3\}$ or containing 1, correspond to r transmitted codewords, one from each node in \mathcal{T} . All other entries correspond only to a single codeword from the node not in $\{2, 3\}$	64
3.4	Let $r = 3$. The table illustrates the codesymbols $\mathbf{b}_{k,\mathcal{T}}^j$ that are premultiplied by the precoding matrix $\mathbf{U}_{\{2,3,4\}}$. Entries for sets \mathcal{T} either equal to $\{2, 3, 4\}$ or containing 1, correspond to r transmitted codewords, one from each node in \mathcal{T} . All other entries correspond only to a single codeword from the node not in $\{2, 3, 4\}$	65
3.5	Choice of precoding matrices in our scheme for $K = 5$ and $r = 2$. Each signal that is precoded with matrix \mathbf{U}_ℓ is zero-forced at the unique Node $i \neq \ell$ that is neither the intended Node j nor part of the transmitting set \mathcal{T}	72
3.6	Codewords precoded by matrix \mathbf{U}_7 when $K = 7$ and $r = 3$. Each codeword is zero-forced at the two nodes not belonging to the transmit \mathcal{T} and not equal to the receive node j or to 7.	80

List of Terms

CDC	Coded Distributed Computing
CGP	Complementary Geometric Programming
CPIC	Cooperative Partially-connected Interference Channel
CPXC	Cooperative Partially-connected X-Channel
CRB	Cramér-Rao Bound
CSI	Channel State Information
CSIT	Channel State Information at Transmitters
DC	Distributed Computing
DoF	Degrees of Freedom
FC	Fusion Center
GP	Geometric Programming
IA	Interference Alignment
IC	Interference Channel
IVA	Intermediate Value
MAC	Multiple Access Channel
MSE	Mean Square Error
NDT	Normalized Delivery Time
NPC	Non-cooperative Partially-connected Channel
PC	Partially-connected Channel
RGG	Random Geometric Graph

Rx Receiver

SDoF Sum Degrees of Freedom

SGP Signomial Geometric Programming

SNR Signal-to-Noise Ratio

Tx Transmitter

Abstract

Distributed systems form the backbone of modern computing applications, enabling collaborative and efficient task execution across multiple networked components. However, the physical separation of system components introduces significant challenges in terms of communication efficiency. This thesis addresses these challenges by investigating information-theoretic limits and proposing practical schemes to enhance the performance of two types of systems: distributed computing (DC) systems and distributed estimation systems.

DC systems exploit task parallelization to significantly reduce execution time of computationally intensive tasks. A popular framework for DC systems, MapReduce, divides computation into three phases: map, shuffle, and reduce. The shuffle phase, which involves transferring intermediate values between nodes, is often the bottleneck in terms of execution time. While various coding schemes have been proposed to optimize the shuffle phase in wired networks, the growing adoption of wireless DC systems, necessitates new solutions tailored to wireless environments.

Distributed estimation systems, on the other hand, aim to estimate a shared parameter collaboratively across multiple nodes. These systems are widely used in sensor networks, environmental monitoring, and distributed machine learning. Communication efficiency is essential for accurate data fusion, since inaccurate data loss can degrade system performance. This thesis focuses on two scenarios: systems with a fusion center and those without.

The main contributions of this thesis are as follows:

- The thesis investigates partially connected wireless networks, introducing new coding schemes that optimize the computation-communication tradeoff for MapReduce frameworks. Specifically, by analyzing the sum degrees of freedom (SDoF) of partially connected channels, the work derives achievable lower bounds using interference alignment (IA). An information-theoretic analysis is also provided in partially connected wireless networks.
- The thesis extends IA techniques to optimize wireless MapReduce systems operating over full-duplex interference channels. By designing schemes tailored for specific network configurations, the proposed methods achieve significant improvements in execution time compared to traditional approaches. The work also establishes theoretical upper and lower bounds for the computation-communication tradeoff.
- For distributed estimation systems requiring a fusion center, the thesis develops a framework to design multi-bit quantizers that minimize the Cramér-Rao bound under

worst-case conditions. By applying signomial geometric programming, the proposed quantizers outperform existing methods, particularly in the mid-to-high signal-to-noise ratio regime.

- The thesis explores distributed estimation in graph-connected networks where nodes exchange information with their neighbors to reach consensus. A synchronous algorithm with stochastic activation is proposed, where nodes activate probabilistically to minimize data collisions while ensuring convergence. By optimizing the activation probability using theoretical bounds, the algorithm achieves a balance between convergence rate and communication efficiency.

In summary, these findings advance the understanding of information-theoretic limits and practical coding strategies, enhancing the performance of distributed systems across diverse applications.

Résumé (Français)

Les systèmes distribués constituent la colonne vertébrale des applications informatiques modernes, permettant l'exécution collaborative et efficace des tâches sur plusieurs composants interconnectés. Cependant, la séparation physique des composants du système introduit des défis majeurs en termes d'efficacité de communication. Cette thèse aborde ces défis en étudiant les limites théoriques de l'information et en proposant des solutions pratiques pour améliorer les performances de deux types de systèmes : les systèmes de calcul distribué (DC) et les systèmes d'estimation distribuée.

Les systèmes de calcul distribué exploitent la parallélisation des tâches pour réduire significativement les temps d'exécution des tâches intensives en calcul. Un cadre populaire pour ces systèmes, MapReduce, divise le calcul en trois phases : map, shuffle et reduce. La phase shuffle, qui implique le transfert de valeurs intermédiaires entre les nœuds, constitue souvent un goulot d'étranglement en termes de temps d'exécution. Bien que diverses méthodes de codage aient été proposées pour optimiser cette phase dans les réseaux câblés, l'adoption croissante des systèmes DC sans fil nécessite de nouvelles solutions adaptées aux environnements sans fil.

Les systèmes d'estimation distribuée, quant à eux, visent à estimer un paramètre commun de manière collaborative entre plusieurs nœuds. Ces systèmes sont largement utilisés dans les réseaux de capteurs, la surveillance environnementale et l'apprentissage machine distribué. L'efficacité de la communication est cruciale pour une fusion de données précise, car les pertes ou les retards de données peuvent dégrader les performances du système. Cette thèse se concentre sur deux scénarios : les systèmes avec un centre de fusion et ceux sans centre de fusion.

Les principales contributions de cette thèse sont les suivantes :

- Cette thèse étudie les réseaux sans fil partiellement connectés en introduisant de nouveaux schémas de codage optimisant le compromis calcul-communication pour les cadres MapReduce. En analysant les degrés de liberté globaux des canaux partiellement connectés, ce travail établit des bornes inférieures réalisables en utilisant l'alignement des interférences (IA). Une analyse théorique de l'information est également fournie pour ces réseaux.
- La thèse étend les techniques d'alignement des interférences pour optimiser les systèmes MapReduce sans fil fonctionnant sur des canaux d'interférences en duplex intégral. En concevant des schémas adaptés à des configurations spécifiques, les méthodes proposées réalisent des améliorations significatives des temps d'exécution par

rapport aux approches traditionnelles. Ce travail établit également des bornes théoriques supérieures et inférieures pour le compromis calcul-communication.

- Pour les systèmes d'estimation distribuée nécessitant un centre de fusion, cette thèse développe un cadre pour concevoir des quantificateurs multi-bits minimisant la borne de Cramér-Rao dans les conditions les plus défavorables. En appliquant la programmation géométrique signomiale, les quantificateurs proposés surpassent les méthodes existantes, en particulier dans les régimes de rapport signal-bruit moyen à élevé.
- La thèse explore les systèmes d'estimation distribuée dans des réseaux connectés par des graphes où les nœuds échangent des informations avec leurs voisins pour parvenir à un consensus. Un algorithme synchrone avec activation stochastique est proposé, où les nœuds s'activent de manière probabiliste pour minimiser les collisions de données tout en assurant la convergence. En optimisant la probabilité d'activation à l'aide de bornes théoriques, l'algorithme équilibre le taux de convergence et l'efficacité de la communication.

En résumé, ces résultats approfondissent la compréhension des limites théoriques de l'information et des stratégies de codage pratiques, améliorant les performances des systèmes distribués dans des applications variées.

摘要 (Abstract in Chinese)

分布式系统构成了现代计算应用的基础，其主要特征是支持多个网络组件之间的协作和高效任务执行。然而，系统组件间的物理分离在通信效率方面引入了显著的挑战。本论文通过研究信息论极限并提出实际可行的方案来应对这些挑战，并提升系统的性能。本论文主要关注两种分布式系统：分布式计算（DC）系统和分布式估计系统。

DC 系统将任务并行化以减少计算密集型任务的执行时间。MapReduce 是 DC 系统中的一种流行框架，其将计算分为三个阶段：map、shuffle 和 reduce。其中，shuffle 阶段涉及节点间中间值的传输，而这一阶段往往成为执行时间的瓶颈。尽管已有多种编码方案被提用于优化有线网络中的 shuffle 阶段，但随着无线 DC 系统的广泛采用，亟需针对无线环境设计新的解决方案。

另一方面，分布式估计系统旨在通过多个节点协作对同一参数进行估计。这类系统广泛应用于传感器网络、环境监测和分布式机器学习中。通信效率对准确的数据融合至关重要，因为数据损失可能导致系统性能下降。本论文主要研究两种场景：具有融合中心的系统和无融合中心的系统。

本论文的主要贡献如下：

- 本论文研究了部分连接的无线网络，提出了针对 MapReduce 框架的新型编码方案，优化了计算-通信权衡。具体来说，通过分析部分连接信道的总自由度，利用干扰对齐（IA）技术推导出可达的下界，并提供了部分连接无线网络的信息论分析。
- 本论文将 IA 技术扩展到全双工干扰信道下运行的无线 MapReduce 系统，通过为特定网络配置设计方案，使所提方法在执行时间上相较于传统方法实现了显著改进。此外，论文还为计算-通信权衡建立了理论上的上下界。
- 对于需要融合中心的分布式估计系统，本论文开发了一种框架，用于设计在最坏条件下最小化 Cramér-Rao 下界的多比特量化器。通过应用信号几何规划，所提量化器在中高信噪比范围内的性能优于现有方法。
- 本论文探索了图连接网络中的分布式估计，其中节点通过与其邻居交换信息以达成共识。提出了一种具有随机激活的同步算法，其中节点以概率方式激活，以减少数据碰撞并确保收敛。通过利用理论界优化激活概率，该算法在收敛速率与通信效率之间实现了平衡。

总之，本研究深化了对信息论极限和实际编码策略的理解，提升了分布式系统在多种应用场景下的性能。

General Introduction

1.1 Context and Objectives of the Thesis

Distributed systems are a category of computing systems in which components distributed across multiple networked computers communicate and coordinate to achieve a shared objective [1], [2]. These systems are fundamental to modern computing applications, including Cloud Computing, Blockchain, and the Internet of Things. Two key motivations for adopting distributed systems are their ability to enhance performance through parallel processing and their facilitation of node autonomy. However, as the name suggests, the components in a distributed system are physically separated, making communication efficiency a critical factor for their success. This thesis focuses on two types of distributed systems: *Distributed Computing (DC) systems* and *distributed estimation systems*, which respectively exploit the advantages of parallel processing and node autonomy.

DC Systems DC systems are computer networks that reduce execution times of complex computing tasks through task-parallelization, such as data mining or computer vision. MapReduce is a popular framework and runs in three phases [3], [4]. In the first *map phase*, nodes calculate Intermediate Values (IVA) from their associated input files. In the following *shuffle phase*, nodes exchange these IVAs in a way to inform each node about all IVAs required for computing its assigned output function during the final *reduce phase*. MapReduce is primarily applied to wired systems where it has been noticed that a significant part of the MapReduce execution time stems from the IVA *delivery time* during the shuffle phase [4], [5]. Various coding schemes [5]–[10] were proposed to reduce this IVA delivery time, and consequently speed up execution time compared to naive approaches. For example, [6] applied their Coded Distributed Computing (CDC) algorithm to sort 12GB of data by running TeraSort on 16 computing nodes of the Amazon EC2 clusters and 100 Mbps network speed. The traditional TeraSort algorithm spends 98% of the execution time (945s/961s) for the Shuffle Phase, and when implementing CDC with replication factor $r = 3$ this execution time can be reduced by a factor of 2.16 to 446s at the cost of multiplying the required storage space by the factor $r = 3$. If one is willing to multiply the available storage space by $r = 5$, then an even larger speed up by a factor of 3.39 to only 283s is possible.

In recent years, MapReduce systems became increasingly popular also for wireless scenarios, such as vehicular networks [11], in-flight entertainment [12] or augmented reality systems [13]s. Even in data centers, the wireless approach attracts a lot of attention since operators would like to reduce the large number of cables [14]. Thus, these developments create a need for coding schemes that perform well over wireless networks in the context of MapReduce systems.

Communications over wireless channels inherently interfere with each other, except that multiple access orthogonal schemes are employed, which however comes at the expense of a degraded performance. Building coding schemes for DC that are adapted to wireless channels and especially to the interference experienced in these channels is important for wireless DC system.

Distributed Estimation Systems In many applications, multiple nodes, e.g., machines, sensors, devices [15]–[17], are deployed across different locations but aim to measure the same phenomenon and estimate a common parameter. These nodes operate independently while collaboratively fusing their observations to reach a consensus. This approach enhances the system’s estimation accuracy and robustness. Such systems are commonly referred to as *distributed estimation systems*.

Efficient data delivery is a fundamental requirement for the success of distributed estimation systems. These systems rely on sharing and processing data across multiple nodes to make accurate estimations, decisions or predictions.

When data from various sensors or sources is passed efficiently, a more accurate "fusion" of this data can occur. This means that each node can make a more informed decision by leveraging the collective knowledge from other parts of the system. However, delays, loss of data, or inconsistent transmission can lead to poor fusion and, as a result, inaccurate estimations. For instance, in distributed sensor networks for environmental monitoring (such as detecting air pollution), different nodes may estimate air quality based on their local measurements. For an accurate global estimation, these data points need to be shared and passed efficiently. If data transfer is slow, some sensors may "time out" before transmitting their data, leading to a less reliable overall estimation. Another example can be found in distributed machine learning. Multiple devices or servers may participate in training a model. Efficient data passing ensures that gradients, weights, and model parameters are transmitted quickly between devices, which improves the accuracy of the model in less time.

Objective Consequently, **the objective of this thesis is to establish information-theoretic limits and develop practical coding schemes for distributed computing and estimation systems.** The methods employed to address such problems naturally depend on the system’s objectives and structure. This work concentrates on four distinct yet interconnected problems.

- Developing novel coding schemes and theoretical analysis for partially-connected channels, with applications to wireless distributed computing systems.

- Developing coding schemes and theoretical analysis for wireless distributed computing systems under fewer assumptions on file placement.
- Developing a multi-bit quantizer specialized for distributed estimation systems operating with a Fusion Center (FC).
- Developing a consensus algorithm for distributed estimation systems operating without a FC, where sensors are interconnected via a graph.

1.2 Related Works

Related Works on DC DC is primarily applied in systems with wired communication links, such as traditional data centers whose servers are connected through high-rate optical fibers. Coding schemes for speeding up communication in such wired systems have been proposed in [6]–[9], [18]–[21]. These schemes mainly rely on *xor* operations between packets.

Similarly to the wired case [6]–[8], delivery time in wireless MapReduce systems can be decreased by sending appropriate linear combinations of the IVAs, from which the receiving nodes can extract their desired IVAs by bootstrapping the IVAs that they can compute from their locally stored input files. Further improvements are however possible by exploiting the superposition nature of wireless networks, e.g., by cooperatively encoding messages, zero-forcing transmissions at specific sets of nodes, or aligning interference at nodes.

In this thesis, we focus on the high Signal-to-Noise Ratio (SNR) regime. There are two critical metrics for wireless MapReduce systems. The first metric is the computation load, which represents the average number of nodes to which each file is assigned. In other words, it is the ratio of the total number of stored input files (including replications) to the total number of original files. The second metric is the Normalized Delivery Time (NDT), which refers to the duration of the wireless shuffle phase normalized by the number of output functions, input files, and the transmission time of a IVA over a point-to-point channel in the high SNR regime. We aim to investigate the minimal NDT for a given computation load, a relationship we call the computation-NDT tradeoff.

Several works have analyzed computation-NDT tradeoffs for different wireless networks. For example, [22], [23] studied the computation-NDT tradeoff of cellular networks, and proposed schemes to reduce the NDTs by sending appropriate linear combinations of the IVAs and applying simple interference cancellation (bootstrapping of known IVAs) at the receiving nodes. (The energy-efficiency latency tradeoff in such cellular systems has been studied in [24].) Interference networks were studied in [25]–[27]. More specifically, [25] considered a half-duplex interference network and proposed a scheme that converted the network into a fully-connected X-channel, while it applied the IA-scheme in [28]. The computation-NDT tradeoff of full-duplex interference networks was considered in [26], [27]. The work [26] proposed a coding scheme based on one-shot beamforming and zero-forcing, and showed that this scheme was optimal for this class of strategies. The works in [25]–[27] all assumed perfect Channel State Information (CSI) at the transmitters. For scenarios with imperfect (delayed) channel-state information, [29] proposed a coding scheme that combines zero-forcing and interference cancellation with superposition coding.

In the existing literature [25], [27], it has been found that the structure of the wireless channel of the shuffle phase is similar to a well study wireless channel called X-channel. The study of the Sum Degrees of Freedom (SDoF) for X-channels, where each transmitter communicates with each receiver, both with and without cooperation, has a well-established history. It has been shown that the SDoF of a fully-connected K-user IC without cooperation is $K/2$ when the channel coefficients are independent and identically distributed (i.i.d.) fading according to a continuous distribution [30] and the SDoF of the corresponding X-channel is $K^2/(2K-1)$. Both these SDoFs were achieved with Interference Alignment (IA) [31]. Meanwhile, numerical approaches to obtain approximate SDoF were also proposed in [32]–[35]. Other research also considered imperfect Channel State Information at Transmitters (CSIT) [36]–[39]. The advantages of IA [30] for DC systems were first demonstrated in [27] within the context of full-duplex wireless networks. Specifically, [27] proposed partitioning nodes into groups and employing a combination of IA and zero-forcing techniques to ensure that signals intended for one node do not interfere with those received by other nodes within the same group. However, the structure of the full-duplex wireless network during the shuffle phase differs from the conventional IC or X-channel models often discussed in the existing literature, necessitating further research.

Recently, distributed computing systems designed for associative functions (also known as linearly separable functions) have garnered significant attention due to their wide-ranging applications in machine learning. Coding schemes for wired channel and broadcast channels were proposed in [40]–[42]. Research on other wireless channel are still needed.

Related Works on Distributed Estimation In many surveillance applications, sensors are distributed across different locations to measure a common phenomenon, often affected by noise, and estimate the same parameter [43]–[45]. Instead of estimating the parameter locally, sensors transmit their quantized data through a propagation channel to a FC, which performs the estimation. The design of effective multi-bit quantizers is a critical challenge in such systems.

Fisher information and the Cramér-Rao Bound (CRB) are foundational concepts in statistics and information theory, measuring the amount of information a random variable contains about an unknown parameter. Fisher information quantifies how much the random variable reveals about the parameter, while the CRB establishes a lower bound on the variance of any unbiased estimator of the parameter, derived directly from the Fisher information. Fisher information and the CRB for estimation system with a FC have been derived under various assumptions [46]–[53]. However, the results provided in [52], [53] independent of specific quantizers. On the other hand, other above-mentioned studies focused on specified quantizers without optimizing their design.

Meanwhile, other works addressed quantizer optimization under different configurations and assumptions. For example, [54] developed an optimal deterministic multi-bit quantizer for low SNR scenarios. Similarly, [55] employed Bayesian CRB and dynamic programming to optimize multi-bit quantizers for single-sensor setups. In [56], a deterministic quantizer was optimized for low SNR using particle swarm algorithms. Multi-sensor scenarios were considered in [57], where sensors used random one-bit quantizers with linear, piecewise, data-dependent thresholds to minimize the worst-case CRB. [58] proposed an iterative

method to optimize thresholds for minimizing the worst-case CRB in multi-sensor systems. In [59], the same setup as [57] was considered but they found the best threshold distribution without the linear-piecewise structure assumption. This setup was extended in [59], where optimal threshold distributions were identified without linear-piecewise constraints. These studies can be classified into two categories: those that assume a prior distribution on the unknown parameter [54]–[56], [58], and those that do not [57], [59]. Among the latter, most research focuses on one-bit quantizers, highlighting the need for further exploration of multi-bit quantizers.

For distributed estimation systems without a FC, each node computes an estimation based on its own data and the data received from its neighbors. These estimations are often updated iteratively. Various algorithms have been proposed for different scenarios. Some popular algorithms used in these systems include [45], [60]: consensus-based algorithms, Kalman Filters and Particle Filters. Consensus-based algorithms involve each node updating its estimation based on the information from neighboring nodes, aiming to achieve consensus across the network (i.e., agreement on a common estimation) [61]–[65]. In certain distributed systems, Kalman filtering is used to estimate the state of a system in real-time by considering both the local measurements and those shared by neighboring nodes [66]–[68]. Particle Filters are used when the system’s state is highly non-linear or non-Gaussian, and distributed versions of particle filters allow multiple agents to contribute their local estimations to refine the global estimation [69], [70].

The problem of achieving consensus on the average of initial sensor measurements is one of the most critical challenges in wireless distributed systems. Regarding node activation, there are two primary approaches for algorithm design: the synchronous approach and the asynchronous approach. In the synchronous approach, multiple sensors can be activated simultaneously to exchange local data. In contrast, the asynchronous approach limits activation to a single sensor or, at most, a pair of sensors at any given time.

Early research primarily focused on the asynchronous approach due to its simplicity and stability. For instance, one of the earliest asynchronous algorithms was introduced in [71] and later extended in [61]. This problem was addressed using the *Random Gossip* algorithm, where a randomly selected sensor communicates with one of its neighbors at each iteration. The two sensors exchanged their values and updated them by averaging the received and previous values. Given the broadcast nature of wireless channels, it is also promising to design algorithms that exploit this property. One example was the *Broadcast Gossip* algorithm [62], where an active node broadcasted its value to all neighbors, which then updated their estimations by averaging the received value with their previous estimations. Moreover, asynchronous algorithms have been studied extensively in various network settings, including directed graphs [72], [73], link failures [74], [75], and unstable sensors [76]. However, as network size increases, the asynchronous approach reveals its drawback of slow convergence. At the same time, advancements in synchronous transmission techniques have mitigated many of their limitations [77]–[79], leading to a renewed interest in the synchronous approach.

1.3 Main Contributions and Organization

This thesis is structured into six chapters, beginning with this chapter, the General Introduction. Chapters 2 and 3 focus on wireless DC systems, while Chapters 4 and 5 address wireless distributed estimation systems. Chapter 6 provides the conclusions and perspectives of the thesis. Additionally, an appendix is included to offer supplementary proofs for the theorems presented. The publications resulting from the research conducted during this thesis are summarized at the end. The specific contributions of Chapters 2 to 5 are detailed below.

In Chapter 2, we begin by considering a general class of channels referred to as *Partially-connected Channel (PC)*. In this type of channel, Transmitters (Tx) and Receivers (Rx) are grouped into sets of consecutive Tx and Rx, with each set containing more than one Tx or Rx. The connectivity between the Tx groups and Rx groups is represented by a connectivity matrix. The message flow between the groups is described by a message flow matrix. Each Rx observes a linear combination of signals from all connected transmitters, affected by Gaussian noise. The Tx within the same group cooperate to send a single message to the receivers in all intended groups, while the receivers decode their respective messages independently of each other. We examine three subclasses of the PC: *Non-cooperative Partially-connected Channel (NPC)*, *Cooperative Partially-connected X-Channel (CPXC)*, and *Cooperative Partially-connected Interference Channel (CPIC)*. First, we focus on NPC where each group contains only one Tx or one Rx. We provide a SDoF lower bound for NPC achieved by using IA for a network with a given connectivity matrix and message flow matrix. To establish this result, we design an IA scheme that assigns precoding matrices to the messages. Additionally, we formalize constraints on the assignment of precoding matrices, simplifying the construction process. CPXC is a specific case of PC where each Rx observes a linear combination of all Tx-signals in Gaussian noise, except for the signals sent by its corresponding Tx-group, and Tx in the same group cooperate to jointly transmit a message to each Rx in all other groups. For this network model, we enhance the scheme by exploiting the partial connectivity of the channel. By redesigning the precoding matrix allocation in accordance with the constraints we formalized, we reduce the number of distinct precoding matrices, thereby decreasing the dimensionality of the interference subspace. This improvement comes at the cost of transmitting slightly fewer messages. Additionally, we derive an information-theoretic upper bound on the SDoF of CPXC using model decomposition and a Multiple Access Channel (MAC) argument. CPIC is similar to the CPXC with the distinction that the k -th Tx group only sends a message to the $k+1$ -th Rx group (the last Tx group sends to the first Rx group). In [80], authors proved the CPIC shared the same SDoF lower bound of $K/2$ as the fully-connected interference channel. By applying a similar MAC argument, we obtain an information-theoretic upper bound that coincides with the lower bound.

We apply the SDoF bounds to improve the computation-NDT tradeoff which is defined as the achievable NDTs for a given computation load. We first consider MapReduce frameworks without making any assumption about the reduce functions, which we refer to as generic wireless DC. We show further improvement in the computation-NDT tradeoff using the new IA scheme for CPXC. We also consider a special case of MapReduce DC system

where the reduce function can be written as a series of associative binary operations. In this scenario, any subset of nodes can compute an intermediate reduce function using the data within that subset while preserving the final result. And only the results of intermediate reduce function need to be exchanged between the subsets, which leads to reduced communication costs. For the MapReduce setup, we propose a round-based scheme based on the IA scheme for CPIC, which achieves a computation-NDT tradeoff improved lower bound comparing to that of the time-sharing scheme.

In Chapter 3, we further improve the computation-NDT tradeoff of MapReduce over full-duplex wireless interference channels with two novel IA schemes. Our first scheme is inspired by the IA scheme in [28], where multi-cast messages are sent over a fully-connected interference network. We however adapt this scheme to our DC setup, where nodes simultaneously act as transmitters and receiver, allowing to achieve improved performance. We present a second IA-DC scheme for systems with an odd number of users K and computation load $r = \frac{K-1}{2}$, i.e., when each node can store almost half of the input files. In this second scheme each node only sees interference pertaining to one of the K utilized IA-precoding matrices, while all other non-intended transmissions at this node are zero-forced. In this sense, the presented IA scheme implies minimum interference space (because any non-trivial IA scheme has interference pertaining to at least one of the precoding matrices), allowing to obtain improved performance compared to other IA schemes. In fact, we also present an information-theoretic lower bound on the computation-NDT tradeoff based on a MAC type argument that is applied in parallel to a set of well-selected sub-systems and by solving a resulting linear program. For computation load $r < \frac{K}{2}$ the lower bound on the NDT is close to the proposed upper bound, but they do not match. For $r \geq \frac{K}{2}$ the lower bound matches the upper bound in [26] thus establishing the exact NDT of wireless MapReduce over full-duplex networks.

In Chapter 4, we consider multi-bit quantizers in multiple sensors scenarios for the min-max approach. We assume that each sensor has a random quantizer coming from a common distribution between sensors. And this common distribution is optimized when the number of sensors is large enough. In [59], the same approach was considered but for one-bit quantizer. Here, the challenge is to extend [59] to multi-bit quantizer. This extension is not straightforward for two reasons: *i*) expressing our CRB in closed-form requires order statistics, *ii*) the obtained optimization problem is not convex anymore. In the asymptotic regime respect to the number of sensors, we propose a framework based on Signomial Geometric Programming (SGP) to obtain optimized quantizer for the worst-case of the target parameter. Under the assumption that all thresholds of quantizers are generated according to a distribution, we express the CRB with order statistics, and the goal of the framework is to explore the optimal distribution for the CRB. After discretizing, we convert the optimization problem into a SGP which can be solved by the algorithm proposed in [81]. The obtained quantizer outperforms uniformly-distributed, regular deterministic quantizers and those proposed in [55], [56] in the mid-to-high SNR regime. Another interesting observation is that the quantizer performs well even with a limited number of sensors, despite being designed under the assumption of a large number of sensors.

In Chapter 5, we explore distributed estimation systems that operate without a FC,

and the sensors are connected by an undirected self-loop-free graph. Similar to the system discussed in the previous chapter, each node observes a noised version of the target parameter θ , and nodes exchange data with its neighbors in order to obtain a more precise estimation of θ and achieve a consensus. The problem of achieving consensus on the average of initial sensor measurements is one of the most critical challenges in wireless distributed systems. Regarding node activation, there are two primary approaches for algorithm design: the synchronous approach and the asynchronous approach. We focus on the synchronous approach, assuming nodes operate in full-duplex mode. In each time slot, a node can be either active, broadcasting its local data to neighbors, or inactive, remaining silent. Regardless of state, nodes always attempt to receive data. However, collisions may occur if too many neighboring nodes are active simultaneously, leading to data reception failure. The parameter α quantifies collision sensitivity, with higher values indicating a higher likelihood of failure. The goal is to design an algorithm that efficiently minimizes the Mean Square Error (MSE) of estimations and achieves consensus. To address this issue, we propose a synchronous scheme with stochastic activation, where each node can become active with a probability of $1 - \gamma$. We assume γ is uniform across nodes for simplicity, but refinements can be made in the future. By establishing a theoretical relationship between the convergence rate and γ , we leverage previous work on upper bounds for convergence rates [82], [83] to guide the design. Despite a gap between the actual and upper-bound convergence rates, their monotonic behavior enables the efficient determination of the optimal γ via one-dimensional search. Numerical results validate the optimized γ via the upper bound and show that the averaged optimized γ performs comparably to a graph-dependent γ , enabling practical use without precise graph knowledge.

1.4 Notations

This section provides the mathematical notations used in the thesis.

- We use **bold** for vectors and matrices, and calligraphic font for sets.
- The sets \mathbb{Z} , \mathbb{Z}^+ , \mathbb{R} and \mathbb{C} denote the sets of integers, positive integers, real numbers and complex numbers.
- For a finite set \mathcal{A} , $|\mathcal{A}|$ denotes its cardinality.
- For any $n \in \mathbb{Z}^+$, define $[n] \triangleq \{1, 2, \dots, n\}$.
- The set $[\mathcal{A}]^n$ denotes the collection of all subsets of \mathcal{A} with cardinality n , i.e., $[\mathcal{A}]^t \triangleq \{\mathcal{T} : \mathcal{T} \subset \mathcal{A}, |\mathcal{T}| = t\}$.
- $\|\mathbf{v}\|$ denote the Euclidean norm of vector \mathbf{v} , and v_i refers to the i -th element of vector \mathbf{v} .
- The transpose of \mathbf{v} is written as \mathbf{v}^T .
- For any vector \mathbf{v} , let $\text{diag}(\mathbf{v})$ be the diagonal matrix with diagonal entries given by the elements of the vector \mathbf{v} .

- For a matrix \mathbf{A} , \mathbf{A}^T denotes the transpose of \mathbf{A} , and $\text{vec}(\mathbf{A})$ denotes the vector of all elements of \mathbf{A} taken column-wise.
- When writing $[\mathbf{v}_i: i \in \mathcal{S}]$ or $[\mathbf{v}_i]_{i \in \mathcal{S}}$ we mean the matrix consisting of the set of columns $\{\mathbf{v}_i\}_{i \in \mathcal{S}}$.
- We use \mathbf{Id}_K to represent a $K \times K$ identity matrix.
- We use $\mathbf{0}$ and $\mathbf{1}$ to represent zero and one matrices/vectors of appropriate dimensions.
- $\mathbf{A} \oplus \mathbf{B}$ denotes the direct sum of matrices \mathbf{A} and \mathbf{B} , and $\mathbf{A} \otimes \mathbf{B}$ denotes their Kronecker product.
- For any function f , we denote its convex lower envelope by $\text{lowc}(f(\ell))$.
- The limit superior and inferior are denoted by $\overline{\lim}$ and $\underline{\lim}$, respectively, with the notation $\overline{\lim}_{l \rightarrow p} f(l)$ and $\underline{\lim}_{l \rightarrow p} f(l)$.

Degrees of Freedom of Channels with Applications to Wireless Distributed Computing (DC)

2.1 Introduction

The exact capacity region of a multi-user channel with interference is generally unknown. One way to provide insights on the capacity region is to resort to the *SDoF* of the channel, which characterizes the pre-log approximation of the sum-capacity in the asymptotic regime of infinite SNR [84], i.e., when the network operates in the interference-limited regime. The study of the SDoF of Interference Channel (IC) and X-channels (where each Tx sends a message to each Rx) with and without cooperation has a rich history, see e.g., [30], [31], [33], [85]–[92]. In particular, it has been shown that the SDoF of a fully-connected K -user IC without cooperation is $K/2$ when the channel coefficients are independent and identically distributed (i.i.d.) fading according to a continuous distribution [30] and the SDoF of the corresponding X-channel is $K^2/(2K - 1)$. Both these SDoFs are achieved with interference alignment (IA) [31]. The essential part of the classic IA scheme presented in [31] is the design of precoding matrices. For ICs, all codewords are assigned the same precoding matrix. For X-channels, codewords intending to the same Rx group are assigned the same precoding matrices. The precoding matrices are obtained with a specific method such that all interfering codewords premultiplied by the same precoding matrix are aligned into the same subspace while keeping useful codewords decodable. Although the total number of communication channels stays constant, a larger proportion of these channels are now available to transmit valuable messages to each user.

We consider a more general channel class called *Partially-connected Channel (PC)*. In this type channel, TxS/Rxs are gathered into groups of $r > 0$ consecutive TxS/Rxs, and the number of group is denoted as \tilde{K} . The connectivity between Tx groups and Rx groups is characterized by a matrix $\mathbf{N} \in \{0, 1\}^{(\tilde{K} \times \tilde{K})}$, where the entry in row- k and column j equals 1 if the signal sent by the k -th Tx group interferes at all Rxs in the j -th group. The message flow is described by a matrix $\mathbf{M} \in \{0, 1\}^{(\tilde{K} \times \tilde{K})}$, where the entry in row- k and column j equals 1 if k -th Tx group cooperatively sends independent messages to each Rx in group j . Each Rx observes a linear combination of all connected Tx-signals in Gaussian noise. TxS in the same group cooperate to jointly transmit a message to each Rx in all

intended groups, while RxS decode their intended messages independently of each other. For example, the following matrices represent the PC channel in Fig. 2.1.

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (2.1)$$

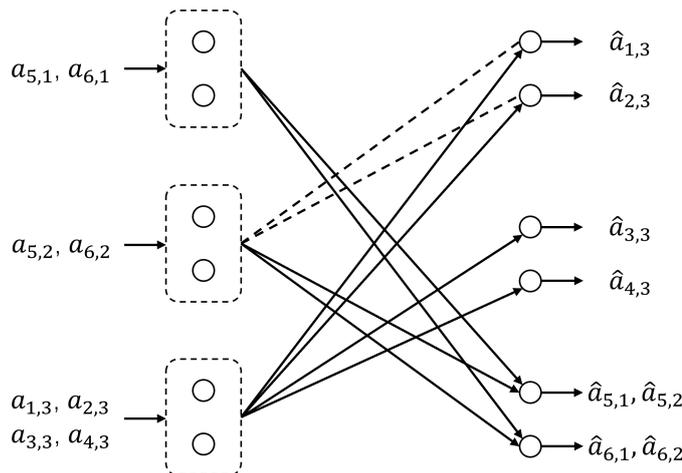
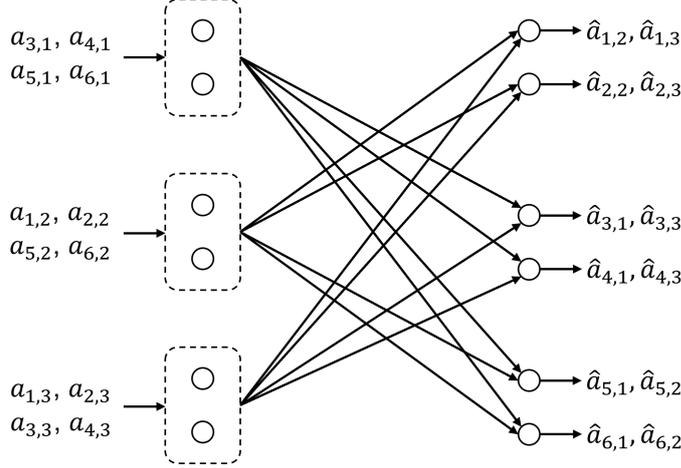


Figure 2.1: An example of PC model for $K = 6$, $r = 2$ and the matrices \mathbf{M} , \mathbf{N} in (2.1). The solid arrow represent the message passing, while the dash line indicates that the Tx group cause interference to the Rx.

In this chapter, we explore three sub-classes of PC.

Non-cooperative Partially-connected Channel (NPC) We first examine the case that $r = 1$. Since there is no cooperation between TxS, we classify this type of channels as NPC. In Theorem 2.1 of Section 2.3, we demonstrate a lower bound for NPC that is achievable through IA for the interference network with connectivity matrix \mathbf{N} and message flow matrix \mathbf{M} . To prove the theorem, we construct an IA scheme that assigns the precoding matrices to messages. We imply a matrix $\mathbf{G} \in (\mathbb{Z}^+)^{K \times K}$ to represent the allocation of precoding matrices. Each entry of \mathbf{G} indicates the index of the precoding matrix applied to the corresponding message. Designing a new IA scheme is a challenging task. Therefore, we establish two rules for \mathbf{G} that ensure the validity of the IA scheme, and enable a straightforward calculation of the achieved SDoF using \mathbf{G} . This approach provides a more intuitive method for designing new IA scheme for any given matrices \mathbf{M} and \mathbf{N} .

Cooperative Partially-connected X-Channel (CPXC) We study a specific case of PC where each Rx observes a linear combination of all Tx-signals in Gaussian noise, except for the signals sent by its corresponding Tx-group, and TxS in the same group cooperate to jointly transmit a message to each Rx in all other groups. That is to say $\mathbf{N} = \mathbf{M} = \mathbf{1} \cdot \mathbf{1}^T - \mathbf{Id}_K$. An example with $K = 6$ is shown in Fig. 2.2. For this network model, comparing to the original IA scheme in [31], we further improve the scheme by

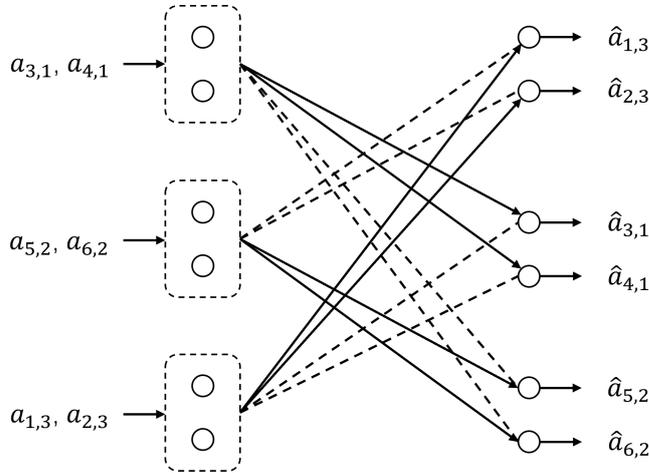

 Figure 2.2: CPXC model for $K = 6$, $r = 2$

exploiting the partially connectivity of the channel. By redesigning the precoding matrix allocation method, we can decrease the number of different precoding matrices, which reduces the dimension of the interference subspace, at the expense of sending slightly fewer messages. The new IA scheme achieves SDoF $(K(K-r) - r)/(2K - 3r)$ whenever the ratio of K by r is an integer larger than 3, and achieves SDoF $2r$ when $K/r \in \{2, 3\}$.

In [31, Theorem 2], authors investigated CPXC for the special case $r = 1$. With the original IA scheme, an lower bound $K/2$ is obtained. Our lower bound improves over this result when $K \geq 3$.

We further show a information-theoretic upper bound on SDoF of $K(K-r)/(2K-3r)$. To prove the upper bound, we select a series of subsystems, and each subsystem is equivalent to a multi-access channel (MAC). The upper bound is obtained by linearly combining the series of subsystems. For $K/r = 2$, the upper bound and lower bound are matched. For $K/r \geq 3$, there is a gap of $r/(2K - 3r)$ between the two bounds.

Cooperative Partially-connected Interference Channel (CPIC) We also study the SDoF of the Cooperative Partially-connected Interference Channel (CPIC). An example with $K = 6$ is shown in Fig. 2.3. This channel is similar with the CPXC with the distinction


 Figure 2.3: The cooperative interference channel model for $K = 6$, $r = 2$.

that the k -th Tx group only sends a message to the $k + 1$ -th Rx group (the last Tx group sends to the first Rx group). In [80], authors proved the CPIC shares the same SDoF lower bound of $K/2$ as the fully-connected interference channel. By applying a similar MAC argument, we obtain an information-theoretic upper bound that coincides with the lower bound.

Mapreduce framework MapReduce is a popular framework to carry out heavy computation tasks [3], [4]. The key feature of MapReduce is decomposing the original function as map functions and reduce functions. The framework runs in three phases. In the first *map phase*, nodes calculate intermediate values (IVA) from their associated input files and map functions. In the subsequent *shuffle phase*, nodes exchange these IVAs to obtain all IVAs required to run the final *reduce phase* where they compute the desired output with reduce function. The largest part of the execution time in MapReduce systems stems from the IVA *delivery time* during the shuffle phase. Several works proposed to reduce this delivery time through smart coding. More specifically, in wired networks, delivery time is decreased by sending appropriate linear combinations of the IVAs [6]–[8]. Over wireless distributed system, we focus on two important metrics of the system:

- *Computation load r* : This metric is defined identically as the one in wired DC system, representing the average number of nodes each file is assigned to. In other words, it is the ratio of the total number of stored input files (including replicas) to the total number of distinct files.¹
- *Normalized Delivery Time (NDT) Δ* : This is the duration of wireless shuffle normalized by the total number of IVAs, and by the transmission time of one IVA over a point-to-point channel in the high SNR regime.

The goal of coding schemes is to improve the *computation-NDT tradeoff* which is defined as the achievable NDT for a given r . We first consider MapReduce frameworks without making any assumption about the reduce functions, which we refer to as generic wireless DC. Over cellular networks [22], [23], similar to the wired network, an improvement is achieved through simple interference cancellation at the receiving nodes. And over wireless full-duplex networks a gain was achieved by zero-forcing [26]. In this chapter, we show further improvement in the computation-NDT tradeoff using the new IA scheme for CPXC.

We also consider a special case of MapReduce DC system where the reduce function can be written as a series of associative binary operations. In this scenario, any subset of nodes can compute an intermediate reduce function using the data within that subset while preserving the final result. And only the results of intermediate reduce function need to be exchanged between the subsets, which leads to reduced communication costs. This type of associative functions are commonly used in distributed system, such as matrix multiplication [40] and gradient descent [93], and are considered in [40] for wired channels

¹We will demonstrate later that the communication channel of MapReduce with Computation load r can be converted into a PC with the number Tx/Rx in each group equals r . Therefore, we use the same notation r for Computation load.

and in [41], [42] for broadcast channels. For the MapReduce setup, we propose a round-based scheme based on the IA scheme for CPIC, which achieves a computation-NDT tradeoff improved lower bounded comparing to that of the TDMA scheme.

Contributions and organization To summarize, the main contribution of this chapter are:

- For NPC, A new lower bound of the SDoF is obtained (Theorem 2.1).
- For CPXC, a new IA scheme is presented that improve the SDoF of CPXC, and an information-theoretic upper bound for the SDoF of CPXC (Theorem 2.2).
- For CPIC, the exact SDoF of the channel is derived. (Theorem 2.3).
- For the generic wireless MapReduce system, an improved computation-NDT tradeoff is obtained based on the new IA scheme for CPXC (Theorem 2.4).
- For wireless MapReduce system with associative reduce function, an improved computation-NDT tradeoff is obtained based on the new IA scheme for CPIC (Theorem 2.5).

The rest of this chapter is organized as follows. In Section 2.2, we present in detail the three channels: NPC, CPXC and CPIC. In Section 2.3, we present our main results on the SDoF of these channels. The application of the CPXC to generic wireless DC is explained in Section 2.4. The application of CPIC to wireless DC system with associative reduce function is explained in Section 2.5. Proofs for the SDoF lower and the upper bounds are given in Sections 2.6. Concluding remarks are drawn in Section 2.7.

2.2 System Models of Partially Connected Channels

Consider an interference network with K TxS and K RxS labeled from 1 to K . TxS and RxS are divided into groups. For a given group-size $r \geq 1$, where K is assumed divisible by r , we define the group of TxS/RxS

$$\mathcal{T}_k \triangleq \{(k-1)r+1, \dots, kr\}, \quad k \in [\tilde{K}], \quad (2.2)$$

where $\tilde{K} \triangleq K/r$.

In our network model, inter-group connectivity is represented by the binary matrix $\mathbf{N} \in \{0, 1\}^{\tilde{K} \times \tilde{K}}$. All TxS within the same group collaboratively transmit messages over a complex channel, while messages from different Tx groups remain independent. Each Rx p in Rx-group \mathcal{T}_j receives a linear combination of signals transmitted by all TxS in Tx-group \mathcal{T}_k whenever $\mathbf{N}[j, k] = 1$, with the received signals further corrupted by Gaussian noise. Denoting Tx q 's slot- t input by $X_q(t) \in \mathbb{C}$ and Rx p 's slot- t output by $Y_p(t) \in \mathbb{C}$, the input-output relation of the network is:

$$Y_p(t) = \sum_{q: q \in \mathcal{T}_k, \mathbf{N}[j, k]=1} H_{p,q}(t)X_q(t) + Z_p(t), \quad p \in \mathcal{T}_j, \quad (2.3)$$

where the sequences of complex-valued channel coefficients $\{H_{p,q}(t)\}$ and standard circularly symmetric Gaussian noises $\{Z_p(t)\}$ are both i.i.d. and independent of each other and

of all other channel coefficients and noises. The real and imaginary parts of a coefficient $H_{p,q}(t)$ are i.i.d. according to a given continuous distribution on some bounded interval $[-\mathbf{H}_{\max}, \mathbf{H}_{\max}]$ and are known by all terminals even before communication starts.

Therefore, a partially connected channel can be represented by a tuple with parameters $(\mathbf{K}, r, \mathbf{M}, \mathbf{N})$. In this chapter, we focus on three specific partially connected channels.

2.2.1 Non-cooperative Partially Connected Channel (NPC)

We first consider the channel model where there is only one Tx/Rx in each group, i.e. each NPC is represented by the tuple $(\mathbf{K}, 1, \mathbf{M}, \mathbf{N})$. As Txs in different groups transmit independent messages, we call this network a Non-cooperative Partially Connected Channel (NPC). In this channel, the input-output relation in (2.3) is rewritten as:

$$Y_p(t) = \sum_{\{q: \mathbf{M}[p,q]=1\}} H_{p,q}(t)X_q(t) + Z_p(t), \quad p \in [\mathbf{K}], \quad (2.4)$$

The message flow in this channel is represented by the matrix \mathbf{M} , in the sense that Tx q transmits an independent message $a_{p,q}$ to each Rx p for which $\mathbf{M}[p, q] = 1$. When communication is of blocklength T , each message is uniformly distributed over $[2^{\mathsf{T}R_{p,q}}]$, where $R_{p,q} \geq 0$ denotes the rate of transmission, and it is independent of all other messages and of all channel coefficients and noise sequences. As a consequence, Tx $q \in [\mathbf{K}]$ produces its block of channel inputs $\mathbf{X}_q \triangleq (X_q(1), \dots, X_q(\mathsf{T}))$ as

$$\mathbf{X}_q = f_q^{(\mathsf{T})}(\{a_{p,q}: \mathbf{M}[p, q] = 1\}) \quad (2.5)$$

by means of an encoding function $f_q^{(\mathsf{T})}$ on appropriate domains and so that the inputs satisfy the block-power constraint

$$\frac{1}{\mathsf{T}} \sum_{t=1}^{\mathsf{T}} \mathbb{E}[|X_q(t)|^2] \leq \mathsf{P}, \quad q \in [\mathbf{K}]. \quad (2.6)$$

Note that the encoding function is deterministic, and the expectation is taken over all possible message realizations.

Given a power $\mathsf{P} > 0$, the *capacity region* $\mathcal{C}(\mathsf{P})$ is defined as the set of all rate tuples $(R_{p,q}: p, q \in [\mathbf{K}], \mathbf{M}[p, q] = 1)$ so that for each blocklength T there exist encoding functions $\{f_q^{(\mathsf{T})}\}_{q \in [\mathbf{K}]}$ as described above and decoding functions $\{g_{p,q}^{(\mathsf{T})}\}$ on appropriate domains producing the estimates

$$\hat{a}_{p,q} = g_{p,q}^{(\mathsf{T})}(Y_p(1), \dots, Y_p(\mathsf{T})), \quad p, q \in [\mathbf{K}], \mathbf{M}[p, q] = 1, \quad (2.7)$$

in a way that the sequence of error probabilities

$$p^{(\mathsf{T})}(\text{error}) \triangleq \Pr \left[\bigcup_{p \in [\mathbf{K}]} \bigcup_{\{q: \mathbf{M}[p,q]=1\}} \hat{a}_{p,q} \neq a_{p,q} \right] \quad (2.8)$$

tends to 0 as the blocklength $\mathsf{T} \rightarrow \infty$.

Our main interest is in the *Degrees of Freedom (DoF)* of the channel:

$$\text{SDoF}_{\text{NPC}} \triangleq \overline{\lim}_{\mathsf{P} \rightarrow \infty} \sup_{\mathbf{R} \in \mathcal{C}(\mathsf{P})} \sum_{(p,q): \mathbf{M}[p,q]=1} \frac{R_{p,q}}{\log \mathsf{P}}. \quad (2.9)$$

2.2.2 Cooperative Partially Connected X-Channel (CPXC)

Next, we consider CPXC, a channel characterized by the tuple $(\mathbf{K}, r, \mathbf{N}, \mathbf{N})$, where

$$\mathbf{N} = \mathbf{1} \cdot \mathbf{1}^T - \mathbf{Id}_K.$$

In other word, each Rx p in Rx-group \mathcal{T}_j observes a linear combination of the signals sent by all Tx's *outside* Tx-group \mathcal{T}_j . All Tx's in Tx-group \mathcal{T}_k cooperatively transmit an individual message $a_{p,k}$ to each Rx $p \in [\mathbf{K}] \setminus \mathcal{T}_k$ *outside* Rx-group k . When communication is of blocklength T , this message is uniformly distributed over $[2^{\mathsf{T}R_{p,k}}]$, where $R_{p,k} \geq 0$ denotes the rate of transmission, and it is independent of all other messages and of all channel coefficients and noise sequences. As a consequence, Tx $q \in [\mathbf{K}]$ produces its block of channel inputs $X_q^{(\mathsf{T})} \triangleq (X_q(1), \dots, X_q(\mathsf{T}))$ as

$$X_q^{(\mathsf{T})} = f_q^{(\mathsf{T})} \left(\left\{ a_{p,k} : k = \left\lceil \frac{q}{r} \right\rceil, p \in [\mathbf{K}] \setminus \mathcal{T}_k \right\} \right) \quad (2.10)$$

by means of an encoding function $f_q^{(\mathsf{T})}$ on appropriate domains and so that the inputs satisfy the block-power constraint

$$\frac{1}{\mathsf{T}} \sum_{t=1}^{\mathsf{T}} \mathbb{E} [|X_q(t)|^2] \leq \mathsf{P}, \quad q \in [\mathbf{K}]. \quad (2.11)$$

Given a power $\mathsf{P} > 0$, the *capacity region* $\mathcal{C}(\mathsf{P})$ is defined as the set of all rate tuples $(R_{p,k} : k \in [\tilde{\mathbf{K}}], p \in [\mathbf{K}] \setminus \mathcal{T}_k)$ so that for each blocklength T there exist encoding functions $\{f_q^{(\mathsf{T})}\}_{q \in [\mathbf{K}]}$ as described above and decoding functions $\{g_{p,k}^{(\mathsf{T})}\}$ on appropriate domains producing the estimates

$$\hat{a}_{p,k} = g_{p,k}^{(\mathsf{T})}(Y_p(1), \dots, Y_p(\mathsf{T})), \quad k \in [\tilde{\mathbf{K}}], p \in [\mathbf{K}] \setminus \mathcal{T}_k, \quad (2.12)$$

in a way that the sequence of error probabilities

$$p^{(\mathsf{T})}(\text{error}) \triangleq \Pr \left[\bigcup_{k \in [\tilde{\mathbf{K}}]} \bigcup_{p \in [\mathbf{K}] \setminus \mathcal{T}_k} \hat{a}_{p,k} \neq a_{p,k} \right] \quad (2.13)$$

tends to 0 as the blocklength $\mathsf{T} \rightarrow \infty$.

Our main interest is in the SDoF, which characterizes the logarithmic growth in the high power regime of the maximum sum of all rates tuples inside the capacity region:

$$\text{SDoF}_{\text{CPXC}} \triangleq \overline{\lim}_{\mathsf{P} \rightarrow \infty} \sup_{\mathbf{R} \in \mathcal{C}(\mathsf{P})} \sum_{k \in [\tilde{\mathbf{K}}]} \sum_{p \in [\mathbf{K}] \setminus \mathcal{T}_k} \frac{R_{p,k}}{\log \mathsf{P}}. \quad (2.14)$$

2.2.3 Cooperative Partially Connected Interference Channel (CPIC)

We consider CPIC, a channel with the tuple $(\mathbf{K}, r, \mathbf{M}, \mathbf{N})$, where

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad \mathbf{N} = \mathbf{1} \cdot \mathbf{1}^T - \mathbf{Id}_K.$$

In this channel, all Tx's in Tx-group \mathcal{T}_k cooperatively multicast a message a_k to every Rx $p \in \mathcal{T}_{k+1}$ in Rx-group $k + 1$ (a_K is sent to Rx $p \in \mathcal{T}_1$). This message is uniformly distributed over $[2^{\mathbf{T}R_k}]$, where $R_k \geq 0$ denotes the rate of transmission and \mathbf{T} denotes the blocklength, and it is independent of all other messages and of all channel coefficients and noise sequences. Similar to the cooperative X-channel, the channel inputs of Tx $q \in [K]$ is produced as

$$X_q^{(\mathbf{T})} = f_q^{(\mathbf{T})} \left(\left\{ a_k : k = \left\lceil \frac{q}{r} \right\rceil \right\} \right). \quad (2.15)$$

The block-power constraint and capacity region are defined similarly to that of cooperative X-channel, and the decoding functions $\{g_k^{\mathbf{T}}\}$ produces the estimates

$$\hat{a}_k = g_k^{(\mathbf{T})}(Y_p(1), \dots, Y_p(\mathbf{T})), \quad k \in [\tilde{K} - 1], p \in \mathcal{T}_{k+1}, \\ k = K, p \in \mathcal{T}_1 \quad (2.16)$$

The encoding functions and decoding functions are well chosen so that the error probabilities tends to 0 when the blocklength \mathbf{T} tends to infinity. The SDoF of cooperative interference channel is defined as

$$\text{SDoF}_{\text{CPIC}} \triangleq \overline{\lim}_{\mathbf{P} \rightarrow \infty} \sup_{\mathbf{R} \in \mathcal{C}(\mathbf{P})} \sum_{k \in [K]} \frac{R_k}{\log \mathbf{P}}. \quad (2.17)$$

2.3 Main Results

The main results of this section are lower bounds on the SDoF_{NPC} of NPC described in the previous Section 2.2.1, new upper and lower bounds on the SDoF of the cooperative X-channel described in the previous Section 2.2.2, and the upper bound on the SDoF of the cooperative interference channel described in Section 2.2.3. We restrict attention to $K/r > 1$ for CPXC and CPIC, because for $r = K$ the Rx's only observe noise and trivially $\text{SDoF} = 0$ in this case.

To comprehend the results on NPC, we first present the definition of a valid precoding index matrix.

Definition 2.1 For a given message matrix $\mathbf{M} \in \{0, 1\}^{K \times K}$, a matrix $\mathbf{G} \in \mathbb{N}^{K \times K}$ is called a **valid precoding index matrix** if the following two requirements are satisfied:

- If $\mathbf{M}[p, q] = 0$ then $\mathbf{G}[p, q] = 0$;
- If $\mathbf{G}[p, q] = \mathbf{G}[p', q]$ for $p \neq p'$, then $\mathbf{G}[p, q] = 0$

Also denote by $\mathbf{G}^{(p)}$ the submatrix of \mathbf{G} obtained by removing the p -th row and the columns $q \in [K]$ for which $\mathbf{N}[p, q] = 0$. More formally:

$$\mathbf{G}^{(p)} \triangleq \mathbf{G}[p' \neq p, \{q : \mathbf{N}[p, q] = 1\}]. \quad (2.18)$$

(Notice the dependence of $\mathbf{G}^{(p)}$ on the connectivity matrix \mathbf{N} , which is not represented in the notation.) Let $g^{(p)}$ be equal to the number of different non-zero integers in the matrix $\mathbf{G}^{(p)}$.

Theorem 2.1 (Lower bound for NPC) For an NPC, the SDoF_{NPC} defined in Section 2.2.1 is lower bounded by:

$$\text{SDoF}_{\text{NPC}} \geq \text{SDoF}_{\text{Lb, NPC}} \triangleq \max_{\mathbf{G} \in \mathcal{G}_{\mathbf{M}}} \frac{\|\mathbf{G}\|_0}{\max_{p \in [K]} \{\|\mathbf{G}[p, :]\|_0 + g(p)\}}, \quad (2.19)$$

where $\mathcal{G}_{\mathbf{M}}$ is the set of all valid precoding index matrices for the given \mathbf{M} .

Proof: See Section 2.6.1. ■

Theorem 2.2 (Lower and upper bounds for CPXC) When K/r is an integer larger than 1, the $\text{SDoF}_{\text{CPXC}}$ defined in Section 2.2.2 is lower bounded as:

$$\text{SDoF}_{\text{CPXC}} \geq \text{SDoF}_{\text{Lb, CPXC}} \triangleq \begin{cases} 2r & \text{if } K/r \in \{2, 3\}, \\ \frac{K(K-r)-r^2}{2K-3r} & \text{if } K/r \geq 4, \end{cases} \quad (2.20)$$

and upper bounded as:

$$\text{SDoF}_{\text{CPXC}} \leq \text{SDoF}_{\text{Ub, CPXC}} \triangleq \frac{K(K-r)}{2K-3r}. \quad (2.21)$$

Proof: See Section 2.6.2 for the proof of the lower bound and Section 2.6.3 for the proof of the upper bound. ■

For $K/r \geq 4$, the additive gap between the lower and upper bounds in (2.20) and (2.21) is $\frac{r}{2K-3r}$. This gap is decreasing in K and in the ratio K/r , and it is increasing in r .

For $K/r \in \{2, 3\}$ the bounds (2.20) and (2.21) match and yield:

Corollary 2.1 For $K/r \in \{2, 3\}$, we have $\text{SDoF}_{\text{CPXC}} = 2r$.

Proof: When $K/r = 2$, we have

$$\text{SDoF}_{\text{Ub, CPXC}} = \frac{2r(2r-r)}{4r-3r} = 2r.$$

When $K/r = 3$, we have

$$\text{SDoF}_{\text{Ub, CPXC}} = \frac{3r(3r-r)}{6r-3r} = 2r. \quad \blacksquare$$

For $r = 1$, our lower bound (2.20) improves over the lower bound $\text{SDoF} \geq K/2$ reported in [31] for all values of K .

Theorem 2.3 (Lower and upper bounds for CPIC) When K/r is an integer larger than 1, the $\text{SDoF}_{\text{CPIC}}$ defined in Section 2.2.3 is lower bounded as:

$$\text{SDoF}_{\text{CPIC}} \geq \text{SDoF}_{\text{Lb, CPIC}} \triangleq \frac{K}{2} \quad (2.22)$$

and is upper bounded as

$$\text{SDoF}_{\text{CPIC}} \leq \text{SDoF}_{\text{Ub, CPIC}} \triangleq \frac{K}{2} \quad (2.23)$$

Proof: The proof of the lower bound is provided in [80] and see Section 2.6.3 for the proof of the upper bound. ■

Table 2.1: Table of bounds for three channels.

Channel	Lower bound	Upper bound
NPC	$\max_{\mathbf{G} \in \mathcal{G}_M} \frac{\ \mathbf{G}\ _0}{\max_{p \in [K]} \{\ \mathbf{G}[p, :]\ _0 + g^{(p)}\}}$	—
CPXC ($K/r \in \{2, 3\}$)	$2r$	$2r$
CPXC ($K/r \geq 4$)	$\frac{K(K-r)-r}{2K-3r}$	$\frac{K(K-r)}{2K-3r}$
CPIC	$\frac{K}{2}$	$\frac{K}{2}$

Corollary 2.2 For all K/r is an integer larger than 1, we have $\text{SDoF}_{\text{CPIC}} = \frac{K}{2}$.

In Table 2.1, we summarise the obtained bounds for the above mentioned channel.

Remark 2.1 By the symmetry of the setup and standard time-sharing arguments, the bound in (2.20) implies the following bound on the Per-Message DoF ($\text{PMDoF}_{\text{CPXC}}$)

$$\text{PMDoF}_X \triangleq \lim_{P \rightarrow \infty} \sup_{\mathbf{R} \in \mathcal{C}(P)} \min_{\substack{k \in [K] \\ p \in [K] \setminus \mathcal{T}_k}} \frac{R_{p,k}}{\log P} \geq \frac{\text{SDoF}_{\text{Lb, CPXC}}}{K(K/r - 1)}. \quad (2.24)$$

And the bound in (2.22) implies the following bound on

$$\text{PMDoF}_{\text{CPIC}} \triangleq \lim_{P \rightarrow \infty} \sup_{\mathbf{R} \in \mathcal{C}(P)} \min_{\substack{k \in [K] \\ p \in [K] \setminus \mathcal{T}_k}} \frac{R_{p,k}}{\log P} \geq \frac{\text{SDoF}_{\text{Lb, CPIC}}}{K}. \quad (2.25)$$

2.4 Application to Generic Wireless DC

2.4.1 The MapReduce System

Consider a DC system with K nodes labelled $1, \dots, K$; N input files W_1, \dots, W_N ; and Q output functions h_1, \dots, h_Q mapping the input files to the desired computations. A *MapReduce* System decomposes the functions h_1, \dots, h_Q as

$$h_q(W_1, \dots, W_N) = \phi_q(a_{q,1}, \dots, a_{q,N}), \quad q \in [Q], \quad (2.26)$$

where ϕ_q is an appropriate *reduce function* and $a_{q,i}$ is an *IVA* calculated from input file W_i through an appropriate *map function*:

$$a_{q,i} = \psi_{q,i}(W_i), \quad i \in [N]. \quad (2.27)$$

For simplicity, all IVAs are assumed independent and consisting of A i.i.d. bits.

Computations are performed in 3 phases:

Map phase: A subset of all input files $\mathcal{M}_p \subseteq [N]$ is assigned to each node $p \in [K]$. Node p computes all IVAs $\{a_{q,i} : i \in \mathcal{M}_p, q \in [Q]\}$ associated with these input files.

Shuffle phase: Computations of the Q output functions is assigned to the K nodes, where we denote by $\mathcal{Q}_p \subseteq [Q]$ the output functions assigned to Node p .

The K nodes in the system communicate over T uses of a wireless network in a full-duplex mode, where T is a design parameter and it depends on K, A , the encoding method

and so on. During this communication, nodes communicate IVAs that they calculated in the Map phase to nodes that are missing these IVAs for the computations of their assigned output functions. So, Node $p \in [\mathsf{K}]$ produces complex channel inputs of the form

$$X_p^{(\mathsf{T})} \triangleq (X_p(1), \dots, X_p(\mathsf{T})) = f_p^{(\mathsf{T})}(\{a_{1,i}, \dots, a_{\mathsf{Q},i}\}_{i \in \mathcal{M}_p}), \quad (2.28)$$

by means of appropriate encoding function $f_p^{(\mathsf{T})}$ satisfying the power constraint (2.6). Given the full-duplex nature of the network, Node p also observes the complex channel outputs

$$Y_p(t) = \sum_{\ell \in [\mathsf{K}]} H_{p,\ell}(t) X_\ell(t) + Z_p(t), \quad t \in [\mathsf{T}], \quad (2.29)$$

where noises $\{Z_p(t)\}$ and channel coefficients $\{H_{p,\ell}(t)\}$ are as defined in Section 2.2.

Based on its outputs $Y_p^{(\mathsf{T})} \triangleq (Y_p(1), \dots, Y_p(\mathsf{T}))$ and the IVAs $\{a_{q,i}: i \in \mathcal{M}_p, q \in [\mathsf{Q}]\}$ it computed during the Map phase, Node p decodes the missing IVAs $\{a_{q,i}: i \notin \mathcal{M}_p, q \in \mathcal{Q}_p\}$ required to compute its assigned output functions $\{h_q\}_{q \in \mathcal{Q}_p}$ as:

$$\hat{a}_{q,i} = g_{q,i}^{(\mathsf{T})}(\{a_{1,i}, \dots, a_{\mathsf{Q},i}\}_{i \in \mathcal{M}_p}, Y_p^{(\mathsf{T})}), \quad i \notin \mathcal{M}_p, q \in \mathcal{Q}_p. \quad (2.30)$$

Reduce phase: Each node applies the reduce functions to the appropriate IVAs calculated during the Map phase or decoded in the Shuffle phase.

The performance of the DC system is measured in terms of its *computation load*

$$r \triangleq \sum_{p \in [\mathsf{K}]} \frac{|\mathcal{M}_p|}{\mathsf{N}}, \quad (2.31)$$

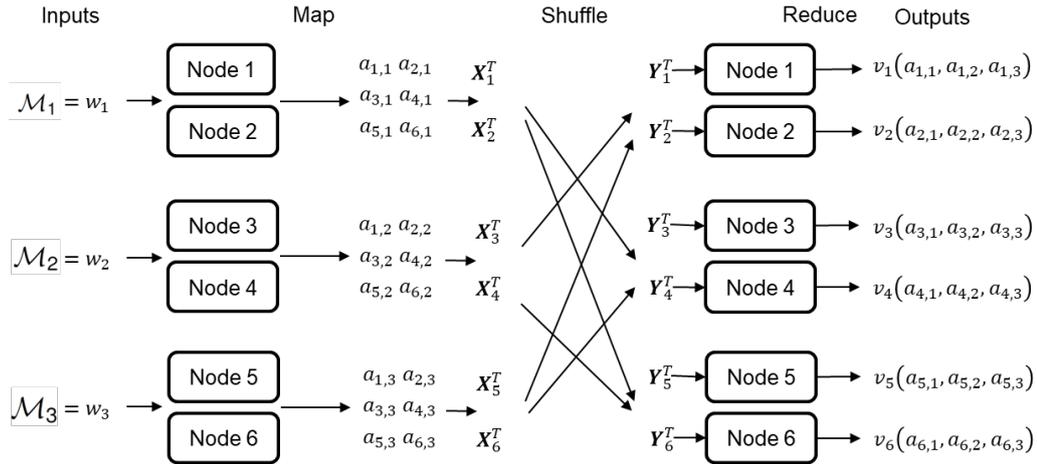
and the *normalized delivery time (NDT)*

$$\Delta = \lim_{\mathsf{P} \rightarrow \infty} \lim_{\mathsf{A} \rightarrow \infty} \frac{\mathsf{T}}{\mathsf{A} \cdot \mathsf{Q} \cdot \mathsf{N}} \cdot \log \mathsf{P}. \quad (2.32)$$

The computation load is defined identically as the one in wired DC system, representing the average number of nodes each file is assigned to. In other words, it is the ratio of the total number of stored input files (including replicas) to the total number of distinct files. Therefore, the computation load is always greater than 1 and less than K . NDT is the duration of wireless shuffle T the number of reduce functions Q and the number of input files N , and by the transmission time of one IVA over a point-to-point channel in the high SNR regime, i.e. $\mathsf{A}/\log(\mathsf{P})$. Here, the product $\mathsf{Q} \cdot \mathsf{N}$ corresponds to the total number of IVAs. The NDT equals to 1 only in the extreme case where all input files are stored on a single node, and the node broadcasts all IVAs using time-sharing, even when this is unnecessary. For a well-designed scheme, the actual number of IVAs transmitted through the channel is strictly less than total number of IVAs as many are only required locally. Furthermore, with communication schemes more efficient than time-sharing, the NDT is guaranteed to be less than 1.

We focus on the *fundamental computation-NDT tradeoff* $\Delta^*(r)$, which is defined as the infimum over all values of Δ satisfying (2.32) for some choice of file assignments $\{\mathcal{M}_p\}$, transmission time T , function assignment $\{\mathcal{Q}_p\}$, and encoding and decoding functions $\{f_p^{(\mathsf{T})}\}$ and $\{g_{q,i}^{(\mathsf{T})}\}$ in (2.28) and (2.30), all depending on A so that the probability IVA decoding error

$$\Pr \left[\bigcup_{p \in [\mathsf{K}]} \bigcup_{q \in \mathcal{Q}_p} \bigcup_{i \notin \mathcal{M}_p} \hat{a}_{q,i} \neq a_{q,i} \right] \rightarrow 0 \quad \text{as } \mathsf{A} \rightarrow \infty. \quad (2.33)$$


 Figure 2.4: Illustrative Example with $K = 6$ and $r = 2$

2.4.2 An Illustrative Example with $K = 6$ and $r = 2$

In Fig. 2.4, we illustrate an example setup with $K = Q = 6$ and $N = 3$, where 6 nodes are divided into three groups, with two nodes per group, i.e. $\mathcal{T}_1 = \{1, 2\}$, $\mathcal{T}_2 = \{3, 4\}$, and $\mathcal{T}_3 = \{5, 6\}$, each containing two nodes.

Map phase: During the map phase, assign the same input file to one entire group: w_1 to \mathcal{T}_1 , w_2 to \mathcal{T}_2 and w_3 to \mathcal{T}_3 . This setup clearly shows that the computation load $r = 2$, which meets the requirement.

Shuffle phase: Assign output function Φ_p to node p , i.e. $\mathcal{Q}_p = \{p\}$. During the shuffle phase, the nodes operate in full duplex, meaning that each node can act both as a transmitter Tx and a Rx. To apply the result in (2.20), we need to confirm the communication channel during the shuffle phase is equivalent to a CPXC. Consider Group \mathcal{T}_1 as an example. Group \mathcal{T}_1 send one IVA to each node outside the group (sending $a_{3,1}$ to node 3, $a_{4,1}$ to node 4 and so on). Meanwhile, nodes in \mathcal{T}_1 can send IVAs cooperatively as they issue the same input file w_1 . Nodes in Group \mathcal{T}_1 are able to cancel all interference inside the group for the same reason. This way forming the CPXC structure necessary for the lower bound to be applicable.

Therefore, the lower bound in (2.20) can be used to calculate the computation-NDT tradeoff. We will give the calculation detail in the next subsection.

2.4.3 Results on Normalized Delivery Time (NDT)

We observe that the channel model of the MapReduce system during the shuffle phase can be represented as a CPXC in Subsection 2.2.2 when a group input file assignment scheme is applied. As we assume each node in the MapReduce system is full duplex, it functions as both a Tx and a Rx simultaneously. Additionally, nodes in the same group can cancel the internal links as the same files are assigned. This configuration forms the necessary CPXC structure for the IA scheme to be applicable. Furthermore, it is straightforward to recognize that the NDT is inversely proportional to the SDoF lower bound by definition. Based on these insights, we provide the formal proof for the lower bound of the computation-NDT tradeoff in this section. Based on the lower bound in (2.24) we obtain:

Theorem 2.4 *Assume N and Q are both multiples of K . If N is large enough, the fundamental computation-NDT tradeoff of the full-duplex wireless DC system is upper bounded as*

$$\Delta^*(r) \leq \text{lowc} \left((K, 0) \cup \left\{ \left(r, \frac{1 - r/K}{\text{SDoFLb, CPXC}} \right) : 1 \leq r < K \text{ and } r|K \right\} \right), \quad (2.34)$$

where $\text{lowc}(\cdot)$ denotes the lower-convex envelope, SDoFLb, CPXC is defined in Eq. (2.20), and $r|K$ indicates that r divides K .

Proof: We prove the result for integer values of $r \in [K]$ that divide K . The final result follows by time- and memory-sharing arguments when N is sufficiently large.

We reuse the group definition in \mathcal{T}_k in (2.2).

Map phase: Choose the same file assignment for all nodes in group \mathcal{T}_k :

$$\mathcal{M}_p = \tilde{\mathcal{M}}_k \triangleq \left\{ (k-1)\frac{rN}{K} + 1, \dots, k\frac{rN}{K} \right\}, \quad p \in \mathcal{T}_k, k \in [\tilde{K}], \quad (2.35)$$

This file assignment satisfies the communication load r in (2.31).

Shuffle phase: Choose the output function assignment:

$$\mathcal{Q}_p \triangleq \{(p-1)Q/K + 1, \dots, pQ/K\}, \quad p \in [K]. \quad (2.36)$$

Further, choose a sequence (in $P > 0$) of rates $R(P) > 0$ such that

$$\overline{\lim}_{P \rightarrow \infty} \frac{R(P)}{\log P} = \frac{\text{SDoFLb}}{K(K/r - 1)} \quad (2.37)$$

and such that for each P the symmetric rate-tuple $(R_{p,k} = R(P), k \in [\tilde{K}], p \in [K] \setminus \mathcal{T}_k)$ lies inside the capacity region $\mathcal{C}(P)$ for the setup in Section 2.2. Fix a power P and consider a sequence (in T') of coding schemes $\{f_p^{(T')}\}_{T'}$ and $\{g_{p,k}^{(T')}\}_{T'}$ for the chosen rate-tuple such that $p^{(T')}$ (error) in (2.13) tends to 0 as $T' \rightarrow \infty$. By Theorem 2.2 and Remark 2.1, all the mentioned sequences exist.

The shuffle phase is split into rounds, where in each round, each group of nodes \mathcal{T}_k communicates a different IVA $a_{\nu,i}$ to each node $\ell \in [K] \setminus \mathcal{T}_k$, for chosen $\nu \in \mathcal{Q}_\ell$ and $i \in \tilde{\mathcal{M}}_k$. To send all missing IVAs, rounds Φ are necessary with

$$\Phi \triangleq |\mathcal{Q}_1| \cdot |\tilde{\mathcal{M}}_1| = (Q/K) \cdot (Nr/K) \quad (2.38)$$

Any node $p \in \mathcal{T}_k$ uses the chosen encoding function $f_p^{(T')}$ to send the IVAs in a given round, for a blocklength T' satisfying

$$\frac{A}{T'} < R(P). \quad (2.39)$$

Notice that all nodes in a group \mathcal{T}_k compute the same IVAs in the Map phase, and they can thus compute each others' inputs. Therefore, after receiving its channel outputs $Y_p^{(T')}$ in a given round, any Node $p \in \mathcal{T}_k$ first uses the IVAs it calculated during the Map phase to reconstruct and mitigate the signals sent by TxS in the same group \mathcal{T}_k :

$$\tilde{Y}_p(t) \triangleq Y_p(t) - \sum_{\ell \in \mathcal{T}_k} H_{p,\ell}(t) X_\ell(t), \quad p \in \mathcal{T}_k, t \in [T']. \quad (2.40)$$

Then, it applies the chosen decoding functions $\{g_{p,k}^{(\mathsf{T}')} : k \in [\tilde{\mathsf{K}}] \setminus [p/r]\}$ to reconstruct the IVAs sent to it in this round from all Tx-groups except for Tx-group $[p/r]$.

Analysis: By our choice of the coding scheme and (2.39), the probability of error in (2.33) tends to 0 as $\mathsf{T}' \rightarrow \infty$. Since the total length of the shuffle phase is $\mathsf{T} \triangleq \Phi \mathsf{T}'$, and by (2.38) and (2.39), the NDT of our scheme is:

$$\begin{aligned} \lim_{\mathsf{P} \rightarrow \infty} \lim_{\mathsf{A} \rightarrow \infty} \frac{\mathsf{T} \log \mathsf{P}}{\mathsf{A} \cdot \mathsf{Q} \cdot \mathsf{N}} &= \lim_{\mathsf{P} \rightarrow \infty} \lim_{\mathsf{A} \rightarrow \infty} \frac{\Phi \mathsf{T}' \log \mathsf{P}}{\mathsf{A} \cdot \mathsf{Q} \cdot \mathsf{N}} = \lim_{\mathsf{P} \rightarrow \infty} \lim_{\mathsf{A} \rightarrow \infty} \frac{r}{\mathsf{K}^2} \frac{\mathsf{T}' \log \mathsf{P}}{\mathsf{A}} \\ &\geq \lim_{\mathsf{P} \rightarrow \infty} \frac{r}{\mathsf{K}^2} \frac{\log \mathsf{P}}{\mathsf{R}(\mathsf{P})} = \frac{1 - \frac{r}{\mathsf{K}}}{\text{SDoF}_{\text{Lb}}}. \end{aligned} \quad (2.41)$$

This proves the desired achievability result. \blacksquare

The SDoF lower bound achieved by basic Time-Division Multiple Access (TDMA) scheme is 1, as only one IVA is sent for given time slot. We obtain the upper bound

$$\Delta^*(r) \leq \text{lowc}(\{(1 - r/\mathsf{K}) : 1 \leq r \leq \mathsf{K}\}). \quad (2.42)$$

With the similar proof in Section 2.4, we can deduce that the SDoF achieved by classic IA scheme in [30] is given by

$$\text{SDoF}'_{\text{Lb}} \triangleq \begin{cases} 2r & \text{if } \mathsf{K}/r \in \{2, 3\}, \\ \mathsf{K}/2 & \text{if } \mathsf{K}/r \geq 4, \end{cases} \quad (2.43)$$

The lower bound is obtained by applying (2.34).

The one-shot scheme in [26], which applies zero-forcing and side information cancellation, achieves the upper bound

$$\Delta^*(r) \leq \text{lowc}\left(\left\{\left(r, \frac{1 - r/\mathsf{K}}{\min(\mathsf{K}, 2r)}\right) : 1 \leq r \leq \mathsf{K}\right\}\right). \quad (2.44)$$

It is straightforward to see that TDMA scheme is sub-optimal compared to all three schemes for any given r and K , as they achieve SDoF greater than one. Meanwhile, the new IA scheme consistently outperforms the classic IA scheme. This can be prove by comparing SDoF_{Lb} in (2.20) with SDoF'_{Lb} in (2.43). These two bounds are identical when $\mathsf{K}/r \in \{2, 3\}$, and when when $\mathsf{K}/r \geq 4$,

$$\text{SDoF}_{\text{Lb}, \text{CPXC}} = \frac{\mathsf{K}(\mathsf{K} - r) - r^2}{2\mathsf{K} - 3r} \geq \frac{\mathsf{K}}{2} = \text{SDoF}'_{\text{Lb}},$$

where the inequality holds because it can be rewritten as $r(\mathsf{K} - 2r) \geq 0$.

Therefore, it is sufficient to compare the bound derived by the one-shot scheme with the bound derived by the new IA scheme. For fixed K and for r a value that divides K but neither equals $\mathsf{K}/2$ nor $\mathsf{K}/3$, our new upper bound in (2.34) is strictly better (lower) than the upper bound in (2.44). If K is even, the two bounds coincide on the interval $r \in [\mathsf{K}/2, \mathsf{K}]$, where they are given by the straight line $(1 - r/\mathsf{K})/\mathsf{K}$. If K is a multiple of 3, the two bounds also coincide for $r = \mathsf{K}/3$, where they are given by $(1 - r/\mathsf{K})/r$. For other values of r , the bound in (2.44) can be smaller. An improved upper bound on $\Delta^*(r)$ is thus obtained by combining the two upper bounds, which results in the lower-convex envelope of the union of the sets in (2.34) and (2.44).

In Fig. 2.5, we numerically compare the bounds in (2.34) and (2.44) for $K = 24$. We observe that on the interval $r \in [0, 8]$ the bound in (2.34) performs better and on the interval $r \in [8, 12]$ the bound (2.44) performs better because (2.34) is simply given by a straight line as $r \in \{9, 10, 11\}$ does not divide 24. On the interval $r \in [12, 24]$ both bounds perform equally-well as explained in the previous paragraph.

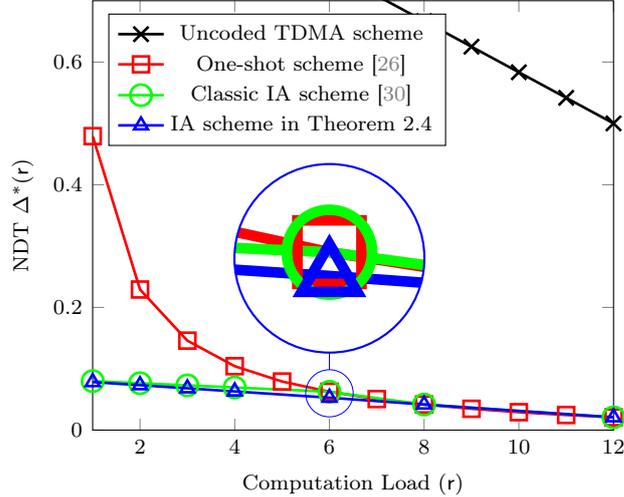


Figure 2.5: Upper bounds on $\Delta^*(r)$ for the TDMA scheme, the one-shot scheme in [26], the classic IA scheme and our new IA scheme when $K = 24$.

2.5 Application to Wireless DC with Associative Reduce Function

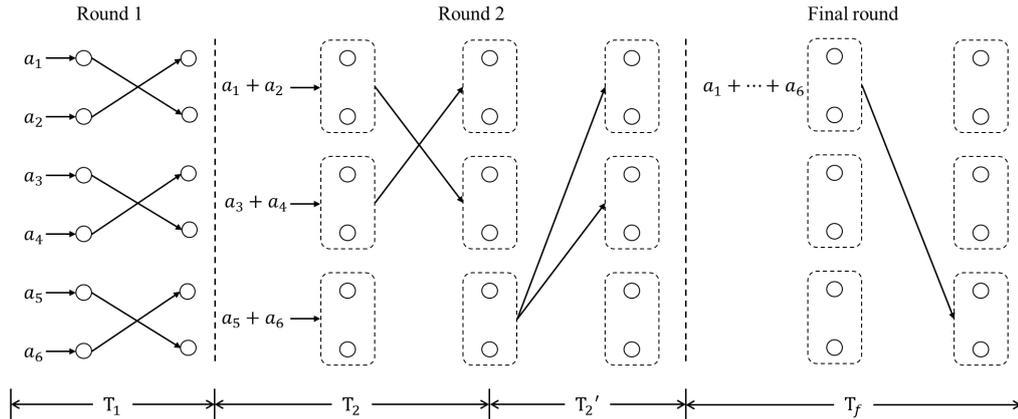
In this section, we consider a special MapReduce setup wherein every node executes an identical reduce function $\phi(a_1, \dots, a_N)$, i.e. $Q = 1$. We also require the function ϕ can be rewrite as

$$\phi(a_1, \dots, a_N) = a_1 * a_2 * \dots * a_N, \quad (2.45)$$

where $*$ is an associative binary operation which means $(a_1 * a_2) * a_3 = a_1 * (a_2 * a_3)$. For instance, the operation $*$ could be addition, multiplication, finding the minimum or more complex operations. Such a system is prevalent in decentralized distributed estimation and decentralized machine learning, where each node endeavors to compute the same objective function. Additionally, we assume $r = 1$ and $N = K$, as these systems typically operate on local data sources.

2.5.1 An Illustrative Example with $K = 6$

As shown in Fig 2.6, we consider a MapReduce system with 6 nodes. Each node $k \in [6]$ calculates its only IVA a_k , and the reduce function is simply the sum of all 6 IVAs. The shuffle phase is split into 2 rounds. In the first round, the nodes 1 and 2 are grouped together, node 3 and 4 are grouped, and node 5 and 6 are grouped. Then, the three groups exchange the IVAs between the two nodes. As this channel is identical to cooperative


 Figure 2.6: Illustrative Example with $K = 6$

interference channel with $K = 6$ and $r = 1$, each IVA can be sent with the data rate $R_1(P) = \frac{1}{2} \log P$ according to Theorem 2.3 and Remark 2.1, and the blocklength satisfies $T_1 > A/R_1(P)$. In the second round, nodes 1 and 2 calculate $a_1 + a_2$ and send cooperatively the sum to nodes 3 and 4. Nodes 3 and 4 also calculate $a_3 + a_4$ and send cooperatively the sum to nodes 5 and 6, while nodes 5 and 6 will be addressed later. As this channel is identical to cooperative interference channel with $K = 4$ and $r = 2$, each sum can be sent with the data rate $R_2(P) = \log P$. After IVAs are exchanged, nodes 5 and 6 broadcast $a_5 + a_6$ to all the other nodes with the data rate $R'_2(P) = 2 \log P$. The blocklength for the second phase is $T_2 + T'_2$ satisfying $T_2 > A/R_2(P)$ and $T'_2 > A/R'_2(P)$. Finally, nodes 1 to 4 are able to calculate the reduce function $\sum_{i=1}^6 a_i$ and the final result is sent cooperatively by nodes 1 and 2 to nodes 5 and 6 with the data rate $R_f(P) = 2 \log P$ and blocklength $T_f > A/R_f(P)$. Therefore, the total blocklength satisfies

$$\begin{aligned} T &> A/R_1(P) + A/R_2(P) + A/R'_2(P) + A/R_f(P) \\ &= 4 \cdot A/\log P. \end{aligned}$$

The computation-NDT tradeoff is thus $2/3$.

2.5.2 General Results

Theorem 2.5 *The fundamental computation-NDT tradeoff of the full-duplex wireless DC system with associative reduce function is upper bounded as*

$$\Delta^* \leq \frac{1}{K} \cdot \left(4 \left(1 - \frac{1}{2^{K_n}} \right) + \sum_{i=0}^{K_n-1} \frac{b_i}{2^i} + 1 \right), \quad (2.46)$$

where $K_n = \lfloor \log K \rfloor$ and $b_{K_n} b_{K_n-1} \cdots b_1 b_0$ represents the binary number of K .

Proof: As $r = 1$ and $N = K$, the file assignment scheme is straightforward. We directly discuss the shuffle phase.

Shuffle phase: The shuffle phase is split into $K_n = \lfloor \log K \rfloor$ rounds.

- In the first round, we divide nodes into groups with group-size 2 if K is even, i.e. $b_0 = 0$. The two nodes inside the same group exchange the locally calculated IVA. As

this channel is the same as the cooperative interference channel mentioned in Section 2.2.3, by Theorem 2.3 and Remark 2.1, we can deduce that the following data rate is achievable

$$\overline{\lim}_{P \rightarrow \infty} \frac{R(P)}{\log P} = \frac{1}{2}, \quad (2.47)$$

and the IVAs are sent for a blocklength satisfying

$$\frac{A}{T_1} < R(P). \quad (2.48)$$

Nodes in the same group calculate the intermediate result of the binary operation using the two IVA. If K is odd, the last node broadcast its IVA to all other nodes for a blocklength T'_1 satisfying

$$\overline{\lim}_{P \rightarrow \infty} \frac{A}{T'_1 \log P} < A. \quad (2.49)$$

Then we perform the operations mentioned above with $K - 1$ nodes.

- Similarly, in the i -th round, if the number of groups of the $i - 1$ round is even, i.e. $b_{i-1} = 0$, we combine two groups of the $i - 1$ round into one new group, and the group-size is 2^i . Without loss of generality, we can assume nodes in the j -th group are $\{(j-1)2^i + 1, (j-1)2^i + 2, \dots, j2^i\}$, and the first half of nodes possess the same intermediate result from previous rounds, i.e. each nodes $(j-1)2^i + 1, \dots, (j-1)2^i + 2^{(i-1)}$ knows

$$a_{(j-1)2^i+1} * a_{(j-1)2^i+2} * \dots * a_{(j-1)2^i+2^{(i-1)}}.$$

The other half of the nodes possess another intermediate result. Then, one intermediate result is transmitted cooperatively to the other half of group inside each group for a blocklength T_i satisfying

$$\overline{\lim}_{P \rightarrow \infty} \frac{A}{T_i \log P} < \frac{2A}{2^{(i-1)}}. \quad (2.50)$$

After the transmission, nodes in the same group calculate the new intermediate value $a_{(j-1)2^i+1} * \dots * a_{j2^i}$. If the number of groups of the $i - 1$ round is odd, the last group of the $i - 1$ round broadcast their intermediate result to all other nodes with a blocklength T'_i such that

$$\overline{\lim}_{P \rightarrow \infty} \frac{A}{T'_i \log P} < \frac{A}{2^{(i-1)}}, \quad (2.51)$$

and we perform the same operations with the rest groups.

- After K_n rounds, only one group remains, wherein nodes can calculate the reduce function ϕ and broadcast it to all nodes outside the group with a blocklength T_f such that

$$\overline{\lim}_{P \rightarrow \infty} \frac{A}{T_f \log P} < A, \quad (2.52)$$

Analysis: By our choice of the coding scheme, (2.47)-(2.52), the probability of error tends to 0 when the blocklength tends to infinity, the NDT of our scheme is

$$\begin{aligned} \lim_{P \rightarrow \infty} \lim_{A \rightarrow \infty} \frac{T \log P}{A \cdot K} \\ = \lim_{P \rightarrow \infty} \lim_{A \rightarrow \infty} \frac{\sum_{i=1}^{K_n} (T_i + T'_i) + T'_{K_n}}{A \cdot K / \log P} \end{aligned} \quad (2.53)$$

$$\geq \frac{1}{K} \cdot \left(\sum_{i=1}^{K_n} \frac{2}{2^{(i-1)}} + \sum_{i=1}^{K_n} \frac{b_{(i-1)}}{2^{(i-1)}} + 1 \right) \quad (2.54)$$

$$= \frac{1}{K} \cdot \left(4(1 - 2^{-K_n}) + \sum_{i=1}^{K_n} \frac{b_{(i-1)}}{2^{(i-1)}} + 1 \right) \quad (2.55)$$

This proves the desired achievability result. \blacksquare

Remark 2.2 From (2.46), we deduce that Δ^* is upper bounded by $\mathcal{O}(\frac{1}{K})$, as

$$(2.46) \leq \frac{1}{K} \left(4(1 - 2^{-K_n}) + \sum_{i=0}^{K_n-1} \frac{1}{2^i} + 1 \right) \quad (2.56)$$

$$= \frac{1}{K} \left(6(1 - 2^{-K_n}) + 1 \right) \quad (2.57)$$

$$\leq \frac{1}{K} \left(6\left(1 - \frac{1}{K}\right) + 1 \right) \quad (2.58)$$

This bound is significantly better compared to the simple TDMA scheme, which yields a bound of $\mathcal{O}(1)$.

2.6 Proof of the SDoF Bounds for NPC, CPXC and CPIX

In this section, we demonstrate the proofs for Theorem 2.1 , 2.2 and 2.3. The SDoF lower bound is proven by applying IA scheme. The SDoF upper bounds are proven by a information theoretic MAC argument.

2.6.1 Proof of the SDoF Lower Bound for NPC (Theorem 2.1)

Coding Scheme We fix a parameter $\eta \in \mathbb{Z}^+$ and a valid precoding index matrix \mathbf{G} . Define

$$p_{max} = \arg \max_{p \in [K]} \left\{ \|\mathbf{G}[p, :]\|_0 + g^{(p)} \right\}. \quad (2.59)$$

Choose

$$\mathbf{T} = \eta^\Gamma \|\mathbf{G}[p_{max}, :]\|_0 + (\eta + 1)^\Gamma g^{(p_{max})}. \quad (2.60)$$

where we specify the value of Γ later in Eq. (2.65). We only send messages $\{a_{p,q}\}$ with $\mathbf{G}[p, q] > 0$. Each message is encoded using a circularly symmetric Gaussian codebook of average power $P/\|\mathbf{G}[:, q]\|_0$ and codeword length η^Γ . Each codeword is sent over a block

of T consecutive channel uses. More precisely, let $\{\mathbf{b}_{p,q}\}$ denote the η^Γ -length codeword symbol for message $a_{p,q}$. Each Tx q form their inputs as:

$$\mathbf{X}_q = \sum_{\{p: \mathbf{G}[p,q]>0\}} \mathbf{U}_{\mathbf{G}[p,q]} \mathbf{b}_{p,q}, \quad (2.61)$$

where matrices $\{\mathbf{U}_g\}_{g \in [g_{max}]}$ are described shortly, and g_{max} is the maximal element in \mathbf{G} . Notice that messages $a_{p,q}$ and $a_{p',q'}$ are actually multiplied by the same precoding matrix if $\mathbf{G}[p,q] = \mathbf{G}[p',q'] > 0$.

We recall that Rx p only receives signals from connected TxS, i.e. from Tx q with $\mathbf{N}[p,q] = 1$, which allows to write the observed signal at each Rx $p \in [\mathsf{K}]$ as:

$$\begin{aligned} \mathbf{Y}_p = & \underbrace{\sum_{\{q: \mathbf{G}[p,q]>0\}} \mathbf{H}_{p,q} \mathbf{U}_{\mathbf{G}[p,q]} \mathbf{b}_{p,q}}_{\text{desired signal}} \\ & + \underbrace{\sum_{\{q: \mathbf{N}[p,q]=1\}} \sum_{\substack{\{p': p' \neq p \\ \mathbf{G}[p',q]>0\}}} \mathbf{H}_{p,q} \mathbf{U}_{\mathbf{G}[p',q]} \mathbf{b}_{p',q}}_{\text{Interference}} + \mathbf{Z}_p, \end{aligned} \quad (2.62)$$

where $\mathbf{H}_{p,q} \triangleq \text{diag}([H_{p,q}(1), H_{p,q}(2) \cdots H_{p,q}(\mathsf{T})])$, $\mathbf{Y}_p \triangleq (Y_p(1), \dots, Y_p(\mathsf{T}))^T$, and \mathbf{Z}_p are the corresponding Gaussian noise vectors observed at Rx p .

IA Matrices $\{\mathbf{U}_g\}$ Inspired by the IA scheme in [89], we choose each $\mathsf{T} \times \eta^\Gamma$ precoding matrix \mathbf{U}_g so that its column-span includes all power products (with powers from 1 to η) of the channel matrices $\mathbf{H}_{p,q}$ that premultiply \mathbf{U}_g in (2.62) when the coded symbol is treated as interference for the receiver. That implies, for $g \in [g_{max}]$:

$$\mathbf{U}_g = \left[\prod_{\mathbf{H} \in \mathcal{H}_g} \mathbf{H}^{\alpha_{g,\mathbf{H}}} \cdot \boldsymbol{\Xi}_g : \forall \alpha_g \in [\eta]^\Gamma \right], \quad (2.63)$$

where $\{\boldsymbol{\Xi}_g\}_{g \in \mathbf{G}}$ are i.i.d. random vectors independent of all channel matrices, noises, and messages,

$$\begin{aligned} \mathcal{H}_g \triangleq & \{ \mathbf{H}_{p,q} : \mathbf{N}[p,q] = 1 \text{ and } \exists p' \in [\mathsf{K}] \setminus \{p\} \\ & \text{such that } \mathbf{G}[p',q] = g \}, \end{aligned} \quad (2.64)$$

$\boldsymbol{\alpha}_g \triangleq (\alpha_{g,\mathbf{H}} : \mathbf{H} \in \mathcal{H}_g)$, and we assume the sizes of \mathcal{H}_g for $g \in [g_{max}]$ are equal ² and

$$\Gamma = |\mathcal{H}_g|. \quad (2.65)$$

Analysis of Signal-and-Interference Subspaces Since the column-span of \mathbf{U}_g contains all power products of powers 1 to η of the channel matrices $\mathbf{H}_{p,q}$ that premultiply \mathbf{U}_g

²If $|\mathcal{H}_g|$ varies for different g , additional i.i.d. diagonal random matrices can be added to \mathcal{H}_g with fewer element, and the rest of proof remains unchanged.

in (2.62), the product of any of these matrices with \mathbf{U}_g is included in the column-space of the $\mathbb{T} \times (\eta + 1)^\Gamma$ -matrix

$$\mathbf{W}_g = \left[\prod_{\mathbf{H} \in \mathcal{H}_g} \mathbf{H}^{\alpha_g, \mathbf{H}} \cdot \boldsymbol{\Xi}_g : \forall \alpha_g \in [\eta + 1]^\Gamma, g \in [g_{max}] \right], \quad (2.66)$$

Formally, for each $g \in [g_{max}]$ and $\mathbf{H} \in \mathcal{H}_g$, we have $\text{span}(\mathbf{H} \cdot \mathbf{U}_g) \subseteq \text{span}(\mathbf{W}_g)$. As a consequence, the signal and interference space at a Rx $p \in [\mathbb{K}]$ is represented by the matrix:

$$\boldsymbol{\Lambda}_p \triangleq [\mathbf{D}_p, \mathbf{I}_p]. \quad (2.67)$$

with the signal subspaces given by the $\mathbb{T} \times (\tilde{\mathbb{K}} - 1)\eta^\Gamma$ -matrices

$$\mathbf{D}_p \triangleq [\mathbf{H}_{p,q} \mathbf{U}_{\mathbf{G}[p,q]} : \mathbf{G}[p,q] > 0]. \quad (2.68)$$

and the interference subspaces given by

$$\mathbf{I}_p \triangleq [\mathbf{W}_{\mathbf{G}[p',q]} : \mathbf{N}[p,q] = 1, p' \neq p, \mathbf{G}[p',q] > 0] \quad (2.69)$$

By observing the signal subspace, we obtain the following property

Property 2.1 For matrix $\mathbf{H}_{p,q} \mathbf{U}_{\mathbf{G}[p,q]}$ in the signal subspace \mathbf{D}_p , we have $\mathbf{H}_{p,q} \notin \mathcal{H}_{\mathbf{G}[p,q]}$ according to the second condition on the valid precoding index matrix.

Proof: The property can be proven by contradiction. We assume $\mathbf{H}_{p,q} \in \mathcal{H}_{\mathbf{G}[p,q]}$. By the definition of \mathcal{H} in (2.64), we deduce that $\exists p' \in [\mathbb{K}] \setminus \{p\}$ such that $\mathbf{G}[p',q] = \mathbf{G}[p,q]$. There is thus at least one non-zero integer repeats within a column of \mathbf{G} , which violates the second requirement of a valid precoding index matrix in Definition 2.1. This concludes the proof. \blacksquare

We shall prove that all matrices $\{\boldsymbol{\Lambda}_p\}$ are of full column rank. This proves that the desired signals intended for Rx p can be separated from each other and from the interference space at this Rx. In the limits $\eta \rightarrow \infty$ (and thus $\mathbb{T} \rightarrow \infty$) and $\mathbb{P} \rightarrow \infty$, this establishes an DoF of

$$\lim_{\eta \rightarrow \infty} \frac{\|\mathbf{G}[p, :]\|_0 \eta^\Gamma}{\mathbb{T}} = \frac{\|\mathbf{G}[p, :]\|_0}{\|\mathbf{G}[p_{max}, :]\|_0 + g^{(p_{max})}}$$

at Rxs p . The SDoF of the entire system is thus given by $\text{SDoF}_{\text{Lb, NPC}}$, which establishes the desired result. In the following, we will explain in detail that the matrix $\boldsymbol{\Lambda}_p$ has the same form as the matrix \mathbf{A} in Lemma 2.1 and satisfy the two conditions mentioned in Lemma 2.1. Then any square submatrix of $\boldsymbol{\Lambda}_p$ has the same form as \mathbf{A} , which by Lemma 2.1 proves that the matrix $\boldsymbol{\Lambda}_p$ is full column rank with probability 1.

To see that $\boldsymbol{\Lambda}_p$ is of the form in (2.70), notice that all matrices involved in (2.67), i.e. $\{\mathbf{H}_{p,q}\}$, are diagonal, and their multiplications with an vector, i.e. $\{\boldsymbol{\Xi}_g\}$, from the right leads to a column-vector consisting of the non-zero entries of these diagonal matrices. More precisely, the random variables in row t are given by the slot- t channel coefficients $\{H_{p,q}(t)\}$ and the t -th elements of vector $\{\boldsymbol{\Xi}_g\}$, which by definition are independent of each other and

of all random variables in the other rows. Therefore, the matrix $\mathbf{\Lambda}_p$ satisfies Condition i) in Lemma 2.1. To see that it also satisfies Condition ii), notice that for any two distinct columns $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ selected from $\mathbf{\Lambda}_p$, the exponents in the corresponding columns differ because:

1. If $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are selected from the same signal subspace $\mathbf{H}_{p,q}\mathbf{U}_g$ or the same interference subspace \mathbf{W}_g , the two vectors have different exponents due to the construction method of \mathbf{U} and \mathbf{W} .
2. If $\mathbf{v}^{(1)}$ is selected from the signal subspace $\mathbf{H}_{p,q}\mathbf{U}_g$, and $\mathbf{v}^{(2)}$ is selected from the signal subspace $\mathbf{H}_{p,q'}\mathbf{U}_{g'}$ or interference subspace $\mathbf{W}_{g'}$ with $g \neq g'$, the two vectors have different exponents as they have distinct factors $\mathbf{\Xi}_g$ and $\mathbf{\Xi}_{g'}$.
3. If $\mathbf{v}^{(1)}$ is selected from the signal subspace $\mathbf{H}_{p,q}\mathbf{U}_g$, and $\mathbf{v}^{(2)}$ is selected from the signal subspace $\mathbf{H}_{p,q'}\mathbf{U}_g$ or interference subspace \mathbf{W}_g , we can deduce that $\mathbf{H}_{p,q} \notin \mathcal{H}_g$ by Property 2.1. $\mathbf{v}^{(1)}$ has the factor $\mathbf{H}_{p,q}$ while $\mathbf{v}^{(2)}$ does not. The two vectors thus have different exponents.

This concludes the proof.

Lemma 2.1 (Lemma 1 in [89]) Consider an M -by- M square matrix \mathbf{A} with i -th row and j -th column entry

$$a_{ij} = \prod_{\ell=1}^L \left(X_i^{[\ell]} \right)^{\alpha_{ij}^{[\ell]}}, \quad i, j \in M, \quad (2.70)$$

for random variables $\{X_i^{[\ell]}\}_{\ell \in [L]}$ and exponents

$$\boldsymbol{\alpha}_{ij} \triangleq \left(\alpha_{ij}^{[1]}, \alpha_{ij}^{[2]}, \dots, \alpha_{ij}^{[L]} \right) \in \mathbb{Z}^{+L}. \quad (2.71)$$

If

1. for any two pairs $(i, \ell) \neq (i', \ell')$ the conditional cumulative probability distribution $P_{X_i^{[\ell]} | X_{i'}^{[\ell]}}$ is continuous; and
2. any pair of vectors $\boldsymbol{\alpha}_{i,j} \neq \boldsymbol{\alpha}_{i,j'}$ for $i, j, j' \in [M]$ with $j \neq j'$;

then the matrix \mathbf{A} is full rank with probability 1.

2.6.2 Proof of the SDoF Lower Bound for CPXC (Theorem 2.2)

We first consider the case that $\tilde{K} \in \{2, 3\}$. Choose Tx/Rx-groups \mathcal{T}_1 and \mathcal{T}_2 , and ignore all Tx, Rx, and messages in the other groups. This reduces the network into two non-interfering r -user broadcast channels, one from Tx-group \mathcal{T}_1 to Rx-group \mathcal{T}_2 and the other from Tx-group \mathcal{T}_2 to Rx-group \mathcal{T}_1 , where SDoF r is achievable on each of them.

Then, we give the IA scheme for $\tilde{K} \geq 4$.

Coding Scheme We fix a parameter $\eta \in \mathbb{Z}^+$, define

$$\Gamma \triangleq \mathsf{K}(\mathsf{K} - 2), \quad (2.72)$$

and choose

$$\mathsf{T} = \eta^\Gamma (\tilde{\mathsf{K}} - 2) + (\eta + 1)^\Gamma (\tilde{\mathsf{K}} - 1). \quad (2.73)$$

Each message $\{a_{p,k}\}$ for $k \in [\tilde{\mathsf{K}}]$, $p \in [\mathsf{K}] \setminus \mathcal{T}_k$ —but not messages $\{a_{p,\tilde{\mathsf{K}}}\}_{p \in \mathcal{T}_1}$ which are not transmitted in our coding scheme—is encoded using a circularly symmetric Gaussian codebook of average power $\mathsf{P}/(\mathsf{K} - r)$ and codeword length η^Γ . Each codeword is sent over a block of T consecutive channel uses. More precisely, let $\mathbf{b}_{p,k}$ denote the η^Γ -length codeword symbol for message $a_{p,k}$ and define for each $j \in [\tilde{\mathsf{K}}]$, $k \in [\tilde{\mathsf{K}}] \setminus \{j\}$ and $(j, k) \neq (1, \tilde{\mathsf{K}})$ the vector

$$\tilde{\mathbf{b}}_{j,k} \triangleq \left(\mathbf{b}_{(j-1)r+1,k}^T, \mathbf{b}_{(j-1)r+2,k}^T, \dots, \mathbf{b}_{j \cdot r,k}^T \right)^T. \quad (2.74)$$

Group the channel inputs and outputs into the vectors

$$\mathbf{X}_q \triangleq (X_q(1), \dots, X_q(\mathsf{T}))^T, \quad q \in [\mathsf{K}], \quad (2.75)$$

$$\mathbf{Y}_p \triangleq (Y_p(1), \dots, Y_p(\mathsf{T}))^T, \quad p \in [\mathsf{K}], \quad (2.76)$$

and for each Tx-group $k \in [\tilde{\mathsf{K}}]$:

$$\tilde{\mathbf{X}}^{(k)} \triangleq \begin{pmatrix} \mathbf{X}_{(k-1) \cdot r+1} \\ \vdots \\ \mathbf{X}_{k \cdot r} \end{pmatrix} \quad (2.77)$$

and Rx-group $j \in [\tilde{\mathsf{K}}]$:

$$\tilde{\mathbf{Y}}^{(j)} \triangleq \begin{pmatrix} \mathbf{Y}_{(j-1) \cdot r+1} \\ \vdots \\ \mathbf{Y}_{j \cdot r} \end{pmatrix} = \sum_{k \neq j} \tilde{\mathbf{H}}^{(j,k)} \tilde{\mathbf{X}}^{(k)} + \tilde{\mathbf{Z}}^{(j)}, \quad (2.78)$$

where $\tilde{\mathbf{Z}}^{(j)}$ is the corresponding Gaussian vector and $\tilde{\mathbf{H}}^{(j,k)}$ is the $r\mathsf{T} \times r\mathsf{T}$ channel matrix

$$\tilde{\mathbf{H}}^{(j,k)} \triangleq \begin{pmatrix} \mathbf{H}_{(j-1) \cdot r+1, (k-1) \cdot r+1} & \cdots & \mathbf{H}_{(j-1) \cdot r+1, k \cdot r} \\ \vdots & & \vdots \\ \mathbf{H}_{j \cdot r, (k-1) \cdot r+1} & \cdots & \mathbf{H}_{j \cdot r, k \cdot r} \end{pmatrix} \quad (2.79)$$

and

$$\mathbf{H}_{p,q} \triangleq \text{diag}([H_{p,q}(1), H_{p,q}(2) \cdots H_{p,q}(\mathsf{T})]). \quad (2.80)$$

Tx-groups form their inputs as:

$$\tilde{\mathbf{X}}^{(1)} = \sum_{i=2}^{\tilde{\mathsf{K}}} \tilde{\mathbf{V}}^{(i,1)} \tilde{\mathbf{U}}_i \tilde{\mathbf{b}}_{i,1}, \quad (2.81)$$

$$\tilde{\mathbf{X}}^{(k)} = \sum_{i \in [\tilde{\mathsf{K}}] \setminus \{1,k\}} \tilde{\mathbf{V}}^{(i,k)} \tilde{\mathbf{U}}_i \tilde{\mathbf{b}}_{i,k} + \tilde{\mathbf{V}}^{(1,k)} \tilde{\mathbf{U}}_k \tilde{\mathbf{b}}_{1,k}, \quad k \in [\tilde{\mathsf{K}} - 1] \setminus \{1\}, \quad (2.82)$$

$$\tilde{\mathbf{X}}^{(\tilde{\mathsf{K}})} = \sum_{i=\{2, \dots, \tilde{\mathsf{K}}-1\}} \tilde{\mathbf{V}}^{(i, \tilde{\mathsf{K}})} \tilde{\mathbf{U}}_i \tilde{\mathbf{b}}_{i, \tilde{\mathsf{K}}}, \quad (2.83)$$

where

$$\tilde{\mathbf{U}}_i \triangleq \mathbf{I}_r \otimes \mathbf{U}_i, \quad i \in \{2, 3, \dots, \tilde{K}\}, \quad (2.84)$$

and matrices $\{\mathbf{U}_i\}$ and $\{\mathbf{V}^{(i,k)}\}$ are described shortly.

Notice that for $i \in \{2, 3, \dots, \tilde{K}\}$, messages $\{\tilde{\mathbf{a}}_{i,k}\}_{k \in [\tilde{K}] \setminus \{i\}}$ and $\tilde{\mathbf{a}}_{1,i}$ are multiplied by the same precoding matrix $\tilde{\mathbf{U}}_i$.

Zero-forcing Matrices $\{\tilde{\mathbf{V}}^{(i,k)}\}$ For each $i, k \in [\tilde{K}]$ with $i \neq k$, construct the $\mathbb{T} \times \mathbb{T}$ diagonal matrices $\mathbf{S}_1^{(i,k)}, \dots, \mathbf{S}_r^{(i,k)}$ by picking the real and imaginary parts of all non-zero entries i.i.d. according to a continuous distribution over $[-\mathbf{H}_{\max}, \mathbf{H}_{\max}]$ and form the diagonal matrix

$$\mathbf{S}^{(i,k)} \triangleq \begin{pmatrix} \mathbf{S}_1^{(i,k)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{S}_r^{(i,k)} \end{pmatrix}. \quad (2.85)$$

Choose the precoding matrices as:³

$$\tilde{\mathbf{V}}^{(i,k)} = \left(\tilde{\mathbf{H}}^{(i,k)} \right)^{-1} \mathbf{S}^{(i,k)}, \quad i, k \in [\tilde{K}], i \neq k, \quad (2.86)$$

so that all information sent to any Rx in group \mathcal{T}_j is zero-forced at all other Rxs in the same group \mathcal{T}_j . Defining for each triple $(i, j, k) \in [\tilde{K}]^3$ with $i \neq j, j \neq k, k \neq i$ the ‘‘generalized’’ channel matrix

$$\tilde{\mathbf{G}}_j^{(i,k)} = \begin{pmatrix} \mathbf{G}_{(j-1) \cdot r+1}^{((i-1) \cdot r+1, k)} & \dots & \mathbf{G}_{(j-1) \cdot r+1}^{(i \cdot r, k)} \\ \vdots & & \vdots \\ \mathbf{G}_{j \cdot r}^{((i-1) \cdot r+1, k)} & \dots & \mathbf{G}_{(j \cdot r)}^{i \cdot r, k} \end{pmatrix} \quad (2.87)$$

$$\triangleq \tilde{\mathbf{H}}^{(j,k)} \cdot \tilde{\mathbf{V}}^{(i,k)} = \tilde{\mathbf{H}}^{(j,k)} \left(\tilde{\mathbf{H}}^{(i,k)} \right)^{-1} \mathbf{S}^{(i,k)}, \quad (2.88)$$

allows to write the signals at the various Rx-groups as:

$$\tilde{\mathbf{Y}}^{(1)} = \underbrace{\sum_{k=2}^{\tilde{K}-1} \mathbf{S}^{(1,k)} \tilde{\mathbf{U}}_k \tilde{\mathbf{b}}_{1,k}}_{\text{desired signal}} + \sum_{i=2}^{\tilde{K}} \sum_{k \notin \{1, i\}} \tilde{\mathbf{G}}_1^{(i,k)} \tilde{\mathbf{U}}_i \tilde{\mathbf{b}}_{i,k} + \tilde{\mathbf{Z}}^{(1)}, \quad (2.89)$$

$$\begin{aligned} \tilde{\mathbf{Y}}^{(j)} &= \underbrace{\sum_{k \neq j} \mathbf{S}^{(j,k)} \tilde{\mathbf{U}}_k \tilde{\mathbf{b}}_{j,k}}_{\text{desired signal}} + \sum_{i \notin \{1, j\}} \sum_{k \notin \{i, j\}} \tilde{\mathbf{G}}_j^{(i,k)} \tilde{\mathbf{U}}_i \tilde{\mathbf{b}}_{i,k} \\ &+ \sum_{k \notin \{1, j, \tilde{K}\}} \tilde{\mathbf{G}}_j^{(1,k)} \tilde{\mathbf{U}}_k \tilde{\mathbf{b}}_{1,k} + \tilde{\mathbf{Z}}^{(j)}, \quad j \in [\tilde{K}] \setminus \{1\}. \end{aligned} \quad (2.90)$$

The third sum in (2.90) has $\tilde{K} - 2$ terms when $j = \tilde{K}$ but only $\tilde{K} - 3$ terms otherwise.

³We assume that all matrices $\{\tilde{\mathbf{H}}^{(i,k)}\}$ are invertible, which happens with probability 1. Otherwise, Tx and Rxs immediately declare an error in the communication. This probability-0 event, does not change the error probability of the system.

IA Matrices $\{\mathbf{U}_i\}$ Inspired by the IA scheme in [30], we choose each $\mathbb{T} \times \eta^\Gamma$ precoding matrix \mathbf{U}_i so that its column-span includes all power products (with powers from 1 to η) of the ‘‘generalized’’ channel matrices $\mathbf{G}_{p'}^{(p,k)}$ that premultiply \mathbf{U}_i in (2.89) and (2.90). That means:

$$\mathbf{U}_i = \left[\prod_{\mathbf{G} \in \mathcal{G}_i} \mathbf{G}^{\alpha_i, \mathbf{G}} \cdot \boldsymbol{\Xi}_i : \forall \alpha_i \in [\eta]^{\Gamma \cdot r^2} \right], \quad i \in [\tilde{\mathcal{K}}] \setminus \{1\}, \quad (2.91)$$

where $\{\boldsymbol{\Xi}_i\}_{i=2}^{\tilde{\mathcal{K}}}$ are i.i.d. random vectors independent of all channel matrices, noises, and messages, and

$$\begin{aligned} \mathcal{G}_i \triangleq & \left\{ \mathbf{G}_{p'}^{(p,k)} : p \in \mathcal{T}_i, k \in [\tilde{\mathcal{K}}] \setminus \mathcal{T}_i, p' \in [\mathcal{K}] \setminus (\mathcal{T}_i \cup \mathcal{T}_k) \right\} \\ & \cup \left\{ \mathbf{G}_p^{(p',i)} : p \in \mathcal{T}_1, p' \in [\mathcal{K}] \setminus \{\mathcal{T}_i \cup \mathcal{T}_1\} \right\}, \end{aligned} \quad (2.92)$$

and $\alpha_i \triangleq (\alpha_{i, \mathbf{G}} : \mathbf{G} \in \mathcal{G}_i)$.

Analysis of Signal-and-Interference Subspaces Since the column-span of \mathbf{U}_i contains all power products of powers 1 to η of the modified channel matrices $\mathbf{G} \in \mathcal{G}_i$ that premultiply \mathbf{U}_i in (2.89) and (2.90), the product of any of these matrices with \mathbf{U}_i is included in the column-space of the $\mathbb{T} \times \eta^\Gamma$ -matrix

$$\mathbf{W}_i = \left[\prod_{\mathbf{G} \in \mathcal{G}_i} \mathbf{G}^{\alpha_i, \mathbf{G}} \cdot \boldsymbol{\Xi}_i : \forall \alpha_i \in [\eta + 1]^{\Gamma \cdot r^2} \right], \quad i \in [\tilde{\mathcal{K}}] \setminus \{1\}, \quad (2.93)$$

where notice that $|\mathcal{G}_i| = \Gamma$. Formally, for each $i \in \{2, 3, \dots, \tilde{\mathcal{K}}\}$ and $\mathbf{G} \in \mathcal{G}_i$, we have $\text{span}(\mathbf{G} \cdot \mathbf{U}_i) \subseteq \text{span}(\mathbf{W}_i)$. As a consequence, the signal and interference space at a Rx $p \in \mathcal{T}_j$, for $j \in \{2, \dots, \tilde{\mathcal{K}}\}$, is represented by the matrix:

$$\mathbf{\Lambda}_p \triangleq \left[\underbrace{\mathbf{D}_p}_{\text{signal space}}, \underbrace{\mathbf{W}_2, \dots, \mathbf{W}_{j-1}, \mathbf{W}_{j+1}, \dots, \mathbf{W}_{\tilde{\mathcal{K}}}}_{\text{interference space}} \right]. \quad (2.94)$$

with the signal subspaces given by the $\mathbb{T} \times (\tilde{\mathcal{K}} - 1)\eta^\Gamma$ -matrices

$$\mathbf{D}_p \triangleq \left[\mathbf{S}_{p \bmod r}^{(j,k)} \cdot \mathbf{U}_j \right]_{k \in [\tilde{\mathcal{K}}] \setminus \{j\}}, \quad p \in \mathcal{T}_j. \quad (2.95)$$

For a Rx p in the first group \mathcal{T}_1 , the signal and interference spaces are represented by the $\mathbb{T} \times \mathbb{T}$ -matrix:

$$\mathbf{\Lambda}_p = \left[\underbrace{\mathbf{D}_{p,2}, \dots, \mathbf{D}_{p, \tilde{\mathcal{K}}-1}}_{\text{signal space}}, \underbrace{\mathbf{W}_2, \mathbf{W}_3, \dots, \mathbf{W}_{\tilde{\mathcal{K}}}}_{\text{interference space}} \right], \quad (2.96)$$

where the signal subspace is given by the $\mathbb{T} \times \eta^\Gamma$ -matrices

$$\mathbf{D}_{p,k} \triangleq \mathbf{S}_p^{(1,k)} \cdot \mathbf{U}_k, \quad k \in \{2, \dots, \tilde{\mathcal{K}} - 1\}, \quad p \in \mathcal{T}_1.$$

We shall prove that all matrices $\{\mathbf{\Lambda}_p\}$ are of full column rank. This proves that the desired signals intended for Rx p can be separated from each other and from the interference

space at this Rx. In the limits $\eta \rightarrow \infty$ (and thus $\mathsf{T} \rightarrow \infty$) and $\mathsf{P} \rightarrow \infty$, this establishes an SDoF of $\lim_{\eta \rightarrow \infty} \frac{(\tilde{K}-1)\eta^{\mathsf{F}}}{\mathsf{T}} = \frac{\tilde{K}-1}{2\tilde{K}-3}$ at Rxs $p \in [\tilde{K}] \setminus \mathcal{T}_1$ and an SDoF of $\frac{\tilde{K}-2}{2\tilde{K}-3}$ for Rxs $p \in \mathcal{T}_1$. The SDoF of the entire system is thus given by SDoF_{Lb} , which establishes the desired achievability result.

To prove the full rankness of each matrix $\mathbf{\Lambda}_p$, we introduce the following lemma.

Lemma 2.2 *Consider positive integers $n_1, n_2, \dots, n_{\tilde{K}}$ summing to $C \triangleq \sum_{i=1}^{\tilde{K}} n_i \leq \mathsf{T}$, and for each $i \in [\tilde{K}]$ and $k \in [n_i]$ a diagonal $\mathsf{T} \times \mathsf{T}$ matrix $\mathbf{B}_{i,k} \in \mathbb{C}$ so that all square sub-matrices of the following matrices are full rank:*

$$\mathbf{B}_i \triangleq [\mathbf{B}_{i,1} \cdot \mathbf{1}, \mathbf{B}_{i,2} \cdot \mathbf{1}, \dots, \mathbf{B}_{i,n_i} \cdot \mathbf{1}], \quad i \in [\tilde{K}]. \quad (2.97)$$

Let $\{\boldsymbol{\Xi}_i\}$ be independent T -length vectors with i.i.d. entries from continuous distributions and define the $\mathsf{T} \times n_i$ -matrices

$$\mathbf{A}_i \triangleq [\mathbf{B}_{i,1} \cdot \boldsymbol{\Xi}_i, \mathbf{B}_{i,2} \cdot \boldsymbol{\Xi}_i, \dots, \mathbf{B}_{i,n_i} \cdot \boldsymbol{\Xi}_i], \quad i \in [\tilde{K}]. \quad (2.98)$$

Then, the $\mathsf{T} \times C$ -matrix $\mathbf{\Lambda} \triangleq [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{\tilde{K}}]$ has full column rank with probability 1.

Proof: We present the proof for the case that the matrix $\mathbf{\Lambda}$ is square, i.e., $C = \mathsf{T}$. If $\mathsf{T} > C$, we take a square C -by- C submatrix of $\mathbf{\Lambda}$ and perform the same proof steps on the submatrix.

Define

$$F(\boldsymbol{\Xi}_1, \dots, \boldsymbol{\Xi}_{\tilde{K}}) \triangleq \det(\mathbf{\Lambda}) \quad (2.99)$$

which is a polynomial of $\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2, \dots, \boldsymbol{\Xi}_{\tilde{K}}$ as the determinant is a polynomial of the entries of $\mathbf{\Lambda}$.

For the vectors

$$\boldsymbol{\xi}_i = \begin{bmatrix} \underbrace{0, \dots, 0}_{(n_1 + \dots + n_{i-1}) \text{ 0s}}, & \underbrace{1, \dots, 1}_{n_i \text{ 1s}}, & \underbrace{0, \dots, 0}_{(n_{i+1} + \dots + n_{\tilde{K}}) \text{ 0s}} \end{bmatrix}^T, \quad i \in [\tilde{K}], \quad (2.100)$$

the polynomial evaluates to

$$F(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{\tilde{K}}) = \det \begin{pmatrix} \mathbf{B}'_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}'_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}'_{\tilde{K}} \end{pmatrix} \quad (2.101)$$

$$= \prod_{i=1}^{\tilde{K}} \det(\mathbf{B}'_i) \neq 0 \quad (2.102)$$

where \mathbf{B}'_i is the $n_i \times n_i$ square sub-matrix of \mathbf{B}_i consisting of its rows $(n_1 + \dots + n_{i-1} + 1)$ to $(n_1 + \dots + n_{i-1} + n_i)$. As all square sub-matrices of \mathbf{B}_i are full rank, any matrix \mathbf{B}'_i for $i \in [\tilde{K}]$ is also full-rank. This leads to $\det(\mathbf{B}'_i) \neq 0$ for $i \in [\tilde{K}]$. Consequently (2.102) holds.

We conclude that F is a non-zero polynomial and thus $F(\boldsymbol{\Xi}_1, \dots, \boldsymbol{\Xi}_{\tilde{K}})$ equals 0 with probability 0 because the entries of $\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2, \dots, \boldsymbol{\Xi}_{\tilde{K}}$ are drawn independently from continuous distributions. \blacksquare

Notice that each matrix \mathbf{A}_p , for $p \in [K]$, is of the form of the matrix \mathbf{A} in Lemma 2.2 at the end of this section. Defining the matrices $\{\hat{\mathbf{U}}_i\}$, $\{\hat{\mathbf{W}}_i\}$, $\{\hat{\mathbf{D}}_p\}$ and $\{\hat{\mathbf{D}}_{p,k}\}$ in the same way as $\{\mathbf{U}_i\}$, $\{\mathbf{W}_i\}$, $\{\mathbf{D}_p\}$, and $\{\mathbf{D}_{p,k}\}$ but with $\mathbf{\Xi}_i$ replaced by the all-one vector $\mathbf{1}$, it suffices to show that with probability 1 all square submatrices of the following matrices (which play the roles of $\{\mathbf{B}_i\}$ when applying Lemma 2.2) are full rank:

$$\{\hat{\mathbf{D}}_p\}_{p \in [\tilde{K}] \setminus \mathcal{T}_1}, \quad \{\hat{\mathbf{W}}_j\}_{j=2}^{\tilde{K}}, \quad \left\{ \left[\hat{\mathbf{D}}_{p,j}, \hat{\mathbf{W}}_j \right] \right\}_{\substack{p \in \mathcal{T}_1 \\ j \in \{2, \dots, \tilde{K}\}}} . \quad (2.103)$$

For matrix $\hat{\mathbf{D}}_p$, $p \in \mathcal{T}_2$, this proof is provided in Appendix A. The proofs for the other matrices follow a similar argument.

2.6.3 Proof of the SDoF Upper Bounds for CPXC and CPIC (Theorem 2.2 and 2.3)

The proof follows immediately by summing up the upper bound in the following Lemma 2.3 for the $\tilde{K}(\tilde{K} - 1)$ distinct pairs $(j, k) \in [\tilde{K}] \times [\tilde{K}]$ with $j \neq k$, and then dividing this sum by $2\tilde{K} - 3$, because each rate has been counted $2\tilde{K} - 3$ times.

Lemma 2.3 *Let $(R_{p,k}(P) : k \in [\tilde{K}], p \in [K] \setminus \mathcal{T}_k)$ be a rate-tuple in $\mathcal{C}(P)$, for each $P > 0$. Then, for any $j, k \in [\tilde{K}]$ with $j \neq k$:*

$$\overline{\lim}_{P \rightarrow \infty} \left[\sum_{p \in \mathcal{T}_j} \sum_{\ell \in [\tilde{K}] \setminus \{j\}} \frac{R_{p,\ell}}{\log P} + \sum_{p \in [K] \setminus (\mathcal{T}_k \cup \mathcal{T}_j)} \frac{R_{p,k}}{\log P} \right] \leq r. \quad (2.104)$$

Proof: Fix $P > 0$ and any rate tuple $(R_{p,k}(P) : k \in [\tilde{K}], p \in [K] \setminus \mathcal{T}_k)$ in $\mathcal{C}(P)$. Then consider a sequence of encoding and decoding functions $\{f_q^{(T)}\}$ and $\{g_{p,k}^{(T)}\}$ such that $p^{(T)}(\text{error})$ tends to 0 as $T \rightarrow \infty$.

Fix a blocklength T and indices $j, k \in [\tilde{K}]$ with $j \neq k$, and define

$$\mathcal{F} \triangleq [K] \setminus (\mathcal{T}_k \cup \mathcal{T}_j) \quad (2.105)$$

Partition the set of messages into the following three sets

$$\mathbf{a}_r \triangleq \{a_{p,\ell} : p \in \mathcal{T}_j, \ell \in [\tilde{K}] \setminus \{j\}\} \quad (2.106)$$

$$\mathbf{a}_t \triangleq \{a_{p,k} : p \in \mathcal{F}\} \quad (2.107)$$

$$\mathbf{a}_c \triangleq \{a_{p,\ell} : p \notin \mathcal{T}_j, \ell \neq k\}. \quad (2.108)$$

Finally, denote by \mathcal{H} the set of *all* channel coefficients in the system, and for any subset $\mathcal{S} \subseteq [K]$ define $\mathbf{Y}_{\mathcal{S}} \triangleq (Y_p^{(T)} : p \in \mathcal{S})$ and $\mathbf{Z}_{\mathcal{S}} \triangleq (Z_p^{(T)} : p \in \mathcal{S})$.

Notice now that by the independence of the IVAs, the channel coefficients, and the

noise sequences:

$$\sum_{p \in \mathcal{T}_j} \sum_{\ell \in [\bar{K}] \setminus \{j\}} \text{TR}_{p,\ell} + \sum_{p \in [\bar{K}] \setminus (\mathcal{T}_k \cup \mathcal{T}_j)} \text{TR}_{p,k} \quad (2.109)$$

$$= H(\mathbf{a}_r, \mathbf{a}_t) \quad (2.110)$$

$$= H(\mathbf{a}_t, \mathbf{a}_r | \mathbf{a}_c, \mathcal{H}) \quad (2.111)$$

$$= I(\mathbf{a}_t, \mathbf{a}_r; \mathbf{Y}_{\mathcal{R}_j} | \mathbf{a}_c, \mathcal{H}) + H(\mathbf{a}_t, \mathbf{a}_r | \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) \quad (2.112)$$

$$= h(\mathbf{Y}_{\mathcal{R}_j} | \mathbf{a}_c, \mathcal{H}) - h(\mathbf{Z}_{\mathcal{R}_j}) \\ + H(\mathbf{a}_r | \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) + H(\mathbf{a}_t | \mathbf{a}_r, \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) \quad (2.113)$$

$$\leq \mathsf{T} \cdot r \log(1 + \text{PH}_{\max}^2(\bar{K} - r)) \\ + H(\mathbf{a}_r | \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) + H(\mathbf{a}_t | \mathbf{a}_r, \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}). \quad (2.114)$$

If communication is reliable, it is possible to reconstruct \mathbf{a}_r from $\mathbf{Y}_{\mathcal{T}_j}$ with probability of error tending to 0 as $\mathsf{T} \rightarrow \infty$. Therefore, by Fano's inequality

$$H(\mathbf{a}_r | \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) \leq \mathsf{T} \cdot \epsilon_{\mathsf{T}}, \quad (2.115)$$

for some sequence $\{\epsilon_{\mathsf{T}}\}$ tending to 0 as $\mathsf{T} \rightarrow \infty$.

To bound the last summand in (2.114), we further notice that for reliable communication, Fano's inequality also implies

$$H(\mathbf{a}_t | \mathbf{a}_r, \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathbf{Y}_{\mathcal{F}}, \mathcal{H}) \leq \mathsf{T} \tilde{\epsilon}_{\mathsf{T}}, \quad (2.116)$$

for some sequence $\{\tilde{\epsilon}_{\mathsf{T}}\}$ tending to 0 as $\mathsf{T} \rightarrow \infty$. Thus,

$$H(\mathbf{a}_t | \mathbf{a}_r, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) \\ \leq H(\mathbf{a}_t | \mathbf{a}_r, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) - H(\mathbf{a}_t | \mathbf{a}_r, \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathbf{Y}_{\mathcal{F}}, \mathcal{H}) + \mathsf{T} \tilde{\epsilon}_{\mathsf{T}} \quad (2.117)$$

$$= I(\mathbf{a}_t; \mathbf{Y}_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}) + \mathsf{T} \tilde{\epsilon}_{\mathsf{T}} \quad (2.118)$$

$$\stackrel{(a)}{=} h(\tilde{\mathbf{Y}}_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \tilde{\mathbf{Y}}_{\mathcal{R}_j}, \mathcal{H}) - h(\mathbf{Z}_{\mathcal{F}}) + \mathsf{T} \tilde{\epsilon}_{\mathsf{T}} \quad (2.119)$$

$$\stackrel{(b)}{=} \mathbb{P}(E = 1) \cdot h(\tilde{\mathbf{Y}}_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \tilde{\mathbf{Y}}_{\mathcal{R}_j}, \mathcal{H}, E = 1) \\ + \mathbb{P}(E = 0) \cdot h(\tilde{\mathbf{Y}}_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \tilde{\mathbf{Y}}_{\mathcal{R}_j}, \mathcal{H}, E = 0) \\ - h(\mathbf{Z}_{\mathcal{F}}) + \mathsf{T} \tilde{\epsilon}_{\mathsf{T}} \quad (2.120)$$

$$\stackrel{(c)}{=} h(\tilde{\mathbf{Y}}_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \tilde{\mathbf{Y}}_{\mathcal{R}_j}, \mathcal{H}, E = 1) - h(\mathbf{Z}_{\mathcal{F}}) + \mathsf{T} \tilde{\epsilon}_{\mathsf{T}} \quad (2.121)$$

$$\stackrel{(d)}{\leq} h(\mathbf{Z}'_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \tilde{\mathbf{Y}}_{\mathcal{R}_j}, \mathcal{H}, E = 1) - h(\mathbf{Z}_{\mathcal{F}}) + \mathsf{T} \tilde{\epsilon}_{\mathsf{T}} \quad (2.122)$$

$$\stackrel{(e)}{\leq} h(\mathbf{Z}'_{\mathcal{F}}) - h(\mathbf{Z}_{\mathcal{F}}) + \mathsf{T} \cdot \tilde{\epsilon}_{\mathsf{T}}, \quad (2.123)$$

where

- in (a) we defined for any $\mathcal{S} \subseteq [\bar{K}]$ the tuple $\tilde{\mathbf{Y}}_{\mathcal{S}} \triangleq (\tilde{Y}_p^{(T)} : p \in \mathcal{S})$ with $\tilde{Y}_p^{(T)} \triangleq (\tilde{Y}_p(1), \dots, \tilde{Y}_p(\mathsf{T}))$ and

$$\tilde{Y}_p(t) \triangleq \sum_{\ell \in \mathcal{T}_k} H_{p,\ell}(t) X_{\ell}(t) + Z_p(t), \quad p \in \mathcal{S}; \quad (2.124)$$

- in (b) we defined the random variable E equal to 1 if each input $X_q(t)$, for $q \in \mathcal{T}_k$ and $t \in [\mathsf{T}]$, can be obtained as a linear combination of the entries in $\tilde{\mathbf{Y}}_{\mathcal{T}_j} - \mathbf{Z}_{\mathcal{T}_j}$. If it exists, we write this linear combination as

$$X_q(t) = \mathcal{L}_{q,t}(\tilde{\mathbf{Y}}_{\mathcal{T}_j} - \mathbf{Z}_{\mathcal{T}_j}). \quad (2.125)$$

Notice that the entries in $\tilde{\mathbf{Y}}_{\mathcal{T}_j} - \mathbf{Z}_{\mathcal{T}_j}$ are themselves linear combinations of the inputs $\{X_q(t) : q \in \mathcal{T}_k, t \in [\mathsf{T}]\}$, and therefore the existence of linear functions $\mathcal{L}_{q,t}$ in (2.125) is equivalent to a given square matrix of channel coefficients being invertible.

- (c) holds because $h(\tilde{\mathbf{Y}}_{\mathcal{F}} | \mathbf{a}_r, \mathbf{a}_c, \mathbf{Y}_{\mathcal{R}_j}, \mathcal{H}, E = 0)$ is bounded and because $\mathbb{P}(E = 1) = 1$. This latter fact holds because $E = 1$ whenever a specific square matrix of the random channel coefficients is invertible (see (b) above), which happens with probability 1 since the channel coefficients are independently drawn from a continuous distribution;
- in (d) we defined the tuple $\mathbf{Z}'_j \triangleq (Z'_p(t) : p \in \mathcal{T}_j, t \in [\mathsf{T}])$ and

$$Z'_p(t) \triangleq Z_p(t) - \sum_{q \in \mathcal{T}_k} H_{q,p}(t) \mathcal{L}_{p,t}(\tilde{\mathbf{Y}}_{\mathcal{T}_j} - \mathbf{Z}_{\mathcal{T}_j}), \quad (2.126)$$

where $\mathcal{L}_{q,t}$ is from (b);

- in (e) we used the independence of the noise from the channel coefficients and the fact that conditioning can only decrease differential entropy.

Finally, combining Eqs. (2.114), (2.115), and (2.123), we obtain

$$\begin{aligned} & \sum_{p \in \mathcal{T}_j} \sum_{\ell \in [\tilde{\mathsf{K}}] \setminus \{j\}} R_{p,\ell} + \sum_{p \in [\mathsf{K}] \setminus (\mathcal{T}_k \cup \mathcal{T}_j)} R_{p,k} \\ & \leq r \log(1 + \text{PH}_{\max}^2(\mathsf{K} - r)) + \epsilon_{\mathsf{T}} \\ & \quad + \frac{1}{\mathsf{T}} h(\mathbf{Z}'_{\mathcal{F}}) - \frac{1}{\mathsf{T}} h(\mathbf{Z}_{\mathcal{F}}) + \tilde{\epsilon}_{\mathsf{T}}. \end{aligned} \quad (2.127)$$

Letting $\mathsf{T} \rightarrow \infty$ and $\mathsf{P} \rightarrow \infty$ establishes the desired inequality in the lemma, because $h(\mathbf{Z}'_{\mathcal{F}})/\mathsf{T}$ and $h(\mathbf{Z}_{\mathcal{F}})/\mathsf{T}$ are both finite constants that do not depend on T nor P , and both sequences ϵ_{T} and $\tilde{\epsilon}_{\mathsf{T}}$ tend to 0 as $\mathsf{T} \rightarrow \infty$. \blacksquare

By replacing notations of X-channel by interference channel and shuffling indexes, we obtain the following lemma for cooperative interference channel.

Lemma 2.4 *Let $(R_k(\mathsf{P}) : k \in [\tilde{\mathsf{K}}], p \in [\mathsf{K}] \setminus \mathcal{T}_k)$ be a rate-tuple in $\mathcal{C}(\mathsf{P})$, for each $\mathsf{P} > 0$. Then, for any $j, k \in [\tilde{\mathsf{K}}]$ with $j \neq k$:*

$$\overline{\lim}_{\mathsf{P} \rightarrow \infty} \left[\frac{R_j}{\log \mathsf{P}} + \frac{R_k}{\log \mathsf{P}} \right] \leq r. \quad (2.128)$$

Then, the upper bound of Theorem 2.3 is proved by summing up the bound in (2.128) for all $\tilde{\mathsf{K}}(\tilde{\mathsf{K}} - 1)$ distinct pairs $(j, k) \in [\tilde{\mathsf{K}}]^2$.

2.7 Conclusion

In this chapter, We investigated the SDoF bounds for three subclasses of partially connected channels: NPC, CPXC and CPIC. Specifically, We derived a new lower bound for NPC, a improved lower bound and an information-theoretic upper bound for CPXC, and the exact SDoF for CPIC. The proposed SDoF lower bounds of CPXC and CPIC were utilized to achieve an improved NDT for wireless distributed Map-Reduce systems.

Bounds on NDT of Wireless DC

3.1 Introduction

In this chapter, we further improve the computation-NDT tradeoff of MapReduce over full-duplex wireless interference channels with two novel IA schemes. Our first scheme is inspired by the IA scheme in [28], where multi-cast messages are sent over a fully-connected interference network. We however adapt this scheme to our DC setup, where nodes simultaneously act as transmitters and receiver, allowing to achieve improved performances. In fact, the equivalence of transmitting and receiving terminals implies only a partial connectivity in the corresponding interference network, which we can exploit through an improved reutilization of IA precoding matrices compared to [28]. More in detail, the IA scheme in [28] assigns a dedicated IA precoding matrix to each set of r receivers, and then uses this precoding matrix for all transmissions that are intended exclusively for this set of receivers. In our scheme we do not assign dedicated precoding matrices to receive sets containing user 1, but instead reuse the other precoding matrices for these transmissions. This trick is possible in our setup because transmitters coincide with receivers and thus the i -th receiver does not observe any signal from the i -th transmitter. The advantage of reducing the number of precoding matrices is that nodes $2, \dots, K$ suffer from fewer interference spaces (since each interference space corresponds to a precoding matrix), thus leaving a larger part of their receive dimensions as signal space and resulting in a improved performance. In the special case of $r = 1$, i.e., when each file is stored only at a single node, the first scheme coincides with the scheme of CPXC in the previous chapter.

We present a second IA-DC scheme for systems with an odd number of users K and computation load $r = \frac{K-1}{2}$, i.e., when each node can store almost half of the input files. In this second scheme each node only sees interference pertaining to one of the K utilized IA-precoding matrices, while all other non-intended transmissions at this node are zero-forced. In this sense, the presented IA scheme implies minimum interference space (because any non-trivial IA scheme has interference pertaining to at least one of the precoding matrices), allowing to obtain improved performances compared to other IA schemes. To achieve this minimum interference, in our scheme, all IVAs are cooperatively transmitted by $r = \frac{K-1}{2}$ transmitters, and zero-forced at a set of $r - 1$ receiving nodes. Since $2 - r - 1 = K - 2$, each transmission will thus only create interference at one of the non-intended

receivers, the receiver that corresponds to the utilized IA-precoding matrix. We point out that zero-forcing influences the construction of the precoding matrices. Under these new constructions it is more complicated to prove that signals corresponding to a given precoding matrix align at the receivers. Our contribution in this part is to carefully assign the precoding matrices to the different transmissions, making the system amenable for a proof and achieving good performance. This assignment is rather technical and we describe it further when we present our scheme.

The upper bound on the NDT implied by our two new IA-schemes improves over the previously proposed bounds in [26] and Theorem 2.4 in the previous chapter whenever the computation load $1 < r < \frac{K}{2}$, i.e., when each file can be stored at more than one node, but each node cannot store half of the total number of files. As already mentioned, for $r = 1$ our upper bound recovers the NDT-bound in Theorem 2.4, which for this value improves over the NDT upper bound in [26]. Since [26] achieves the optimal NDT when restricting to zero-forcing and interference cancellation, our results show that these techniques fail to achieve the optimal NDT for all computation loads $1 \leq r < \frac{K}{2}$, i.e., whenever nodes cannot store half of the number of input files. On the contrary, in this manuscript we show that for $r \geq \frac{K}{2}$ the zero-forcing and interference cancellation scheme in [26] is even optimal among *all* coding scheme (beyond the class of zero-forcing and interference cancellation schemes).

In fact, we also present an information-theoretic lower bound on the computation-NDT tradeoff based on a MAC type argument that is applied in parallel to a set of well-selected sub-systems and by solving a resulting linear program. For computation load $r < \frac{K}{2}$ the lower bound on the NDT is close to the proposed upper bound, but they do not match. As mentioned, for $r \geq \frac{K}{2}$ the lower bound matches the upper bound in [26] thus establishing the exact NDT of wireless MapReduce over full-duplex networks.

To summarize, the main contributions of this chapter are:

- Improved coding schemes based on IA for wireless MapReduce over full-duplex interference networks. (Sections 3.4–3.6 and Theorem 3.1 and Corollary 3.2)
- A lower bound (converse) on the NDT of wireless MapReduce systems. (Theorem 3.1)
- The exact NDT of wireless MapReduce for computation loads $r \geq \lceil \frac{K}{2} \rceil$. (Corollary 3.1)
- Proof that zero-forcing and interference cancellation cannot achieve the NDT tradeoff for all computation loads $1 < r < \lceil \frac{K}{2} \rceil$. (Remark 3.2)

This chapter is outlined as follows. We terminate this section with notation. The following Section 3.2 describes the detailed system model, while Section 3.3 presents and discusses our bounds on the NDT tradeoff. Sections 3.5 and 3.6 explain our two novel IA schemes, where in Section 3.4, we first describe our first novel IA scheme using some simple examples. Section 3.7 proves our NDT-lower bound.

3.2 Wireless MapReduce Systems

The MapReduce system, introduced in Section 2.4.1 of the previous chapter, is revisited in this chapter with slight modifications to some notations. We focus on the specific case where the number of nodes equals the number of output functions and review some key definitions in this section.

We consider a DC system consisting of a fixed number of K nodes, labeled $1, \dots, K$; a large number N of input files, denoted as W_1, \dots, W_N ; and K output functions, h_1, \dots, h_K , which map the input files to the desired computations. Each output function h_k is assigned to node k , where $k \in [K]$.

The definitions of the reduce function ϕ_q and the IVA remain unchanged, as provided in (2.26) and (2.27), respectively. For simplicity, we introduce new notation for the channel input and output:

$$\mathbf{X}_k \triangleq (X_k(1), \dots, X_k(T))^T = f_k^{(T)}(\{a_{1,p}, \dots, a_{K,p}\}_{p \in \mathcal{M}_k}), \quad (3.1)$$

$$\mathbf{Y}_k \triangleq (Y_k(1), \dots, Y_k(T))^T, \quad (3.2)$$

while preserving all other definitions for the wireless channel from (2.28) to (2.30).

We also recall two key performance metrics for wireless DC systems: the computation load r , defined in (2.31), and the NDT, defined in (2.32). Our primary focus remains the *fundamental computation-NDT tradeoff* $\Delta^*(r)$, whose definition was provided in the final paragraph of Section 2.4.1.

3.2.1 Sufficiency of Symmetric File Assignments

Our model exhibits a perfect symmetry between the various nodes in the network in the sense that the channels from any Tx-node to any Rx-node has same statistical behaviour and the various channels are independent of each other. The optimal computation-NDT tradeoff is therefore achieved by a symmetric file assignment where any subset of nodes $\mathcal{T} \subseteq [K]$ of size i is assigned the same number of files to be stored at all nodes in \mathcal{T} . In fact, any non-symmetric file assignment can be symmetrized without decreasing the computation-NDT tradeoff. It suffices to time-share $K!$ instances of the original scheme for a number of files N that is also multiplied by $K!$, where in each instance the K nodes are relabeled according to a different permutation and a different subset of files is used. The resulting scheme has a symmetric file assignment and achieves the same computation-NDT tradeoff as the original scheme because $|\mathcal{M}_k|$, N , and T are multiplied by $K!$ while the other parameters remain unchanged and because the new scheme still satisfies (2.33) whenever the original scheme satisfies (2.33).

By the optimality of symmetric file assignments, the optimization problem over the optimal file assignment reduces to finding the optimal fraction of files that should be assigned to exactly i nodes, for any $i \in [K]$. It is well-known that when communication is over noiseless broadcast links, then it suffices to assign some of the files to $\lfloor r \rfloor$ nodes and the remaining files to $\lceil r \rceil$ nodes. We apply the same strategy in this chapter. For $r < \lfloor \frac{K}{2} \rfloor$ however we cannot prove optimality of these file assignments.

3.2.2 Relation to the Network's SDoF with r -fold Cooperation

A well-studied property of wireless networks is the *SDoF*, which characterizes the maximum throughput of a network. In this chapter we are specifically interested in the SDoF that one can achieve over the wireless network described by (2.29), when the inputs are subject to the average power constraints (2.6) and any set of r nodes $\mathcal{T} \in [\mathbf{K}]^r$ has a message $M_{\mathcal{T}}^j$ that it wishes to convey to Node j , for any $j \in [\mathbf{K}] \setminus \mathcal{T}$. Each message $M_{\mathcal{T}}^j$ is uniformly distributed over a set $\{1, \dots, 2^{nR_{\mathcal{T}}^j}\}$, and a rate-tuple $(R_{\mathcal{T}}^j: \mathcal{T} \in [\mathbf{K}]^r, j \in [\mathbf{K}] \setminus \mathcal{T})$ is called achievable if there exists a sequence of encoding and decoding functions such that the probabilities of error tend to 0 in the asymptotic regime of infinite blocklengths. The SDoF is then defined as

$$\text{SDoF}(r) \triangleq \sup \overline{\lim}_{P \rightarrow \infty} \frac{\sum_{\mathcal{T} \in [\mathbf{K}]^r, j \in [\mathbf{K}] \setminus \mathcal{T}} R_{\mathcal{T}}^j(P)}{\frac{1}{2} \log P}, \quad (3.3)$$

where the supremum is over all sequences of rate tuples $\{(R_{\mathcal{T}}^j(P): \mathcal{T} \in [\mathbf{K}]^r, j \in [\mathbf{K}] \setminus \mathcal{T})\}_{P > 0}$ so that for each $P > 0$ each tuple $(R_{\mathcal{T}}^j(P): \mathcal{T} \in [\mathbf{K}]^r, j \in [\mathbf{K}] \setminus \mathcal{T})$ is achievable with power P .

We have the following lemma, which we use in this chapter:

Lemma 3.1 *For any $r \in [\mathbf{K}]$:*

$$\Delta(r) \leq \left(1 - \frac{r}{\mathbf{K}}\right) \frac{1}{\text{SDoF}(r)}. \quad (3.4)$$

Proof: We show how to construct a distributed computing scheme achieving the NDT upper bound in (3.4). Assume a sequence (in $P > 0$) of rates $(R_{\mathcal{T}}^j: \mathcal{T} \in [\mathbf{K}]^r, j \in [\mathbf{K}] \setminus \mathcal{T})$ that achieves $\text{SDoF}(r)$ and is completely symmetric with respect to indices j and sets \mathcal{T} . By the same time-sharing and relabeling arguments as described in Subsection 3.2.1 such a sequence must exist.

In the Map Phase we choose a regular file assignment. Partition the input files $\{W_1, \dots, W_N\}$ into $\binom{\mathbf{K}}{r}$ disjoint bundles and assign each bundle to a size- r subset $\mathcal{T} \in [[\mathbf{K}]]^r$. Since each file is stored at r nodes, our file assignment satisfies the constraint on the computation load.

Each node computes all IVAs associated with its stored files. During the Wireless Shuffle Phase, each transmit set \mathcal{T} communicates to any receive node $j \notin \mathcal{T}$ all IVAs that can be calculated from its bundle using the encoding and decoding functions achieving $\text{SDoF}(r)$. Since there are $N(\mathbf{K} - r)$ IVAs to be sent (one from each file to each receiver that does not store this file), the probability of error of these transmissions tends to 0 as $n \rightarrow \infty$, whenever

$$\overline{\lim}_{P \rightarrow \infty} \overline{\lim}_{A \rightarrow \infty} \frac{A \cdot N(\mathbf{K} - r)}{T \cdot \log P} \leq \text{SDoF}(r), \quad (3.5)$$

which is equivalent to

$$\overline{\lim}_{P \rightarrow \infty} \overline{\lim}_{A \rightarrow \infty} \frac{T}{A \cdot \mathbf{K} \cdot N} \cdot \log P \geq \frac{\mathbf{K} - r}{\mathbf{K}} \frac{1}{\text{SDoF}(r)}. \quad (3.6)$$

This proves the desired achievability result. ■

3.3 Main Results

The main results of this chapter are new upper and lower bounds on the computation-NDT tradeoff of the wireless DC system described in Section 3.2.

For fixed K , define for each $r \in [\lceil K/2 \rceil - 1]$:

$$\Delta_{\text{Ub},1}(r) \triangleq \left(1 - \frac{r}{K}\right) \cdot \frac{r(K-1) + K - r - 1}{r(K-1)^2 + r(K-2)}. \quad (3.7)$$

Further, for $K = 5$ and $r = 2$, define

$$\Delta_{\text{Ub},2}(r) \triangleq \left(1 - \frac{r}{K}\right) \cdot \frac{7}{30} \quad (3.8)$$

and for all odd values $K \geq 7$ and $r = (K-1)/2$, set:

$$\Delta_{\text{Ub},2}(r) \triangleq \frac{1}{K} \left(1 - \frac{r}{K}\right) \left(1 + \frac{1}{(K-r-1)(K-1)}\right). \quad (3.9)$$

For all other values of r and K , set $\Delta_{\text{Ub},2}(r) = \infty$.¹

Define for any integer value $r \in [K]$:

$$\Delta_{\text{Ub}}(r) \triangleq \begin{cases} \min_{i \in \{1,2\}} \Delta_{\text{Ub},i}(r) & \text{if } r < K/2 \\ \frac{1}{K} \left(1 - \frac{r}{K}\right) & \text{if } r \geq K/2 \end{cases}. \quad (3.10)$$

Also, let

$$\Delta_{\text{Lb}}(r) \triangleq \begin{cases} \frac{1}{K} \left(2 - \frac{3}{K}\right) & \text{if } r = 1, \\ \frac{1}{K} \left(1 - \frac{r}{K} + \max_{t \in [\lceil K/2 \rceil]} \text{lowc}(C_t(r))\right) & \text{if } r \in \left(1, \left\lceil \frac{K}{2} \right\rceil\right), \\ \frac{1}{K} \left(1 - \frac{r}{K}\right) & \text{if } r \in \left[\left\lceil \frac{K}{2} \right\rceil, K\right], \end{cases} \quad (3.11)$$

where for any $t \in [\lceil K/2 \rceil]$:

$$C_t(i) = \begin{cases} \frac{\binom{K-i}{t-i}}{\binom{K}{t}} \cdot (K-2t), & \text{if } i \in [t], \\ 0, & \text{if } i \in [K] \setminus [t], \end{cases} \quad (3.12)$$

and recall that $\text{lowc}(C_t(r))$ denotes the lower convex envelope of $\{(r, C_t(r))\}_{r=1}^K$.

Theorem 3.1 *The computation-NDT tradeoff $\Delta^*(r)$ is upper- and lower-bounded as:*

$$\Delta_{\text{Lb}}(r) \leq \Delta^*(r) \leq \text{lowc}(\Delta_{\text{Ub}}(r)). \quad (3.13)$$

¹The second upper bound is interesting and nontrivial only when $K = 2r + 1$ and $r = 2, 3, \dots$

Proof: For integers $r \geq K/2$ achievability of the upper bound $\Delta_{\text{Ub}}(r)$ is proved in [26]. For integers $r < K/2$ achievability of the two upper bounds $\Delta_{\text{Ub},1}(r)$ and $\Delta_{\text{Ub},2}(r)$ follows by Lemma 3.1 and the coding schemes described in Sections 3.5 and 3.6. (Section 3.4 illustrates simple special cases of the scheme in Section 3.5.) Achievability of the lower convex envelope follows by selecting the best strategy for each value of r and by simple time- and memory-sharing strategies. The lower bound is proved in Section 3.7. ■

Remark 3.1 *The upper bound in (3.13) is convex and piece-wise constant. The lower bound is piece-wise constant with segments spanning the intervals $[i, i+1]$, for $i = 2, \dots, K-1$. On the interval $[1, 2)$, the lower bound is constant over smaller sub-intervals only but not over the entire segment.*

For all $r \geq \lceil K/2 \rceil$ the lower bound (3.10) and the upper bound (3.11) match.

Corollary 3.1 *For all $r \geq \lceil K/2 \rceil$:*

$$\Delta^*(r) = \left(1 - \frac{r}{K}\right) \cdot \frac{1}{K}. \quad (3.14)$$

Proof: For $r \geq \lceil K/2 \rceil$ the upper bound $\text{lowc}(\Delta_{\text{Ub}}(r))$ is equal to the lower bound $\Delta_{\text{Lb}}(r)$ because $C_{\lceil K/2 \rceil}(i) = 0$ for all $i \geq \lceil K/2 \rceil$. ■

Remark 3.2 *The computation-NDT tradeoff in (3.14), is achieved with linear zero-forcing and side-information cancellation, see [26]. These simple strategies are thus sufficient to achieve the optimal computation-NDT tradeoff in the regime $r \geq \lceil K/2 \rceil$. This statement however does not apply for smaller values of r where more sophisticated strategies such as interference alignment (IA) strategies are necessary to achieve the optimal computation-NDT tradeoff. This follows from the converse result in [26] and our achievability part in Theorem 3.1, see Corollary 3.2 ahead.*

3.3.1 Comparison to Previous Upper Bounds

We compare the bound in Theorem 3.1 to the upper bounds in [26] and Theorem 2.4. The upper bound in [26] is given as follows:

$$\Delta^*(r) \leq \Delta_{\text{UB-BF}}(r) \triangleq \text{lowc} \left\{ \left(r, \frac{1 - r/K}{\min(K, 2r)} \right) : r \in [K] \right\}, \quad (3.15)$$

and is tight when restricting to zero-forcing, one-shot beamforming, and side-information cancellation. We recall that The upper bound in Theorem 2.4 has the form:

$$\Delta^*(r) \leq \Delta_{\text{Ub-Groups}}(r) \triangleq \text{lowc} \left((K, 0) \cup \left\{ \left(r, \frac{1 - r/K}{\text{SDoFLb}_r(r)} \right) : 1 \leq r < K, r|K \right\} \right), \quad (3.16)$$

where

$$\text{SDoF}_{\text{Lb}, (r)} \triangleq \begin{cases} 2r & \text{if } K/r \in \{2, 3\}, \\ \frac{K(K-r)-r^2}{2K-3r} & \text{if } K/r \geq 4, \end{cases} \quad (3.17)$$

and coincides with the upper bound in Theorem 3.1 for $r = 1$, i.e., $\Delta_{\text{Ub}}(1) = \Delta_{\text{Ub-Groups}}(1)$.

Notice that the sequences $\{\Delta_{\text{UB-BF}}(r): r = 1, \dots, \lceil \frac{K}{2} \rceil\}$ and $\{\Delta_{\text{UB-Groups}}(r): r = 1, \dots, \lceil \frac{K}{2} \rceil\}$ are strictly convex and the lower convex-envelope of these points is piece-wise linear.

Corollary 3.2 (Strict Improvement) *For all $1 < r < \lceil \frac{K}{2} \rceil$:*

$$\Delta^*(r) \leq \text{lowc}(\Delta_{\text{Ub}}(r)) < \Delta_{\text{Ub-Groups}}(r), \quad (3.18)$$

and for all $1 \leq r < \lceil \frac{K}{2} \rceil$:

$$\Delta^*(r) \leq \text{lowc}(\Delta_{\text{Ub}}(r)) < \Delta_{\text{Ub-ZF}}(r), \quad (3.19)$$

where notice the strict inequality on the two inequalities to the right.

Figures 3.1 and 3.2 show the bounds in Theorem 3.1 and compare them to the previous upper bounds $\Delta_{\text{Ub-Groups}}(r)$ and $\Delta_{\text{Ub-ZF}}(r)$.

Proof of the Corollary: For $r < \lceil \frac{K}{2} \rceil$, our upper bound in Theorem 3.1 is strictly better than the bounds in [26] and Theorem 2.4, as we argue in the following. We start by noticing that for $r = \lceil \frac{K}{2} \rceil$ all three upper bounds coincide:

$$\Delta_{\text{Ub}}\left(\left\lceil \frac{K}{2} \right\rceil\right) = \Delta_{\text{Ub-Groups}}\left(\left\lceil \frac{K}{2} \right\rceil\right) = \Delta_{\text{Ub-ZF}}\left(\left\lceil \frac{K}{2} \right\rceil\right). \quad (3.20)$$

Consider now $r < \lceil \frac{K}{2} \rceil$. We first prove the strict inequality in (3.19). Comparing the new upper bound in (3.7) with the previous upper bound in (3.15), we see that for any integer computation load r the new bound is better (smaller) than the old one if

$$\frac{r(K-1) + K - r - 1}{r(K-1)^2 + r(K-2)} < \frac{1}{2r}, \quad (3.21)$$

which is equivalent to

$$2r \leq \frac{K^2 - K - 1}{K - 2}. \quad (3.22)$$

For $K \geq 4$ and any integer computation load

$$r \leq (K-2)/2 \quad (3.23)$$

the condition (3.22) is satisfied and thus also (3.21), proving the improvement of the new bound. I.e., for K even and integer-valued $r \geq \frac{K}{2} - 1$ as well as for K odd and integer-valued $r \geq \frac{K-3}{2}$, the upper bound in Theorem 3.1 is strictly lower than the previous upper bound $\Delta_{\text{Ub-ZF}}(r)$.

We next focus on K odd and $r = \frac{K-1}{2}$. For $K = 5$ and $r = 2$ the bound (3.15) evaluates to $(1 - \frac{r}{K}) \cdot \frac{1}{4}$ and is thus strictly higher than bound (3.8), which is $(1 - \frac{r}{K}) \cdot \frac{7}{30}$. For $K \geq 7$, the new bound in (3.9) improves over the old bound in (3.15) if

$$\frac{1}{K} \left(1 + \frac{1}{(K-r-1)(K-1)} \right) < \frac{1}{2r} = \frac{1}{K-1}, \quad (3.24)$$

which is equivalent to (multiply both sides by K and $(K - 1)$):

$$K - 1 + \frac{1}{K - r - 1} < K \quad (3.25)$$

and is satisfied for all values of $r \leq K - 3$ and thus for $K \geq 7$ and $r = \frac{K-1}{2}$. We have thus shown that for all integers $r \leq \frac{K-1}{2}$ the new bound is better than (3.15). Combined with (3.20) and the piece-wise linearity of the bounds, this proves (3.19).

We continue to prove (3.18), and still focus on integers r satisfying $1 < r < \lceil \frac{K}{2} \rceil$. For these integers:

$$(r - 1)(K - 2r - 1) \geq 0, \quad (3.26)$$

which is equivalent to

$$r(2K - 3r) \geq r(K - 1) + K - r - 1. \quad (3.27)$$

Moreover, for all integers $r > 1$:

$$(K - 1)^2 + (K - 2) > K(K - r) - r^2, \quad (3.28)$$

which combined with (3.27) establishes that

$$\frac{r(K - 1) + K - r - 1}{r(K - 1)^2 + r(K - 2)} < \frac{2K - 3r}{K(K - r) - r^2}. \quad (3.29)$$

This implies for all integers r satisfying $1 < r \leq \frac{K-1}{2}$, that the new upper bound is better than the upper bound (3.16). Combined with (3.20) and the piece-wise linearity of the bounds, this proves (3.18), except when there is no integer r in the range $(1, \lceil \frac{K}{2} \rceil)$ and thus $\Delta_{\text{Ub}}(r)$ and $\Delta_{\text{Ub-Groups}}(r)$ are both given by the straight line between $\Delta_{\text{Ub-Groups}}(1)$ and $\Delta_{\text{Ub-Groups}}(\lceil \frac{K}{2} \rceil)$. In this case the statement (3.18) is void and there is nothing to prove. \blacksquare

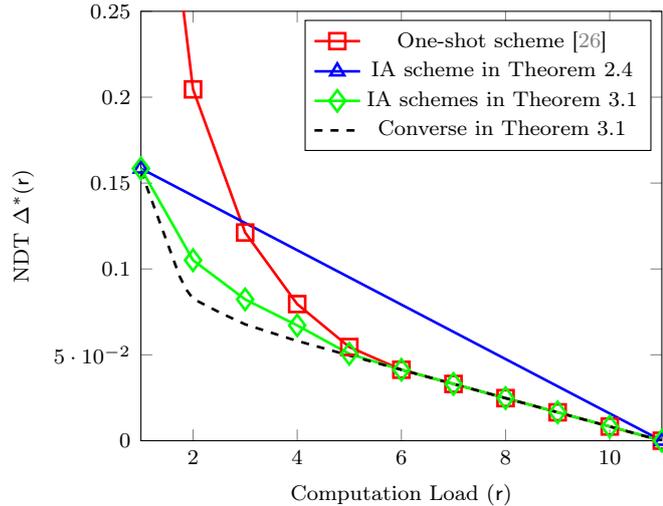


Figure 3.1: Bounds on $\Delta^*(r)$ from Theorem 3.1 compared to the optimal zero-forcing and interference cancellation scheme in [26] and to the upper bound obtained by the IA scheme in Theorem 2.4 for $K = 11$.

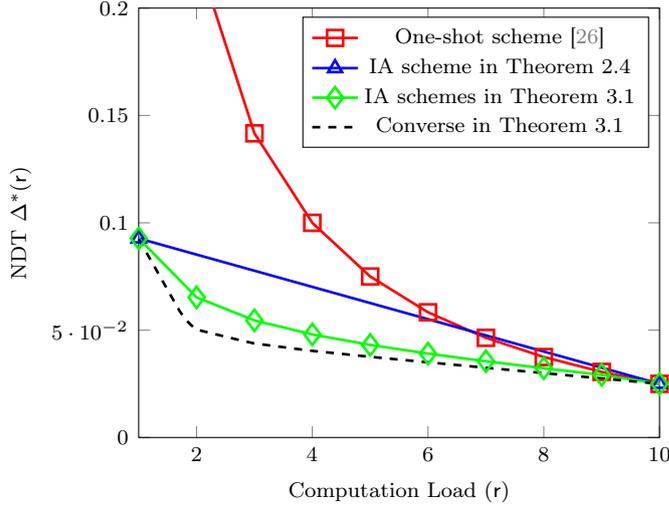


Figure 3.2: Bounds on $\Delta^*(r)$ from Theorem 3.1 compared to the optimal NDT achieved by the optimal zero-forcing and interference cancellation scheme in [26] and to the upper bound obtained by the IA scheme in Theorem 2.4 for $K = 20$.

3.4 Examples of our IA Scheme without Zero-forcing (ZF)

We describe our first coding scheme based on IA but without zero-forcing, and we evaluate the lower bound on $\text{SDoF}(r)$ that it achieves. In this section we only present some simple examples and attempt to build up intuition. The scheme and its corresponding upper bound on the SDoF are described and analyzed in detail in the next-following Section 3.5.

3.4.1 Example 1: $K \geq 3, r = 1$

Consider first the simple case with computation load $r = 1$, i.e., when each IVA can be stored only at a single node. In our scheme we transmit the $K(K-1) - 1$ Messages (or IVAs)

$$\{M_k^j : j, k \in [K], j \neq k, (j, k) \neq (1, K)\}. \quad (3.30)$$

That means, each Node k transmits a message M_k^j to each other node $j \neq k$, except for Node K that only transmits messages to Nodes $2, \dots, K-1$ but not to Node 1.

All messages $\{M_k^j\}_k$ intended for Node j , for $j = 2, \dots, K$, are precoded by the precoding matrix \mathbf{U}_j , whose construction we shall present shortly. In contrast to the standard IA-scheme in [30], here Node 1 does not have a dedicated IA precoding matrix. Instead, each Node k precodes the message M_k^1 that it sends to Node 1 with the precoding matrix \mathbf{U}_k that is typically reserved to its own intended transmissions.

Table 3.1 depicts the precoding matrices used to transmit information from a given Node k to another Node j . The entry “x” indicates that nodes do not transmit messages to themselves. The entry “o” indicates that Node 7 chooses not to send a message to Node 1. The motivation for our precoding assignment is to have no duplications in a given row (because otherwise the corresponding receive nodes won’t be able to distinguish their intended signals from interference) and to use as few precoding matrices as possible so as to keep the nodes’ interference spaces small (this will become more clear shortly).

Table 3.1: Table showing the precoding matrix used to send each message M_k^j .

$k \setminus j$	1	2	3	4	5	6	7
1	x	\mathbf{U}_2	\mathbf{U}_3	\mathbf{U}_4	\mathbf{U}_5	\mathbf{U}_6	\mathbf{U}_7
2	\mathbf{U}_2	x	\mathbf{U}_3	\mathbf{U}_4	\mathbf{U}_5	\mathbf{U}_6	\mathbf{U}_7
3	\mathbf{U}_3	\mathbf{U}_2	x	\mathbf{U}_4	\mathbf{U}_5	\mathbf{U}_6	\mathbf{U}_7
4	\mathbf{U}_4	\mathbf{U}_2	\mathbf{U}_3	x	\mathbf{U}_5	\mathbf{U}_6	\mathbf{U}_7
5	\mathbf{U}_5	\mathbf{U}_2	\mathbf{U}_3	\mathbf{U}_4	x	\mathbf{U}_6	\mathbf{U}_7
6	\mathbf{U}_6	\mathbf{U}_2	\mathbf{U}_3	\mathbf{U}_4	\mathbf{U}_5	x	\mathbf{U}_7
7	o	\mathbf{U}_2	\mathbf{U}_3	\mathbf{U}_4	x	\mathbf{U}_6	x

Node 1 thus transmits the signal

$$\mathbf{X}_1 = \sum_{j \in [\mathbf{K}] \setminus \{1\}} \mathbf{U}_j \mathbf{b}_1^j, \quad (3.31)$$

Nodes 2, ..., K - 1 transmit the signals

$$\mathbf{X}_k = \mathbf{U}_k \mathbf{b}_k^1 + \sum_{j \in [\mathbf{K}] \setminus \{1, k\}} \mathbf{U}_j \mathbf{b}_k^j, \quad k \in [\mathbf{K} - 1] \setminus \{1\}, \quad (3.32)$$

and Node K transmits the signal

$$\mathbf{X}_K = \sum_{j \in [\mathbf{K} - 1] \setminus \{1\}} \mathbf{U}_j \mathbf{b}_K^j, \quad (3.33)$$

where \mathbf{b}_k^j is a Gaussian codeword encoding Message M_k^j .

Node 1 observes the receive signal

$$\begin{aligned} \mathbf{Y}_1 = & \underbrace{\sum_{k \in [\mathbf{K} - 1] \setminus \{1\}} \mathbf{H}_{1,k} \mathbf{U}_k \mathbf{b}_k^1}_{\text{desired signal}} \\ & + \underbrace{\sum_{\substack{k, \ell \in [\mathbf{K}] \setminus \{1\} \\ k \neq \ell}} \mathbf{H}_{1,k} \mathbf{U}_\ell \mathbf{b}_k^\ell}_{\text{interference}} + \mathbf{Z}_1, \end{aligned} \quad (3.34)$$

and Nodes 2, ..., K the receive signals

$$\begin{aligned} \mathbf{Y}_j = & \underbrace{\sum_{k \in [\mathbf{K}] \setminus \{j\}} \mathbf{H}_{j,k} \mathbf{U}_k \mathbf{b}_k^j}_{\text{desired signal}} + \underbrace{\sum_{\substack{\ell, k \in [\mathbf{K}] \setminus \{j\} \\ \ell \neq k, \ell \neq 1}} \mathbf{H}_{j,k} \mathbf{U}_\ell \mathbf{b}_k^\ell}_{\text{interference}} \\ & + \underbrace{\sum_{k \in [\mathbf{K} - 1] \setminus \{1, j\}} \mathbf{H}_{j,k} \mathbf{U}_k \mathbf{b}_k^1}_{\text{interference}} + \mathbf{Z}_j, \end{aligned} \quad (3.35)$$

where $\mathbf{H}_{j,k}$ denotes the diagonal T-by-T channel matrix consisting of the entries $\{H_{j,k}(t)\}_{t=1}^T$.

Remark 3.3 *The $K - 1$ desired signals at any Node $j \in [K] \setminus \{1\}$ are precoded by the same precoding matrix \mathbf{U}_j , while its interference signals are precoded by the remaining $K - 2$ precoding matrices $\mathbf{U}_2, \dots, \mathbf{U}_{j-1}, \mathbf{U}_{j+1}, \dots, \mathbf{U}_K$. In contrast, for Node 1, matrices $\mathbf{U}_2, \dots, \mathbf{U}_{K-1}$ precode both desired and interference signals while matrix \mathbf{U}_K only precodes interference.*

We choose the precoding matrices $\mathbf{U}_2, \dots, \mathbf{U}_K$ according to the IA principle [30]. That means, we construct each column of matrix \mathbf{U}_j using all channel matrices that pre-multiply \mathbf{U}_j in the interference terms of the receive signals $\mathbf{Y}_1, \dots, \mathbf{Y}_K$ and exponentiate these channel matrices with a different set of exponents for each column. More formally, we choose²

$$\mathbf{U}_\ell \triangleq \left[\prod_{\mathbf{H} \in \mathcal{H}_\ell} \mathbf{H}^{\alpha_{\ell, \mathbf{H}}} \cdot \boldsymbol{\Xi}_\ell : \forall \alpha_\ell \in [\eta]^\Gamma \right], \quad (3.36)$$

where each column of the matrix is constructed using a different exponent-vector $\alpha_\ell = (\alpha_{\ell, \mathbf{H}} : \mathbf{H} \in \mathcal{H}_\ell) \in [\eta]^\Gamma$; η is a large number depending on the blocklength T that tends to ∞ with T ; $\{\boldsymbol{\Xi}_\ell\}_{\ell \in [K] \setminus \{1\}}$ are i.i.d. random vectors independent of all channel matrices, noises, and messages; and

$$\begin{aligned} \mathcal{H}_\ell &= \{\mathbf{H}_{j,k} : j \in [K] \setminus \{1, \ell\}, k \in [K] \setminus \{\ell\}, j \neq k\} \\ &\cup \{\mathbf{H}_{1,k} : k \in [K] \setminus \{1, \ell\}\}, \end{aligned} \quad (3.37)$$

and $\Gamma \triangleq |\mathcal{H}_\ell|$ does not depend on ℓ .

With the proposed construction, for any $j \in [K]$ and $\ell \in [K] \setminus \{1, j\}$, the signals that are precoded by matrix \mathbf{U}_ℓ and interfere at Node j lie in the column space of the matrix

$$\mathbf{W}_\ell \triangleq \left[\prod_{\mathbf{H} \in \mathcal{H}_\ell} \mathbf{H}^{\alpha_{\ell, \mathbf{H}}} \cdot \boldsymbol{\Xi}_\ell : \forall \alpha_\ell \in [\eta + 1]^\Gamma \right]. \quad (3.38)$$

The signals that are desired at Node $j \in \{2, \dots, K\}$ lies in the subspace spanned by the columns of the matrix

$$\mathbf{D}_j \triangleq \left[\mathbf{H}_{j,k} \mathbf{U}_j \right]_{k \in [K] \setminus \{j\}}, \quad j \in \{2, \dots, K\}. \quad (3.39)$$

The signals desired at Node 1 lies in the subspace spanned by the column space of the matrix

$$\mathbf{D}_1 \triangleq \left[\mathbf{H}_{1,k} \mathbf{U}_k \right]_{k \in [K-1] \setminus \{1\}}. \quad (3.40)$$

As is proved in Section 2.6.2 and follows from our analysis in Section 3.5, with probability 1 (over the random channel matrices) the matrices

$$\boldsymbol{\Lambda}_j = [\mathbf{D}_j \mathbf{W}_2 \cdots \mathbf{W}_{j-1} \mathbf{W}_{j+1} \cdots \mathbf{W}_K], \quad j \in [K] \setminus \{1\}, \quad (3.41)$$

²By the memorylessness of the channel the matrices $\mathbf{H}_{j,k}$ are diagonal and their multiplications and exponentiations are effectively multiplications and exponentiations of the corresponding diagonal elements.

and

$$\mathbf{\Lambda}_1 = [\mathbf{D}_1 \mathbf{W}_2 \cdots \mathbf{W}_K] \quad (3.42)$$

have full column-rank.

Since the matrices have full column-rank, a simple zero-forcing strategy at the receiving nodes allows to achieve DoF

$$\frac{\#\text{columns}(\mathbf{D}_j)}{\#\text{columns}(\mathbf{\Lambda}_j)} \quad (3.43)$$

to each Node j . I.e., in the limit as $\eta \rightarrow \infty$ (and thus $\frac{\eta}{\eta+1} \rightarrow 1$) a DoF $\frac{K-1}{2K-3}$ at Nodes $2, \dots, K$ and DoF $\frac{K-2}{2K-3}$ at Node 1 is achievable, yielding a SDoF of

$$\text{SDoF}_{\text{LB}} = \frac{(K-1)^2 + K - 2}{2K - 3}. \quad (3.44)$$

3.4.2 Example 2: $K = 4, r = 2$

Consider now a computation load of $r = 2$ and only $K = 4$ nodes. (For this set of parameters our scheme is simple to present. Other schemes however can perform better.)

For $r = 2$ and $K = 4$ our scheme transmits 22 different messages depicted in (3.45) ahead. Here, Message $M_{k,\mathcal{T}}^j$ is a message that is known by the set of nodes \mathcal{T} and intended to Node $j \notin \mathcal{T}$. Though known to the entire set \mathcal{T} , Message $M_{k,\mathcal{T}}^j$ is only transmitted by a single Node $k \in \mathcal{T}$. The remaining nodes in $\mathcal{T} \setminus \{k\}$ do not participate in the transmission. They however exploit their knowledge of $M_{k,\mathcal{T}}^j$ to cancel the corresponding transmission from their receive signal.

Notice that for certain sets \mathcal{T} and receive nodes $j \notin \mathcal{T}$ our scheme sends two messages to the same node j : $M_{k_1,\mathcal{T}}^j$ and $M_{k_2,\mathcal{T}}^j$ for different indices $k_1, k_2 \in \mathcal{T}$. (In (3.45) the two messages $M_{1,\{1,4\}}^2$ and $M_{4,\{1,4\}}^2$ for example have this form.) These messages $M_{k_1,\mathcal{T}}^j$ and $M_{k_2,\mathcal{T}}^j$ actually represent two independent submessages of Message $M_{\mathcal{T}}^j$ as we defined it in Section 3.2.2. For the sets \mathcal{T} and Nodes $j \notin \mathcal{T}$ for which there exists only a single Message $M_{k,\mathcal{T}}^j$, this message is really the message $M_{\mathcal{T}}^j$. As we will only analyze the SDoF of our scheme, this distinction between submessages and messages is not important and we shall simply omit it in the following.

We send the following messages in our scheme. To Node 1, we send messages

$$M_{2,\{2,3\}}^1, M_{3,\{2,3\}}^1, M_{2,\{2,4\}}^1, M_{3,\{3,4\}}^1; \quad (3.45a)$$

to Node 2 we send messages

$$\begin{aligned} M_{1,\{1,3\}}^2, M_{3,\{1,3\}}^2, M_{1,\{1,4\}}^2, M_{4,\{1,4\}}^2 \\ M_{3,\{3,4\}}^2, M_{4,\{3,4\}}^2; \end{aligned} \quad (3.45b)$$

to Node 3 we send messages

$$\begin{aligned} M_{1,\{1,2\}}^3, M_{2,\{1,2\}}^3, M_{1,\{1,4\}}^3, M_{4,\{1,4\}}^3, \\ M_{2,\{2,4\}}^3, M_{4,\{2,4\}}^3; \end{aligned} \quad (3.45c)$$

and to Node 4 we send messages

$$\begin{aligned} M_{1,\{1,2\}}^4, M_{2,\{1,2\}}^4, M_{1,\{1,3\}}^4, M_{3,\{1,3\}}^4, \\ M_{2,\{2,3\}}^4, M_{3,\{2,3\}}^4. \end{aligned} \quad (3.45d)$$

We observe that each Node j obtains two messages from each subset of nodes \mathcal{T} of size 2 not containing j , where each node in \mathcal{T} sends one of the $|\mathcal{T}| = 2$ messages. An exception are messages $M_{4,\{2,4\}}^1$ and $M_{4,\{3,4\}}^1$ which are not transmitted because in our scheme the last Node $K = 4$ does not send any message to the first Node 1. (As we will see, this omission allows to reuse some of the precoding matrices, similarly to the scheme for $r = 1$, and thus achieve an improved SDoF.) Prior to transmission, each message $M_{k,\mathcal{T}}^j$ is encoded into a Gaussian codeword $\mathbf{b}_{k,\mathcal{T}}^j$. We use the interference-alignment (IA) technique with three precoding matrices $\mathbf{U}_{\{2,3\}}$, $\mathbf{U}_{\{2,4\}}$, and $\mathbf{U}_{\{3,4\}}$. Precoding matrix $\mathbf{U}_{\{2,3\}}$ is used to send codewords

$$\mathbf{b}_{1,\{1,3\}}^2, \mathbf{b}_{1,\{1,2\}}^3, \mathbf{b}_{4,\{3,4\}}^2, \mathbf{b}_{4,\{2,4\}}^3, \quad (3.46)$$

$$\mathbf{b}_{2,\{2,3\}}^1, \mathbf{b}_{3,\{2,3\}}^1, \mathbf{b}_{3,\{1,3\}}^2, \mathbf{b}_{2,\{1,2\}}^3, \quad (3.47)$$

precoding matrix $\mathbf{U}_{\{2,4\}}$ is used to send codewords

$$\mathbf{b}_{1,\{1,4\}}^2, \mathbf{b}_{1,\{1,2\}}^4, \mathbf{b}_{3,\{3,4\}}^2, \mathbf{b}_{3,\{2,3\}}^4, \quad (3.48)$$

$$\mathbf{b}_{2,\{2,4\}}^1, \mathbf{b}_{2,\{1,2\}}^4, \mathbf{b}_{4,\{1,4\}}^2, \quad (3.49)$$

and precoding matrix $\mathbf{U}_{\{3,4\}}$ is used to send codewords

$$\mathbf{b}_{1,\{1,4\}}^3, \mathbf{b}_{1,\{1,3\}}^4, \mathbf{b}_{2,\{2,4\}}^3, \mathbf{b}_{2,\{2,3\}}^4, \quad (3.50)$$

$$\mathbf{b}_{3,\{3,4\}}^1, \mathbf{b}_{4,\{1,4\}}^3, \mathbf{b}_{3,\{1,3\}}^4. \quad (3.51)$$

Remark 3.4 *The choice of precoding matrices is inspired by [28] where Message $M_{k,\mathcal{T}}^j$ is precoded by the matrix $\mathbf{U}_{\mathcal{R}}$ for $\mathcal{R} = \mathcal{T} \setminus \{k\} \cup \{j\}$. The idea behind the choice of precoding matrices in [28] is that any node in \mathcal{R} is either interested in learning Message $M_{k,\mathcal{T}}^j$ or it can compute it itself and remove the interference from its receive signal. A given node j thus only experiences interference from precoding matrices $\mathbf{U}_{\mathcal{R}}$ for which $j \notin \mathcal{R}$.*

In contrast to [28], in our IA scheme we do not use precoding matrices $\mathbf{U}_{\mathcal{R}'}$ for sets \mathcal{R}' containing index 1, but reuse precoding matrices $\mathbf{U}_{\mathcal{R}}$ for sets \mathcal{R} not containing 1. Specifically, we use the precoding matrix $\mathbf{U}_{\mathcal{R}}$ also to send the codewords (if they exist)

$$\mathbf{b}_{k,\mathcal{R}}^1, \quad \mathbf{b}_{k,\mathcal{R} \cup \{1\} \setminus \{j\}}^j, \quad \forall j, k \in \mathcal{R}, j \neq k. \quad (3.52)$$

One can verify that the codewords in lines (3.47), (3.49), (3.51) are of the form in (3.52).

We illustrate our assignment of the precoding matrices also using the following table. The entries in column 1 or in rows $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$ correspond to two submessages $M_{k_1,\mathcal{T}}^j$ and $M_{k_2,\mathcal{T}}^j$, where k_1 and k_2 denote the two entries in \mathcal{T} . For all other entries in Table 3.2 not equal to "x", we have only one message per precoding matrix, see (3.46), (3.48), and (3.50).

Table 3.2: Messages $M_{k,\mathcal{T}}^j$ precoded by the three precoding matrices $\mathbf{U}_{\{2,3\}}$, $\mathbf{U}_{\{2,4\}}$, and $\mathbf{U}_{\{3,4\}}$.

$\mathcal{T} \setminus j$	1	2	3	4
$\{1, 2\}$	x	x	$\mathbf{U}_{\{2,3\}}$	$\mathbf{U}_{\{2,4\}}$
$\{1, 3\}$	x	$\mathbf{U}_{\{2,3\}}$	x	$\mathbf{U}_{\{3,4\}}$
$\{1, 4\}$	x	$\mathbf{U}_{\{2,4\}}$	$\mathbf{U}_{\{3,4\}}$	x
$\{2, 3\}$	$\mathbf{U}_{\{2,3\}}$	x	x	$\mathbf{U}_{\{2,4\}}, \mathbf{U}_{\{3,4\}}$
$\{2, 4\}$	$\mathbf{U}_{\{2,4\}}$	x	$\mathbf{U}_{\{2,3\}}, \mathbf{U}_{\{3,4\}}$	x
$\{3, 4\}$	$\mathbf{U}_{\{3,4\}}$	$\mathbf{U}_{\{2,3\}}, \mathbf{U}_{\{2,4\}}$	x	x

During the shuffling phase, Nodes 1–4 send the following signals. Node 1 sends:

$$\begin{aligned}
 \mathbf{X}_1 = & \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{1,\{1,3\}}^2 + \mathbf{b}_{1,\{1,2\}}^3 \right) \\
 & + \mathbf{U}_{\{2,4\}} \left(\mathbf{b}_{1,\{1,4\}}^2 + \mathbf{b}_{1,\{1,2\}}^4 \right) \\
 & + \mathbf{U}_{\{3,4\}} \left(\mathbf{b}_{1,\{1,4\}}^3 + \mathbf{b}_{1,\{1,3\}}^4 \right). \tag{3.53}
 \end{aligned}$$

Node 2 sends:

$$\begin{aligned}
 \mathbf{X}_2 = & \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{2,\{2,3\}}^1 + \mathbf{b}_{2,\{1,2\}}^3 \right) \\
 & + \mathbf{U}_{\{2,4\}} \left(\mathbf{b}_{2,\{2,4\}}^1 + \mathbf{b}_{2,\{1,2\}}^4 \right) \\
 & + \mathbf{U}_{\{3,4\}} \left(\mathbf{b}_{2,\{2,4\}}^3 + \mathbf{b}_{2,\{2,3\}}^4 \right). \tag{3.54}
 \end{aligned}$$

Node 3 sends:

$$\begin{aligned}
 \mathbf{X}_3 = & \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{3,\{2,3\}}^1 + \mathbf{b}_{3,\{1,3\}}^2 \right) \\
 & + \mathbf{U}_{\{2,4\}} \left(\mathbf{b}_{3,\{3,4\}}^2 + \mathbf{b}_{3,\{2,3\}}^4 \right) \\
 & + \mathbf{U}_{\{3,4\}} \left(\mathbf{b}_{3,\{3,4\}}^1 + \mathbf{b}_{3,\{1,3\}}^4 \right). \tag{3.55}
 \end{aligned}$$

Node 4 sends:

$$\begin{aligned}
 \mathbf{X}_4 = & \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{4,\{3,4\}}^2 + \mathbf{b}_{4,\{2,4\}}^3 \right) \\
 & + \mathbf{U}_{\{2,4\}} \mathbf{b}_{4,\{1,4\}}^2 + \mathbf{U}_{\{3,4\}} \mathbf{b}_{4,\{1,4\}}^3. \tag{3.56}
 \end{aligned}$$

As mentioned, each receiving node can subtract all the interference of the signals that it can compute itself. We can rewrite the four receive signals after this interference elimination

step as follows. Node 1 can construct:

$$\begin{aligned}
 \mathbf{Y}'_1 = & \underbrace{\mathbf{H}_{1,2}\mathbf{U}_{\{2,3\}}\mathbf{b}_{2,\{2,3\}}^1 + \mathbf{H}_{1,3}\mathbf{U}_{\{2,3\}}\mathbf{b}_{3,\{2,3\}}^1}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{1,2}\mathbf{U}_{\{2,4\}}\mathbf{b}_{2,\{2,4\}}^1 + \mathbf{H}_{1,3}\mathbf{U}_{\{3,4\}}\mathbf{b}_{3,\{3,4\}}^1}_{\text{desired signal}} \\
 & + \mathbf{H}_{1,2}\mathbf{U}_{\{3,4\}} \left(\mathbf{b}_{2,\{2,4\}}^3 + \mathbf{b}_{2,\{2,3\}}^4 \right) \\
 & + \mathbf{H}_{1,3}\mathbf{U}_{\{2,4\}} \left(\mathbf{b}_{3,\{3,4\}}^2 + \mathbf{b}_{3,\{2,3\}}^4 \right) \\
 & + \mathbf{H}_{1,4}\mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{4,\{3,4\}}^2 + \mathbf{b}_{4,\{2,4\}}^3 \right) + \mathbf{Z}_1, \tag{3.57}
 \end{aligned}$$

Node 2 can construct:

$$\begin{aligned}
 \mathbf{Y}'_2 = & \underbrace{\mathbf{H}_{2,1}\mathbf{U}_{\{2,3\}}\mathbf{b}_{1,\{1,3\}}^2 + \mathbf{H}_{2,1}\mathbf{U}_{\{2,4\}}\mathbf{b}_{1,\{1,4\}}^2}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{2,3}\mathbf{U}_{\{2,3\}}\mathbf{b}_{3,\{1,3\}}^2 + \mathbf{H}_{2,3}\mathbf{U}_{\{2,4\}}\mathbf{b}_{3,\{3,4\}}^2}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{2,4}\mathbf{U}_{\{2,3\}}\mathbf{b}_{4,\{3,4\}}^2 + \mathbf{H}_{2,4}\mathbf{U}_{\{2,4\}}\mathbf{b}_{4,\{1,4\}}^2}_{\text{desired signal}} \\
 & + \mathbf{H}_{2,1}\mathbf{U}_{\{3,4\}} \left(\mathbf{b}_{1,\{1,4\}}^3 + \mathbf{b}_{1,\{1,3\}}^4 \right) \\
 & + \mathbf{H}_{2,3}\mathbf{U}_{\{3,4\}} \left(\mathbf{b}_{3,\{3,4\}}^1 + \mathbf{b}_{3,\{1,3\}}^4 \right) \\
 & + \mathbf{H}_{2,4}\mathbf{U}_{\{3,4\}}\mathbf{b}_{4,\{1,4\}}^3 + \mathbf{Z}_2, \tag{3.58}
 \end{aligned}$$

Node 3 can construct:

$$\begin{aligned}
 \mathbf{Y}'_3 = & \underbrace{\mathbf{H}_{3,1}\mathbf{U}_{\{2,3\}}\mathbf{b}_{1,\{1,2\}}^3 + \mathbf{H}_{3,1}\mathbf{U}_{\{3,4\}}\mathbf{b}_{1,\{1,4\}}^3}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{3,2}\mathbf{U}_{\{2,3\}}\mathbf{b}_{2,\{1,2\}}^3 + \mathbf{H}_{3,2}\mathbf{U}_{\{3,4\}}\mathbf{b}_{2,\{2,4\}}^3}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{3,4}\mathbf{U}_{\{2,3\}}\mathbf{b}_{4,\{3,4\}}^3 + \mathbf{H}_{3,2}\mathbf{U}_{\{3,4\}}\mathbf{b}_{4,\{1,4\}}^3}_{\text{desired signal}} \\
 & + \mathbf{H}_{3,1}\mathbf{U}_{\{2,4\}} \left(\mathbf{b}_{1,\{1,4\}}^2 + \mathbf{b}_{1,\{1,2\}}^4 \right) \\
 & + \mathbf{H}_{3,2}\mathbf{U}_{\{2,4\}} \left(\mathbf{b}_{2,\{2,4\}}^1 + \mathbf{b}_{2,\{1,2\}}^4 \right) \\
 & + \mathbf{H}_{3,4}\mathbf{U}_{\{2,4\}}\mathbf{b}_{4,\{1,4\}}^2 + \mathbf{Z}_3, \tag{3.59}
 \end{aligned}$$

Node 4 can construct:

$$\begin{aligned}
 \mathbf{Y}'_4 = & \underbrace{\mathbf{H}_{4,1} \mathbf{U}_{\{2,4\}} \mathbf{b}_{1,\{1,2\}}^4 + \mathbf{H}_{4,1} \mathbf{U}_{\{2,4\}} \mathbf{b}_{1,\{1,3\}}^4}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{4,2} \mathbf{U}_{\{2,4\}} \mathbf{b}_{2,\{1,2\}}^4 + \mathbf{H}_{4,2} \mathbf{U}_{\{3,4\}} \mathbf{b}_{2,\{2,3\}}^4}_{\text{desired signal}} \\
 & + \underbrace{\mathbf{H}_{4,3} \mathbf{U}_{\{2,4\}} \mathbf{b}_{3,\{2,3\}}^4 + \mathbf{H}_{4,3} \mathbf{U}_{\{3,4\}} \mathbf{b}_{3,\{1,3\}}^4}_{\text{desired signal}} \\
 & + \mathbf{H}_{4,1} \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{1,\{1,3\}}^2 + \mathbf{b}_{1,\{1,2\}}^3 \right) \\
 & + \mathbf{H}_{4,2} \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{2,\{2,3\}}^1 + \mathbf{b}_{2,\{1,2\}}^3 \right) \\
 & + \mathbf{H}_{4,3} \mathbf{U}_{\{2,3\}} \left(\mathbf{b}_{3,\{2,3\}}^1 + \mathbf{b}_{3,\{1,3\}}^2 \right) + \mathbf{Z}_4. \tag{3.60}
 \end{aligned}$$

Remark 3.5 We remark that Node 2's desired signals are all precoded by precoding matrices $\mathbf{U}_{\{2,3\}}$ and $\mathbf{U}_{\{2,4\}}$ while all interference signals are precoded by matrix $\mathbf{U}_{\{3,4\}}$. Similar observations hold for Nodes 3 and 4. Node 1 instead observes desired and interference signals precoded by all three precoding matrices.

In more general terms, each node $j \in [\mathbf{K}] \setminus \{1\}$, observes desired signals multiplied by the precoding matrices $\{\mathbf{U}_{\mathcal{R}}\}_{j \in \mathcal{R}}$ and interference signals multiplied by the precoding matrices $\{\mathbf{U}_{\mathcal{R}}\}_{j \notin \mathcal{R}}$. For Node 1, both desired and interference signals are multiplied by all possible precoding matrices.

The IA matrices $\mathbf{U}_{\{2,3\}}$, $\mathbf{U}_{\{2,4\}}$, and $\mathbf{U}_{\{3,4\}}$ are constructed based on the interference alignment idea in [30] taking into account the channel matrices that premultiply the IA matrices in the interference signals of (3.57)–(3.60). Specifically, we choose

$$\mathbf{U}_{\mathcal{R}} \triangleq \left[\prod_{\mathbf{H} \in \mathcal{H}_{\mathcal{R}}} \mathbf{H}^{\alpha_{\mathcal{R},\mathbf{H}}} \cdot \boldsymbol{\Xi}_{\mathcal{R}} : \forall \alpha_{\mathcal{R}} \in [\eta]^4 \right], \tag{3.61}$$

where each column of the matrix is constructed using a different exponent-vector $\alpha_{\mathcal{R}} = (\alpha_{\mathcal{R},\mathbf{H}} : \mathbf{H} \in \mathcal{H}_{\mathcal{R}}) \in [\eta]^4$; η is a large number depending on the blocklength T that tends to ∞ with T ; $\boldsymbol{\Xi}_{\mathcal{R}}$ are i.i.d. random vectors drawn according to a continuous distribution, and

$$\mathcal{H}_{\{2,3\}} \triangleq \{\mathbf{H}_{1,4}, \mathbf{H}_{4,1}, \mathbf{H}_{4,2}, \mathbf{H}_{4,3}\}, \tag{3.62}$$

$$\mathcal{H}_{\{2,4\}} \triangleq \{\mathbf{H}_{1,3}, \mathbf{H}_{3,1}, \mathbf{H}_{3,2}, \mathbf{H}_{3,4}\}, \tag{3.63}$$

$$\mathcal{H}_{\{3,4\}} \triangleq \{\mathbf{H}_{1,2}, \mathbf{H}_{2,1}, \mathbf{H}_{2,3}, \mathbf{H}_{2,4}\}. \tag{3.64}$$

By this choice of the precoding matrices, all interference signals at a Node 2 will lie in the columnspan of the matrix

$$\mathbf{W}_{\{3,4\}} \triangleq \left[\prod_{\mathbf{H} \in \mathcal{H}_{\{3,4\}}} \mathbf{H}^{\alpha_{\mathcal{R},\mathbf{H}}} \cdot \boldsymbol{\Xi}_{\{3,4\}} : \forall \alpha_{\mathcal{R}} \in [\eta + 1]^4 \right], \tag{3.65}$$

while the desired signals will be separable from each other and from this interference space. As we show in the following Section 3.5. The DoF achieved to this Node 2 is thus

$$\frac{6}{7}. \quad (3.66)$$

Similar observations hold for Nodes 3 and 4. Node 1 has a larger interference space and smaller desired signal space and only achieves a DoF of 4/7. The SumDoF achieved by the scheme is thus

$$\text{SDoF} = 22/7. \quad (3.67)$$

3.5 The General IA-Scheme without ZF

We fix a large parameter $\eta \in \mathbb{Z}^+$ (which we shall let tend to ∞) and let

$$\Gamma \triangleq \mathsf{K} \cdot (\mathsf{K} - r - 1) \quad (3.68)$$

$$\mathsf{T} \triangleq (\mathsf{K} - 2) \cdot \binom{\mathsf{K} - 2}{r - 1} \cdot \eta^\Gamma + \binom{\mathsf{K} - 1}{r} \cdot (\eta + 1)^\Gamma. \quad (3.69)$$

The time of transmission T thus tends to ∞ as $\eta \rightarrow \infty$ by (3.69). The parameters Γ and η are used in our construction of the precoding matrices, as will become clear in the following.

In our scheme, we send the following messages to any Node $j \in [\mathsf{K}] \setminus \{1\}$:

$$\left\{ M_{k,\mathcal{T}}^j : \mathcal{T} \in [[\mathsf{K}] \setminus \{j\}]^r, k \in \mathcal{T} \right\} \quad (3.70)$$

and to Node 1 we send messages

$$\left\{ M_{k,\mathcal{T}}^1 : \mathcal{T} \in [[\mathsf{K}] \setminus \{1\}]^r, k \in \mathcal{T} \setminus \{\mathsf{K}\} \right\}. \quad (3.71)$$

Thus, as in the examples of the previous section, the last node K does not send any message to the first node 1.

For each message, construct a Gaussian codebook of power $\mathsf{P}/\binom{\mathsf{K}-1}{r}$ and length η^Γ to encode each Message $M_{k,\mathcal{T}}^j$ into a codeword $\mathbf{b}_{k,\mathcal{T}}^j$. As in the previous sections, we shall use a linear precoding scheme, and thus Node $i \in [\mathsf{K}]$ can mitigate the interference caused by the codewords

$$\left\{ \mathbf{b}_{k,\mathcal{T}}^j \right\}_{\forall \mathcal{T}: i \in \mathcal{T}}. \quad (3.72)$$

As a consequence, for each set $\mathcal{R} \in [[\mathsf{K}]]^r$, without causing non-desired interference to nodes in \mathcal{R} , we can use the same precoding matrix $\mathbf{U}_{\mathcal{R}}$ (whose choice we describe later) for all the codewords:

$$\left\{ \mathbf{b}_{k,\mathcal{R} \cup \{k\} \setminus \{j\}}^j \right\}_{\substack{k \in [\mathsf{K}] \setminus \mathcal{R} \\ j \in \mathcal{R}}}. \quad (3.73)$$

This idea was already used in the related works [25], [28]. In contrast to these previous works, here we do not introduce the precoding matrices $\mathbf{U}_{\mathcal{R}}$ for sets \mathcal{R} containing 1. Instead, for any \mathcal{R} not containing 1 and any $k \in \mathcal{R}$, we use the matrix $\mathbf{U}_{\mathcal{R}}$ also to precode the set of codewords

$$\left\{ \mathbf{b}_{k,\mathcal{R} \cup \{1\} \setminus \{j\}}^j \right\}_{j,k \in \mathcal{R}, j \neq k} \cup \left\{ \mathbf{b}_{1,\mathcal{R} \cup \{1\} \setminus \{j\}}^j \right\}_{j \in \mathcal{R}}, \quad (3.74)$$

Table 3.3: Let $r = 2$. The table illustrates the codesymbols $\mathbf{b}_{k,\mathcal{T}}^j$ that are premultiplied by the precoding matrix $\mathbf{U}_{\{2,3\}}$. Entries for sets \mathcal{T} either equal to $\{2,3\}$ or containing 1, correspond to r transmitted codewords, one from each node in \mathcal{T} . All other entries correspond only to a single codeword from the node not in $\{2,3\}$.

$\mathcal{T} \setminus j$	1	2	3
$\{1, 2\}$	x	x	$\mathbf{U}_{\{2,3\}}$
$\{1, 3\}$	x	$\mathbf{U}_{\{2,3\}}$	x
$\{2, 3\}$	$\mathbf{U}_{\{2,3\}}$	x	x
$\{2, 4\}$	o	x	$\mathbf{U}_{\{2,3\}}$
$\{2, 5\}$	o	x	$\mathbf{U}_{\{2,3\}}$
\vdots	\vdots	\vdots	\vdots
$\{2, K\}$	o	x	$\mathbf{U}_{\{2,3\}}$
$\{3, 4\}$	o	$\mathbf{U}_{\{2,3\}}$	x
$\{3, 5\}$	o	$\mathbf{U}_{\{2,3\}}$	x
\vdots	o	\vdots	x
$\{3, K\}$	o	$\mathbf{U}_{\{2,3\}}$	x
$\{4, 5\}$	o	o	o
\vdots	o	\vdots	o
$\{K-1, K\}$	o	o	o

and

$$\{\mathbf{b}_{k,\mathcal{R}}^1\}_{k \in \mathcal{R} \setminus \{K\}}. \quad (3.75)$$

All non-intended nodes in \mathcal{R} can subtract these interferences from their receive signals because they know the codewords. This trick allows us to reduce the dimension of the interference space and thus improve performance.

In Table 3.3 we illustrate which codewords $\mathbf{b}_{k,\mathcal{T}}^j$ are premultiplied by the precoding matrix $\mathbf{U}_{\{2,3\}}$ when $r = 2$ and $K > 3$. The entry "o" indicates that a given Node k chooses not to send a message to a given Node j or the message is premultiplied by other matrices. According to (3.73) the entry in column-3 and row- $\{k, 2\}$, for each $k \in \{1, 4, \dots, K\}$, corresponds to the codeword $\mathbf{b}_{k,\{k,2\}}^3$, and the entry in column-2 and row- $\{k, 3\}$, for each $k \in \{1, 4, \dots, K\}$, corresponds to codeword $\mathbf{b}_{k,\{k,3\}}^2$. According to (3.74), the entries in rows \mathcal{T} containing index 1 correspond to the r codewords $\{\mathbf{b}_{k,\mathcal{T}}^j\}_{k \in \mathcal{T}}$. And finally, according to (3.75), the entry in column-1 and row- $\{2, 3\}$ corresponds to the two codewords $\mathbf{b}_{2,\{2,3\}}^1$ and $\mathbf{b}_{3,\{2,3\}}^1$. We thus conclude that all entries of the table in rows \mathcal{T} containing index 1 and in row $\{2, 3\}$ correspond to $r = 2$ different codewords, while all other entries correspond to only a single codeword. Similar tables can be drawn for all pairs $(k_1, k_2) \in [K]$, where recall however that node K does not send any information to node 1.

Similarly, Table 3.4 illustrates which codesymbols $\mathbf{b}_{k,\mathcal{T}}^j$ are premultiplied by the precoding matrix $\mathbf{U}_{\{2,3,4\}}$, when $r = 3$ and $K \geq 5$. The entries in rows \mathcal{T} containing index

Table 3.4: Let $r = 3$. The table illustrates the codesymbols $\mathbf{b}_{k,\mathcal{T}}^j$ that are premultiplied by the precoding matrix $\mathbf{U}_{\{2,3,4\}}$. Entries for sets \mathcal{T} either equal to $\{2,3,4\}$ or containing 1, correspond to r transmitted codewords, one from each node in \mathcal{T} . All other entries correspond only to a single codeword from the node not in $\{2,3,4\}$.

$\mathcal{T} \setminus j$	1	2	3	4
$\{1,2,3\}$	x	x	x	$\mathbf{U}_{\{2,3,4\}}$
$\{1,2,4\}$	x	x	$\mathbf{U}_{\{2,3,4\}}$	x
$\{1,3,4\}$	x	$\mathbf{U}_{\{2,3,4\}}$	x	x
$\{2,3,4\}$	$\mathbf{U}_{\{2,3,4\}}$	x	x	x
$\{2,3,5\}$	o	x	x	$\mathbf{U}_{\{2,3,4\}}$
\vdots	\vdots	\vdots	\vdots	\vdots
$\{2,3,K\}$	o	x	x	$\mathbf{U}_{\{2,3,4\}}$
$\{2,3,5\}$	o	x	$\mathbf{U}_{\{2,3,4\}}$	x
\vdots	\vdots	\vdots	\vdots	\vdots
$\{2,3,K\}$	o	x	$\mathbf{U}_{\{2,3,4\}}$	x
$\{3,4,5\}$	o	$\mathbf{U}_{\{2,3,4\}}$	x	x
\vdots	\vdots	\vdots	\vdots	\vdots
$\{3,4,K\}$	o	$\mathbf{U}_{\{2,3,4\}}$	x	x

1 correspond to $r = 3$ different codewords $\mathbf{b}_{k,\mathcal{T}}^j$, one for each $k \in \mathcal{T}$, see (3.74). Similarly, the entry in column-1 and row $\{2,3,4\}$ corresponds to the r codewords $\mathbf{b}_{k,\{2,3,4\}}^1$, for each $k \in \{2,3,4\}$. Any other entry of the table showing $\mathbf{U}_{\{2,3,4\}}$ corresponds to a single codeword $\mathbf{b}_{k,\mathcal{T}}^j$, where k is the single element in $\mathcal{T} \setminus \{2,3,4\}$.

We now describe encodings and decodings.

Encoding: Define the T -length vector of channel inputs $\mathbf{X}_k \triangleq (X_k(1), \dots, X_k(T))^T$ for each Node k . Nodes $1, \dots, K$ form the channel inputs as:

$$\mathbf{X}_1 = \sum_{\mathcal{R} \in [[K] \setminus \{1\}]^r} \sum_{j \in \mathcal{R}} \mathbf{U}_{\mathcal{R}} \mathbf{b}_{1, \mathcal{R} \cup \{1\} \setminus \{j\}}^j, \quad (3.76)$$

$$\begin{aligned} \mathbf{X}_k = & \sum_{\mathcal{R} \in [[K] \setminus \{1,k\}]^r} \sum_{j \in \mathcal{R}} \mathbf{U}_{\mathcal{R}} \mathbf{b}_{k, \mathcal{R} \cup \{k\} \setminus \{j\}}^j \\ & + \sum_{\substack{\mathcal{R} \in [[K] \setminus \{k\}]^r: \\ 1 \in \mathcal{R}}} \sum_{j \in \mathcal{R}} \mathbf{U}_{\mathcal{R} \cup \{k\} \setminus \{1\}} \mathbf{b}_{k, \mathcal{R} \cup \{k\} \setminus \{j\}}^j, \end{aligned} \quad (3.77)$$

$k \in [K-1] \setminus \{1\},$

$$\begin{aligned} \mathbf{X}_K = & \sum_{\mathcal{R} \in [[K-1] \setminus \{1\}]^r} \sum_{j \in \mathcal{R}} \mathbf{U}_{\mathcal{R}} \mathbf{b}_{K, \mathcal{R} \cup \{K\} \setminus \{j\}}^j \\ & + \sum_{\substack{\mathcal{R} \in [[K-1]]^r: \\ 1 \in \mathcal{R}}} \sum_{j \in \mathcal{R} \setminus \{1\}} \mathbf{U}_{\mathcal{R} \cup \{K\} \setminus \{1\}} \mathbf{b}_{K, \mathcal{R} \cup \{K\} \setminus \{j\}}^j, \end{aligned} \quad (3.78)$$

where the precoding matrices $\{\mathbf{U}_{\mathcal{R}}\}_{\mathcal{R} \in [[K] \setminus \{1\}]^r}$ are described shortly.

Decoding: After receiving the respective sequence of T channel outputs $\mathbf{Y}_j \triangleq (Y_{j,1}, \dots, Y_{j,T})$,

for $j \in [K]$, each node removes the influence of the codewords corresponding to the messages that it can compute itself. The nodes' "cleaned" signals can then be written as:

$$\begin{aligned} \mathbf{Y}'_1 &= \underbrace{\sum_{\substack{\mathcal{R} \in [[K]]^r \\ 1 \in \mathcal{R}}} \sum_{k \in [K-1] \setminus \mathcal{R}} \mathbf{H}_{1,k} \mathbf{U}_{\mathcal{R} \cup \{k\} \setminus \{1\}} \mathbf{b}_{k, \mathcal{R} \cup \{k\} \setminus \{1\}}^1}_{\text{desired signal}} \\ &+ \sum_{\mathcal{R} \in [[K] \setminus \{1\}]^r} \sum_{k \in [K] \setminus \mathcal{R}} \mathbf{H}_{1,k} \mathbf{U}_{\mathcal{R}} \mathbf{v}_{\mathcal{R},k} + \mathbf{Z}_1, \end{aligned} \quad (3.79a)$$

$$\begin{aligned} \mathbf{Y}'_j &= \underbrace{\sum_{\substack{\mathcal{R} \in [[K] \setminus \{1\}]^r \\ j \in \mathcal{R}}} \sum_{k \in [K] \setminus \mathcal{R}} \mathbf{H}_{j,k} \mathbf{U}_{\mathcal{R}} \mathbf{b}_{k, \mathcal{R} \cup \{k\} \setminus \{j\}}^j}_{\text{desired signal}} \\ &+ \underbrace{\sum_{\substack{\mathcal{R} \in [[K]]^r \\ 1, j \in \mathcal{R}}} \sum_{k \in [K] \setminus \mathcal{R}} \mathbf{H}_{j,k} \mathbf{U}_{\mathcal{R} \cup \{k\} \setminus \{1\}} \mathbf{b}_{k, \mathcal{R}}^j}_{\text{desired signal}} \\ &+ \sum_{\substack{\mathcal{R} \in [[K] \setminus \{1\}]^r \\ j \notin \mathcal{R}}} \sum_{\substack{k \in [K] \setminus \mathcal{R} \\ k \neq j}} \mathbf{H}_{j,k} \mathbf{U}_{\mathcal{R}} \mathbf{v}_{\mathcal{R},k} \\ &+ \sum_{\substack{\mathcal{R} \in [[K]]^r \\ 1 \in \mathcal{R}, j \notin \mathcal{R}}} \sum_{k \in [K] \setminus \mathcal{R}} \mathbf{H}_{j,k} \mathbf{U}_{\mathcal{R} \cup \{k\} \setminus \{1\}} \mathbf{v}_{\mathcal{R},k} + \mathbf{Z}_j, \end{aligned} \quad j \in [K] \setminus \{1\}, \quad (3.79b)$$

where for ease of notation we defined for Nodes $k \in [K-1]$:

$$\mathbf{v}_{\mathcal{R},k} \triangleq \sum_{j \in \mathcal{R}} \mathbf{b}_{k, \mathcal{R} \cup \{k\} \setminus \{j\}}^j, \quad \forall \mathcal{R} \in [[K] \setminus \{k\}]^r, \quad (3.80)$$

and for the last Node K , since its signal to Node 1 is absent:

$$\mathbf{v}_{\mathcal{R},K} \triangleq \sum_{j \in \mathcal{R} \setminus \{1\}} \mathbf{b}_{k, \mathcal{R} \cup \{k\} \setminus \{j\}}^j, \quad \forall \mathcal{R} \in [[K-1]]^r. \quad (3.81)$$

Each Node j zero-forces the non-desired interference terms of its "cleaned" signal and decodes its intended messages $\{M_{k, \mathcal{T}}^j\}$.

Choice of IA Matrices $\{\mathbf{U}_{\mathcal{R}}\}$ and Analysis of Signal and Interference Spaces: Inspired by the IA scheme in [30], we choose each $T \times \eta^\Gamma$ precoding matrix $\mathbf{U}_{\mathcal{R}}$ so that its column-span includes all power products (with powers from 1 to η) of the channel matrices $\mathbf{H}_{j,k}$ that premultiply $\mathbf{U}_{\mathcal{R}}$ in (3.79) in the non-desired interference terms. Thus, for $\mathcal{R} \in [[K] \setminus \{1\}]^r$:

$$\mathbf{U}_{\mathcal{R}} \triangleq \left[\prod_{\mathbf{H} \in \mathcal{H}_{\mathcal{R}}} \mathbf{H}^{\alpha_{\mathcal{R}, \mathbf{H}}} \cdot \boldsymbol{\Xi}_{\mathcal{R}} : \forall \alpha_{\mathcal{R}} \in [\eta]^\Gamma \right], \quad (3.82)$$

where $\{\boldsymbol{\Xi}_{\mathcal{R}}\}_{\mathcal{R} \in [[K] \setminus \{1\}]^r}$ are i.i.d. random vectors independent of all channel matrices, noises, and messages,

$$\mathcal{H}_{\mathcal{R}} \triangleq \{\mathbf{H}_{j,k} : j \in [K] \setminus \mathcal{R}, k \in [K] \setminus \{j\}\} \setminus \{\mathbf{H}_{1,k} : k \in \mathcal{R}\}, \quad (3.83)$$

and $\alpha_{\mathcal{R}} \triangleq (\alpha_{\mathcal{R},\mathbf{H}} : \mathbf{H} \in \mathcal{H}_{\mathcal{R}})$. Notice that $|\mathcal{H}_{\mathcal{R}}| = \Gamma$ for any $\mathcal{R} \in [[\mathbf{K}] \setminus \{1\}]^r$.

Since the column-span of $\mathbf{U}_{\mathcal{R}}$ contains all power products of powers 1 to η of the channel matrices $\mathbf{H} \in \mathcal{H}_{\mathcal{R}}$, we have

$$\text{span}(\mathbf{H} \cdot \mathbf{U}_{\mathcal{R}}) \subseteq \text{span}(\mathbf{W}_{\mathcal{R}}), \quad \mathbf{H} \in \mathcal{H}_{\mathcal{R}}, \quad (3.84)$$

where we defined the $\mathbb{T} \times (\eta + 1)^{\Gamma}$ -matrix

$$\mathbf{W}_{\mathcal{R}} = \left[\prod_{\mathbf{H} \in \mathcal{H}_{\mathcal{R}}} \mathbf{H}^{\alpha_{\mathcal{R},\mathbf{H}}} \cdot \Xi_{\mathcal{R}} : \forall \alpha_{\mathcal{R}} \in [\eta + 1]^{\Gamma} \right],$$

for $\mathcal{R} \in [[\mathbf{K}] \setminus \{1\}]^r$.

(3.85)

The signal and interference space at Rx 1 is represented by the matrix:

$$\mathbf{\Lambda}_1 = \left[\underbrace{\mathbf{D}_1}_{\text{signal space}}, \underbrace{[\mathbf{W}_{\mathcal{R}}]_{\mathcal{R} \in [[\mathbf{K}] \setminus \{1\}]^r}}_{\text{interference space}} \right], \quad (3.86)$$

where

$$\mathbf{D}_1 \triangleq \left[\mathbf{H}_{1,k} \mathbf{U}_{\mathcal{R}} \right]_{\substack{k \in [\mathbf{K}-1] \setminus \{1\}, \\ \mathcal{R} \in [[\mathbf{K}] \setminus \{1\}]^r: \\ k \in \mathcal{R}}} \quad (3.87)$$

The matrix \mathbf{D}_1 represents the receiver's signal subspace and consists of $(\mathbf{K} - 2) \cdot \binom{\mathbf{K}-2}{r-1}$ matrices of dimension $\mathbb{T} \times \eta^{\Gamma}$. Matrix \mathbf{D}_1 is thus of dimension $\mathbb{T} \times (\mathbf{K} - 2) \cdot \binom{\mathbf{K}-2}{r-1} \cdot \eta^{\Gamma}$. Since the interference space consists of $\binom{\mathbf{K}-1}{r}$ matrices of dimension $\mathbb{T} \times (\eta + 1)^{\Gamma}$, and by the choice of \mathbb{T} in (3.69), Receiver 1's receive matrix $\mathbf{\Lambda}_1$ is square $\mathbb{T} \times \mathbb{T}$.

The receive space at Rx $j \in [\mathbf{K}] \setminus \{1\}$ is represented by the matrix:

$$\mathbf{\Lambda}_j \triangleq \left[\underbrace{\mathbf{D}_j}_{\text{signal space}}, \underbrace{[\mathbf{W}_{\mathcal{R}}]_{\mathcal{R} \in [[\mathbf{K}] \setminus \{1\}]^r: j \notin \mathcal{R}}}_{\text{interference space}} \right]. \quad (3.88)$$

where the signal subspace \mathbf{D}_j is given by a collection of $(\mathbf{K} - 1) \cdot \binom{\mathbf{K}-2}{r-1}$ matrices of dimension $\mathbb{T} \times \eta^{\Gamma}$:

$$\mathbf{D}_j \triangleq \left[\mathbf{H}_{j,k} \mathbf{U}_{\mathcal{R}} \right]_{\substack{j \in \mathcal{R}, \\ k \in [\mathbf{K}] \setminus \{j\}}} \quad (3.89)$$

Matrix \mathbf{D}_j represents the Receiver j 's signal subspace, consists of $(\mathbf{K} - 1) \cdot \binom{\mathbf{K}-2}{r-1}$ matrices of dimension $\mathbb{T} \times \eta^{\Gamma}$, and is thus itself of dimension $\mathbb{T} \times (\mathbf{K} - 1) \cdot \binom{\mathbf{K}-2}{r-1} \cdot \eta^{\Gamma}$. Receiver j 's interference space consists of a collection of $\binom{\mathbf{K}-2}{r}$ matrices of dimension $\mathbb{T} \times (\eta + 1)^{\Gamma}$, and thus the receive matrix $\mathbf{\Lambda}_j$ is of dimension $\mathbb{T} \times \tilde{\mathbb{T}}$, where

$$\tilde{\mathbb{T}} \triangleq (\mathbf{K} - 1) \cdot \binom{\mathbf{K} - 2}{r - 1} \cdot \eta^{\Gamma} + \binom{\mathbf{K} - 2}{r} \cdot (\eta + 1)^{\Gamma}. \quad (3.90)$$

According to Lemmas 3.2 and 3.3 below, matrices $\{\mathbf{\Lambda}_j\}_{j=1}^{\mathbf{K}}$ are full column-rank if each column has different exponent vector α , which follows by the way we constructed the matrices $\mathbf{U}_{\mathcal{R}}$ and $\mathbf{W}_{\mathcal{R}}$. Indeed:

- For each $\mathcal{R} \in [[\mathbf{K}] \setminus \{1\}]^r$, matrices $\mathbf{U}_{\mathcal{R}}$ and $\mathbf{W}_{\mathcal{R}}$ are constructed using a dedicated i.i.d. vector $\mathbf{\Xi}_{\mathcal{R}}$ that is independent of all other random variables in the system and thus the vectors $\mathbf{\Xi}_{\mathcal{R}}$ can play the roles of the vectors $\mathbf{\Xi}_i$ in Lemma 3.3.
- For each term $\mathbf{H}\mathbf{U}_{\mathcal{R}}$ in (3.87) and (3.89), we have $\mathbf{H} \notin \mathcal{H}_{\mathcal{R}}$. Thus \mathbf{H} is not used in the construction of neither $\mathbf{U}_{\mathcal{R}}$ nor $\mathbf{W}_{\mathcal{R}}$ and induces a unique exponent on the corresponding columns in the signal space which is 0 in all columns of the interference space $\mathbf{W}_{\mathcal{R}}$.

This proves that based on the “cleaned” signal (3.79), each receiving node j can separate the various desired signals from each other as well as from the non-desired interfering signals. Since each codeword $\mathbf{b}_{k,\mathcal{T}}^j$ occupies η^Γ dimensions out of the \mathbb{T} dimensions, we obtain that whenever

$$\frac{|\mathbf{b}_{k,\mathcal{T}}^j|}{\mathbb{T}} \leq \frac{\eta^\Gamma}{\mathbb{T}} \log P + o(\log P), \quad (3.91)$$

for an appropriate function $o(\log P)$ that grows slower than $\log P$, each codeword $\mathbf{b}_{k,\mathcal{T}}^j$ can be decoded with arbitrary small probability of error as $\eta \rightarrow \infty$.

Since $(\mathbf{K} - 2) \cdot \binom{\mathbf{K}-2}{r-1}$ codewords are sent to Node 1, and $r \binom{\mathbf{K}-1}{r}$ codewords to any other Node $j = 2, \dots, \mathbf{K}$, and since

$$\lim_{\eta \rightarrow \infty} \frac{\eta^\Gamma}{\mathbb{T}} = \frac{1}{(\mathbf{K} - 2) \binom{\mathbf{K}-2}{r-1} + \binom{\mathbf{K}-1}{r}}, \quad (3.92)$$

we conclude that a SDoF of

$$\begin{aligned} \text{SDoF} &= \frac{(\mathbf{K} - 2) \cdot \binom{\mathbf{K}-2}{r-1} + (\mathbf{K} - 1)r \binom{\mathbf{K}-1}{r}}{(\mathbf{K} - 2) \binom{\mathbf{K}-2}{r-1} + \binom{\mathbf{K}-1}{r}} \\ &= \frac{r(\mathbf{K} - 1)^2 + r(\mathbf{K} - 2)}{r(\mathbf{K} - 2) + \mathbf{K} - 1} \end{aligned} \quad (3.93)$$

is achievable over the system. This establishes achievability of (3.7).

Lemma 3.2 *Let $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m$ be independent random vectors with i.i.d. entries drawn according to continuous distributions. for any $L \leq m$ and L different exponent vectors*

$$\boldsymbol{\alpha}_j = (\alpha_{j,1}, \dots, \alpha_{j,m}) \in \mathbb{Z}_+^m, \quad j \in [L],$$

the $m \times L$ matrix \mathbf{M} with row- i and column- j entry

$$M_{i,j} = \prod_{k=1}^m (s_i)^{\alpha_{j,k}}, \quad i \in [m], j \in [L], \quad (3.94)$$

is full rank almost surely.

Lemma 3.3 *Consider numbers $\{n_1, n_2, \dots, n_{\tilde{\mathbf{K}}}\} \in \mathbb{Z}_+^{\tilde{\mathbf{K}}}$ so that their sum $\mathbf{C} \triangleq \sum_{i=1}^{\tilde{\mathbf{K}}} n_i \leq \mathbb{T}$. Assume that for each $i \in [\tilde{\mathbf{K}}]$ and $k \in [n_i]$, $\mathbf{B}_{i,k} \in \mathbb{C}^{\mathbb{T} \times \mathbb{T}}$ is a diagonal matrix so that all square sub-matrices of the following matrices $\{\mathbf{B}_i\}_{i \in [\tilde{\mathbf{K}}]}$ are full rank:*

$$\mathbf{B}_i \triangleq [\mathbf{B}_{i,1} \cdot \mathbf{1}_{\mathbb{T}}, \mathbf{B}_{i,2} \cdot \mathbf{1}_{\mathbb{T}}, \dots, \mathbf{B}_{i,n_i} \cdot \mathbf{1}_{\mathbb{T}}], \quad i \in [\tilde{\mathbf{K}}], \quad (3.95)$$

where $\mathbf{1}_T$ denotes a T -dimensional all-one column vector.

Let further $\{\Xi_i\}_{i \in \tilde{K}}$ be independent T -vectors with entries drawn i.i.d. from continuous distributions and define the $T \times n_i$ -matrices

$$\mathbf{A}_i \triangleq [\mathbf{B}_{i,1} \cdot \Xi_i, \mathbf{B}_{i,2} \cdot \Xi_i, \dots, \mathbf{B}_{i,n_i} \cdot \Xi_i], \quad i \in [\tilde{K}]. \quad (3.96)$$

Then, the $T \times C$ -matrix

$$\mathbf{\Lambda} \triangleq [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{\tilde{K}}] \quad (3.97)$$

has full column rank almost surely.

Proof: We assume that the matrix $\mathbf{\Lambda}$ is a square matrix i.e. $C = T$. If $T > C$, we take a square submatrix of $\mathbf{\Lambda}$ and perform the same proof steps on the submatrix.

Define

$$F(\Xi_1, \dots, \Xi_{\tilde{K}}) \triangleq \det(\mathbf{\Lambda}) \quad (3.98)$$

which is a polynomial of $\Xi_1, \Xi_2, \dots, \Xi_{\tilde{K}}$ as the determinant is a polynomial of the entries of $\mathbf{\Lambda}$.

For the vectors

$$\mathbf{d}_i = [\underbrace{0, \dots, 0}_{(n_1 + \dots + n_{i-1}) \text{ 0s}}, \underbrace{1, \dots, 1}_{n_i \text{ 1s}}, \underbrace{0, \dots, 0}_{(n_{i+1} + \dots + n_{\tilde{K}}) \text{ 0s}}]^T, \quad i \in \tilde{K}, \quad (3.99)$$

the polynomial evaluates to

$$F(\mathbf{d}_1, \dots, \mathbf{d}_{\tilde{K}}) = \det \begin{pmatrix} \mathbf{B}'_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}'_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}'_{\tilde{K}} \end{pmatrix} \quad (3.100)$$

$$= \prod_{i=1}^{\tilde{K}} \det(\mathbf{B}'_i) \neq 0 \quad (3.101)$$

where \mathbf{B}'_i is the $n_i \times n_i$ square sub-matrix of \mathbf{B}_i consisting of its rows $(n_1 + \dots + n_{i-1} + 1)$ to $(n_1 + \dots + n_{i-1} + n_i)$. The inequality holds by our assumption that all square sub-matrices of \mathbf{B}_i are full rank.

We conclude that F is a non-zero polynomial and thus $F(\Xi_1, \dots, \Xi_{\tilde{K}})$ equals 0 with probability 0 because the entries of $\Xi_1, \Xi_2, \dots, \Xi_{\tilde{K}}$ are drawn independently from continuous distributions. \blacksquare

3.6 A Scheme with IA and ZF for K odd and $r = (K - 1)/2$

Our second scheme is also based on the IA idea. However, now each message is cooperatively transmitted by a size- r set of transmitters \mathcal{T} so that it is received at a given node $j \in [K] \setminus \mathcal{T}$ while zero-forced at a group \mathcal{S} of $r - 1$ nodes in $[K] \setminus \{\mathcal{T} \cup \{j\}\}$. Since $r + r - 1 + 1 = K - 1$, there is only a single remaining node $\ell \in [K] \setminus \{\mathcal{S} \cup \mathcal{T} \cup \{j\}\}$ where the signal is experienced as interference. We choose to precode all Messages that cause interference at a given Node ℓ by the same precoding matrix \mathbf{U}_ℓ , for $\ell \in [K]$. As we explained, each

message will interfere only at a single receiving node, and thus this index ℓ is well-defined for each message. We construct the precoding matrix \mathbf{U}_ℓ so that all interferences at this Node ℓ align, thus leaving the remaining space for signaling dimensions. To summarize, if we use precoding matrix \mathbf{U}_ℓ for the transmission of a message from group \mathcal{T} to Node j , then we zero-force this signal at all nodes in $[\mathbf{K}] \setminus \{\mathcal{T} \cup \{j, \ell\}\}$ and we ensure that this signal aligns with all other interference signals at Node ℓ , the only node where it causes interference.

Given above idea, the main design parameter of our scheme is to choose for each message the receiving Node ℓ where it will cause interference, or equivalently the precoding matrix \mathbf{U}_ℓ employed for this message. Table 3.5 indicates the precoding matrix used to transmit each Message $M_{\mathcal{T}}^j$, for the case $\mathbf{K} = 5$ and $r = 2$. We observe that in this case our scheme transmits a message for each set $\mathcal{T} \in [[\mathbf{K}]]^r$ and each $j \in [\mathbf{K}] \setminus \mathcal{T}$. The table is thus full in the sense that all meaningful entries show a precoding matrix. Moreover, each meaningful entry describes the transmission of a single message $M_{\mathcal{T}}^j$. The table implicitly also indicates how to zero-force the transmit codewords. For example, Message $M_{\{1,2\}}^3$ is precoded by matrix \mathbf{U}_5 and its transmission is thus zero-forced at the remaining node 4 which neither transmits the message nor receives it, nor is chosen to handle it as interference.

Our task is to choose similar tables for any odd number $\mathbf{K} \geq 7$. In the choice that we propose shortly for general $\mathbf{K} \geq 7$, not all meaningful entries will be associated with a precoding matrix and some entries will be associated with more than one precoding matrix. (In this latter case, the corresponding message will have to be split into different submessages, one for each assigned precoding matrix.). It will then be more convenient to illustrate the usage of each precoding matrix in a separate table. For $\mathbf{K} = 7$ (and $r = 3$), Table 3.6 indicates the messages precoded by the matrix \mathbf{U}_7 and thus causing interference to Node 7. The general rule for $\mathbf{K} \geq 7$ with \mathbf{K} odd, is that we use precoding matrix \mathbf{U}_ℓ exactly $\mathbf{K} - r - 1 = r$ times in each column j , namely for the sets

$$\mathcal{T} \in \{\{j - r + 1, \dots, j - 1, t\} : t \in [\mathbf{K}] \setminus \{j - r + 1, \dots, j, \ell\}\}, \quad (3.102)$$

where the indices in (3.102) need to be understood with omission of the index ℓ and then taken modulo $\mathbf{K} - 1$. For example, as depicted in Table 3.6, for $\mathbf{K} = 7$, $r = 3$, and $j = 2$ the following sets \mathcal{T} send a message to Node 2 using precoding matrix \mathbf{U}_7 :

$$\begin{aligned} \mathcal{T} &\in \{\{1, 6, t\} : t \in [7] \setminus \{1, 2, 6, 7\}\} \\ &= \{\{1, 3, 6\}, \{1, 4, 6\}, \{1, 5, 6\}\}, \end{aligned} \quad (3.103)$$

where here we associated the index 1 with $j - 1$, the index 6 as $j - 2$, and t runs over the remaining set $[7] \setminus \{1, 2, 6, 7\}$. For the same parameters $\mathbf{K} = 7$, $r = 3$, and now column $j = 3$, the following sets send a message to node 3 using precoding matrix \mathbf{U}_7 :

$$\begin{aligned} \mathcal{T} &\in \{\{1, 2, t\} : t \in [7] \setminus \{1, 2, 3, 7\}\} \\ &= \{\{1, 2, 4\}, \{1, 2, 5\}, \{1, 2, 6\}\}. \end{aligned} \quad (3.104)$$

The choice of the tables in (3.102) is based on the following properties:

Remark 3.6 1. Consider the r rows $\mathcal{T}_1, \dots, \mathcal{T}_r$ that contain precoding matrix \mathbf{U}_ℓ in a given column j . Notice that they share $r - 1$ elements in common: the elements $j - r + 1, \dots, j - 1$ in (3.102), i.e., elements (1, 6) in (3.103) and elements (1, 2) in (3.104). The remaining element (namely the t in (3.102)) in each set is distinctive in the sense that it is not contained in any of the other transmit sets used in the same column.

2. In each row there is a single precoding matrix \mathbf{U}_ℓ . I.e., for each set \mathcal{T} there is a unique column-index j with entry \mathbf{U}_ℓ in the table.

3. For each $\ell \in [K]$, precoding matrix \mathbf{U}_ℓ occurs $K - r - 1 = r$ times in each column, except in column ℓ where it does not occur at all. I.e., each node in $[K] \setminus \{\ell\}$ receives $K - r - 1 = r$ different codewords that are precoded with matrix \mathbf{U}_ℓ .

Items 1) and 2) apply also to the precoding matrix assignment in Table 3.5 for $K = 5$ and $r = 2$.

To see item 2) in above remark, notice that for $r \geq 3$ the tuple of $r - 1$ consecutive (omitting index ℓ and modulo $K - 1$) numbers $(j - r + 1, \dots, j - 1)$ can only be present in a set \mathcal{T} that we use in column j but not in other columns.

Using above rule, for large values of K there are transmit sets $\mathcal{T} \in [[K]]^r$ and receiving Nodes j so that two different precoding matrices \mathbf{U}_ℓ and $\mathbf{U}_{\ell'}$, for $\ell, \ell' \in [K] \setminus (\mathcal{T} \cup \{j\})$ and $\ell \neq \ell'$, are assigned to the same row- \mathcal{T} and column- j entry of the table. For ease of notation, we capture this phenomena in the set $\mathcal{L}_\mathcal{T}^j$, which for each $\mathcal{T} \in [[K]]^r$ and $j \in [K] \setminus \mathcal{T}$ contains all indices ℓ so that the row- \mathcal{T} and column- j entry contains matrix \mathbf{U}_ℓ . We can then rephrase above observation as the remark that $|\mathcal{L}_\mathcal{T}^j|$ can be larger than 1. If this is the case, in our scheme Message $M_\mathcal{T}^j$ needs to be split into two submessages, which are then precoded by the matrices \mathbf{U}_ℓ and $\mathbf{U}_{\ell'}$, respectively. We shall therefore introduce the general notation $M_\mathcal{T}^{j,\ell}$ to denote the message that a transmit set \mathcal{T} sends to the receiving Node j using precoding matrix \mathbf{U}_ℓ . With some slight abuse of notation, we assume that the message $M_\mathcal{T}^{j,\ell}$ either denotes $M_\mathcal{T}^j$ or a submessage thereof.

We now describe the encoding and decodings and analyze the signal and interference spaces.

We fix a large parameter $\eta \in \mathbb{Z}^+$ (which we shall let tend to ∞) and define

$$\Gamma \triangleq r(K - 1) \tag{3.105}$$

$$\Upsilon \triangleq r(K - 1) \cdot \eta^\Gamma + (\eta + 1)^\Gamma. \tag{3.106}$$

For each message $M_\mathcal{T}^{j,\ell}$, construct a Gaussian codebook of power P and length η^Γ to encode each Message $M_\mathcal{T}^{j,\ell}$ into a codeword $\mathbf{b}_\mathcal{T}^{j,\ell}$.

Encoding Tx $q \in [K]$ forms its inputs as:

$$\mathbf{X}_q = \sum_{\substack{\mathcal{T} \subseteq [[K]]^r: \\ q \in \mathcal{T}}} \sum_{j \in [K] \setminus \mathcal{T}} \sum_{\ell \in \mathcal{L}_\mathcal{T}^j} \mathbf{v}_{[K] \setminus \{j, \ell\}, \mathcal{T}}^q \mathbf{U}_\ell \mathbf{b}_\mathcal{T}^{j,\ell}, \tag{3.107}$$

Table 3.5: Choice of precoding matrices in our scheme for $K = 5$ and $r = 2$. Each signal that is precoded with matrix \mathbf{U}_ℓ is zero-forced at the unique Node $i \neq \ell$ that is neither the intended Node j nor part of the transmitting set \mathcal{T} .

$\mathcal{T} \setminus j$	1	2	3	4	5
{1, 2}	x	x	\mathbf{U}_5	\mathbf{U}_3	\mathbf{U}_4
{1, 3}	x	\mathbf{U}_5	x	\mathbf{U}_2	\mathbf{U}_4
{1, 4}	x	\mathbf{U}_5	\mathbf{U}_2	x	\mathbf{U}_3
{1, 5}	x	\mathbf{U}_4	\mathbf{U}_2	\mathbf{U}_3	x
{2, 3}	\mathbf{U}_4	x	x	\mathbf{U}_5	\mathbf{U}_1
{2, 4}	\mathbf{U}_5	x	\mathbf{U}_1	x	\mathbf{U}_3
{2, 5}	\mathbf{U}_3	x	\mathbf{U}_4	\mathbf{U}_1	x
{3, 4}	\mathbf{U}_5	\mathbf{U}_1	x	x	\mathbf{U}_2
{3, 5}	\mathbf{U}_4	\mathbf{U}_1	x	\mathbf{U}_2	x
{4, 5}	\mathbf{U}_2	\mathbf{U}_3	\mathbf{U}_1	x	x

where $\mathbf{U}_1, \dots, \mathbf{U}_K$ denote the precoding-matrices that we will construct shortly and $\mathbf{V}_{\mathcal{R}, \mathcal{T}}$ denotes the node- q component of a matrix that zero-forces the signals emitted by the set of nodes \mathcal{T} at the receiving nodes $\mathcal{R} \setminus \mathcal{T}$ but not at the other nodes. This precoding matrix is also scaled in a way to satisfy the block-power constraint for all channel input signals. (Implicitly here we assume that \mathcal{T} and $\mathcal{R} \setminus \mathcal{T}$ are of sizes r and $r - 1$, respectively, so that the desired precoding matrix exists with probability 1.) For any set $\mathcal{T} = \{q_1, \dots, q_r\}$ and \mathcal{R} , define

$$\mathbf{V}_{\mathcal{R}, \mathcal{T}} \triangleq \begin{pmatrix} \mathbf{V}_{\mathcal{R}, \mathcal{T}}^{q_1} \\ \vdots \\ \mathbf{V}_{\mathcal{R}, \mathcal{T}}^{q_r} \end{pmatrix}. \quad (3.108)$$

With the proposed precoding matrices, and after each receive Node p removes the signals it can produce itself (i.e., the signals stemming from sets \mathcal{T} containing p), we can rewrite Node p 's equivalent receive signal as:

$$\begin{aligned} \mathbf{Y}'_p &= \sum_{\substack{\mathcal{T} \subseteq [K]^r: \\ p \notin \mathcal{T}}} \mathbf{H}_{p, \mathcal{T}} \sum_{\ell \in \mathcal{L}_{\mathcal{T}}^p} \mathbf{V}_{[K] \setminus \{p, \ell\}, \mathcal{T}} \mathbf{S}_{\mathcal{T}}^p \mathbf{U}_\ell \mathbf{b}_{\mathcal{T}}^{p, \ell} \\ &+ \sum_{j \in [K] \setminus p} \sum_{\substack{\mathcal{T} \subseteq [K] \setminus \{j, p\}: \\ p \in \mathcal{L}_{\mathcal{T}}^j}} \mathbf{G}_{\mathcal{T}}^{j, p} \mathbf{U}_p \mathbf{b}_{\mathcal{T}}^{j, p} + \mathbf{Z}_p. \end{aligned} \quad (3.109)$$

with

$$\mathbf{G}_{\mathcal{T}}^{j, p} \triangleq \mathbf{H}_{p, \mathcal{T}} \mathbf{V}_{[K] \setminus \{j, p\}, \mathcal{T}} \mathbf{S}_{\mathcal{T}}^j \quad (3.110)$$

Remark 3.7 Notice that all interfering signals at receiving Node p are precoded by the same matrix \mathbf{U}_p .

IA Matrices $\{\mathbf{U}_\ell\}$ Inspired by the IA schemes in [30], [90], we choose each $T \times \eta^\Gamma$ precoding matrix \mathbf{U}_ℓ so that its column-span includes all power products (with powers from 1 to η) of the “generalized” channel matrices $\mathbf{G}_T^{j,\ell}$ that premultiply \mathbf{U}_ℓ in (3.109):

$$\mathcal{G}_\ell \triangleq \left\{ \mathbf{G}_T^{j,\ell} : \forall j \in [K] \setminus \{\ell\}, \forall T \text{ s.t. } \ell \in \mathcal{L}_T^j \right\}. \quad (3.111)$$

Since the network is memoryless, the “generalized” channel matrices $\mathbf{G}_T^{j,\ell}$ are diagonal T -by- T matrices. Then, for each $k \in [K]$ construct the T -by- η^Γ matrix \mathbf{U}_ℓ by selecting each of its columns as a product of the elements in \mathcal{G}_ℓ multiplied with an independent i.i.d. random vector $\mathbf{\Xi}_\ell$:

$$\mathbf{U}_\ell = \left[\prod_{\mathbf{G} \in \mathcal{G}_\ell} \mathbf{G}^{\alpha_\ell, \mathbf{G}} \cdot \mathbf{\Xi}_\ell : \forall \alpha_\ell \in [\eta]^\Gamma \right], \quad (3.112)$$

where $\alpha_\ell \triangleq (\alpha_{\ell, \mathbf{G}} : \mathbf{G} \in \mathcal{G}_\ell)$ are exponent vectors of length Γ .

Decoding at Rx p The way we constructed our precoding matrices, we have:

$$\text{span}(\mathbf{G} \cdot \mathbf{U}_p) \subseteq \text{span}(\mathbf{W}_p), \quad \mathbf{G} \in \mathcal{G}_p, \quad (3.113)$$

where we defined the $T \times (\eta + 1)^\Gamma$ -matrix

$$\mathbf{W}_p = \left[\prod_{\mathbf{G} \in \mathcal{G}_p} \mathbf{G}^{\alpha_p, \mathbf{G}} \cdot \mathbf{\Xi}_p : \forall \alpha_p \in [\eta + 1]^\Gamma \right]. \quad (3.114)$$

The signal subspace at Rx p is given by:

$$\mathbf{D}_p \triangleq \left[\mathbf{H}_{p, \mathcal{T}} \mathbf{V}_{[K] \setminus \{p, \ell\}, \mathcal{T}} \mathbf{S}_T^p \mathbf{U}_\ell \right]_{\substack{\mathcal{T} \in [K]^r : p \notin \mathcal{T} \\ \ell \in \mathcal{L}_T^p}} \quad (3.115)$$

and its interference subspace is included in \mathbf{W}_p .

For each column of the signal space \mathbf{D}_p , Rx p projects its receive signal \mathbf{Y}'_p onto a vector that is orthogonal to all columns in the interference space \mathbf{W}_p and also to all other columns of \mathbf{D}_p . It can then decode the desired messages in an interference-free manner based on the various projections.

Analysis of Signal and Interference Subspaces If the columns of the matrix \mathbf{D}_p are linearly independent of each other and of the columns of \mathbf{W}_p , the following DoF is achievable to each Node p when we let $\eta \rightarrow \infty$:

$$\begin{cases} 6/7, & K = 5 \\ \frac{r(K-1)}{r(K-1)+1}, & K \geq 7 \end{cases}. \quad (3.116)$$

Remark 3.8 Notice the difference in our expressions (3.116) for $K = 5$ and $K \geq 7$. In fact, for $K = 5$, we fill all $\binom{K-1}{r}$ rows of each column with one of the precoding matrices. For $K \geq 7$ however we use each precoding matrix only $K - r - 1$ times in each column, and since in each row we can use $(K - 1)$ precoding matrices and $(K - r - 1)(K - 1) \leq \binom{K-1}{r}$ for $K \geq 7$, some of the entries in the table remain empty.

Accordingly, the SDoF of the entire system is given by

$$\text{SDoF} = \begin{cases} 30/7, & \text{K} = 5 \\ \frac{\text{Kr}(\text{K}-1)}{r(\text{K}-1)+1}, & \text{K} \geq 7 \end{cases}. \quad (3.117)$$

In the remainder of this section, we prove that for each p the columns of D_p are independent of each other and of the columns of W_p . The way we constructed the precoding matrices and by Lemma 3.3 at the end of the previous Section 3.5, it suffices to show that for each $\ell \neq p$ the matrix

$$\mathbf{\Lambda}_{p,\ell} = \left[\bar{\mathbf{G}}_{\mathcal{T}}^{p,\ell} \prod_{\substack{i \in [\text{K}] \setminus \{\ell\} \\ \tilde{\mathcal{T}} \in [[\text{K}] \setminus \{i,\ell\}]^r \\ \ell \in \mathcal{L}_{\tilde{\mathcal{T}}}^i}} \left(\mathbf{G}_{\mathcal{T}}^{i,\ell} \right)^{\alpha_{\ell,i,\tilde{\mathcal{T}}}} \mathbf{1}_{\mathcal{T}} : \forall \alpha_{\ell} \in [\eta]^{\Gamma} \right]_{\substack{\mathcal{T} \in [[\text{K}] \setminus \{\ell,p\}]^r \\ \ell \in \mathcal{L}_{\mathcal{T}}^p}} \quad (3.118)$$

has only full-rank square submatrices, where

$$\bar{\mathbf{G}}_{\mathcal{T}}^{p,\ell} \triangleq \mathbf{H}_{p,\mathcal{T}} \mathbf{V}_{[\text{K}] \setminus \{p,\ell\}, \mathcal{T}} \mathbf{S}_{\mathcal{T}}^p \quad (3.119)$$

is a diagonal T-by-T matrix.

By construction and the diagonal structure of the ‘‘generalized’’ channel coefficients, any square sub-matrix of the matrix $\mathbf{\Lambda}_{p,\ell}$, for $\ell \neq p$, has the same form as matrix \mathbf{M} in Equation (3.120) of Lemma 3.4 ahead, when one considers the diagonal entries of the ‘‘generalized’’ channel matrices $\{\bar{\mathbf{G}}_{\mathcal{T}}^{p,\ell}\}_{\mathcal{T}: \ell \in \mathcal{L}_{\mathcal{T}}^p}$ and $\{\mathbf{G}_{\mathcal{T}}^{i,\ell}\}_{\substack{i \in [\text{K}] \setminus \{\ell\} \\ \mathcal{T}: \ell \in \mathcal{L}_{\mathcal{T}}^i}}$ as the outcomes of the functions $\mathbf{f}_1, \dots, \mathbf{f}_L$. The inputs of these functions are the random channel coefficients $\{H_{p',q}(t)\}$ and the entries of the diagonal matrices $\{\mathbf{S}_{\mathcal{T}}^j\}$ which in Lemma 3.4 can thus play the role of the i.i.d. random variables in the vector \mathbf{x}_t . By Lemma 3.4 it thus suffices to show that the ‘‘generalized’’ channel matrices $\{\bar{\mathbf{G}}_{\mathcal{T}}^{p,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^p}$ and $\{\mathbf{G}_{\mathcal{T}}^{i,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^i}$ are algebraically independent functions of the channel coefficients $\{H_{p',q}(t)\}$ and the entries of $\{\mathbf{S}_{\mathcal{T}}^j\}$. In our proof, we will exploit the structure that we imposed for our choice of the precoding matrices, see Remark 3.6. Especially observation 1) in Remark 3.6 that for given $j \neq \ell$ precoding matrix U_{ℓ} is used to send messages to Receiver j from transmit sets $\mathcal{T}_1, \dots, \mathcal{T}_r$ that each have one distinct index that is not present in the other transmit sets.

Notice that $\{\bar{\mathbf{G}}_{\mathcal{T}}^{p,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^p}$ and $\{\mathbf{G}_{\mathcal{T}}^{i,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^i}$ are all diagonal matrices with the t -th elements only depending on the time- t channel coefficients $\{H_{p',q}(t)\}$ and the t -th components of the diagonal matrices $\{\mathbf{S}_{\mathcal{T}}^j\}$. We restrict to a single time-instance $t \in \{1, \dots, \text{T}\}$ and drop this time-index for convenience. Henceforth, the random variables $\{\bar{G}_{\mathcal{T}}^{p,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^p}$ and $\{G_{\mathcal{T}}^{i,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^i}$, and $\{S_{\mathcal{T}}^j\}$ refer to the t -th diagonal elements of the corresponding matrices and $\{H_{p',q}\}$ to the corresponding time- t channel coefficients.

Recall that p and ℓ are fixed and notice the following:

- Since each transmitted codeword interferes only at a single node, each element $G_{\mathcal{T}}^{i,\ell}$, with \mathcal{T} and i so that $\ell \in \mathcal{L}_{\mathcal{T}}^i$, depends on a different auxiliary random variable $S_{\mathcal{T}}^i$. The functions $\{G_{\mathcal{T}}^{i,\ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^i}$ are thus algebraically independent because the factors in

front of these auxiliary random variables $\mathbf{H}_{\ell, \mathcal{T}} \mathbf{V}_{[\mathcal{K}] \setminus \{i, \ell\}, \mathcal{T}}$ are non-zero with probability 1.

- Each function $\bar{G}_{\mathcal{T}}^{p, \ell}$ only depends on $S_{\mathcal{T}}^p$ but not on the other S -random variables.
- Each set \mathcal{T} for which $\ell \in \mathcal{L}_{\mathcal{T}}^p$ has a distinct element t (see (3.102) and Item 1) of Remark 3.6). For a given such set \mathcal{T} for which $\ell \in \mathcal{L}_{\mathcal{T}}^p$, the two functions $\bar{G}_{\mathcal{T}}^{p, \ell}$ and $G_{\mathcal{T}}^{p, \ell}$ thus each depends on a different channel coefficient $H_{p, t}$ and $H_{\ell, t}$, respectively, that does not influence any of the other functions $\bar{G}_{\tilde{\mathcal{T}}}^{p, \ell}$ and $G_{\tilde{\mathcal{T}}}^{p, \ell}$ for sets $\tilde{\mathcal{T}} \neq \mathcal{T}$ satisfying $\ell \in \mathcal{L}_{p, \tilde{\mathcal{T}}}$. Again, with probability 1 the factors in front of the distinct channel coefficients $H_{p, t}$ and $H_{\ell, t}$ are non-zero, which establishes algebraic independence of our functions.

All these considerations can be combined to conclude that the functions $\{\bar{G}_{\mathcal{T}}^{p, \ell}\}_{\ell \in \mathcal{L}_{\mathcal{T}}^p}$ and $\{G_{\mathcal{T}}^{i, \ell}\}_{\substack{i \in [\mathcal{K}] \setminus \{\ell\} \\ \ell \in \mathcal{L}_{\mathcal{T}}^p}}$ are algebraically independent.

Lemma 3.4 (Lemmas 3 and 4 in [90]) *Let $\mathbf{f} = (f_1, f_2, \dots, f_m) \in \mathbb{C}^m$ be a vector of rational functions and let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\tau}$ be i.i.d. random vectors with i.i.d. entries drawn according to continuous distributions. Define*

$$\mathbf{s}_i \triangleq \mathbf{f}(\mathbf{x}_i), \quad i \in [\tau].$$

For any $L \leq \tau$ and L different exponent vectors

$$\boldsymbol{\alpha}_j = (\alpha_{j,1}, \dots, \alpha_{j,m}) \in \mathbb{Z}_+^m, \quad j \in [L],$$

the $T \times L$ matrix \mathbf{M} with row- i and column- j entry

$$M_{i,j} = \prod_{k=1}^m (s_{i,k})^{\alpha_{j,k}}, \quad i \in [\tau], j \in [L], \quad (3.120)$$

is full rank almost surely, if and only if the functions \mathbf{f} are algebraically independent, i.e., if and only if the Jacobian $[\frac{\partial f_n}{\partial x_i}]_{(i,n)}$ is of rank m .

3.7 Proof of the NDT Lower Bound in Theorem 3.1

Consider a fixed file assignment (map phase), and for any positive power P a sequence (in T) of wireless distributed computing systems satisfying (2.33) for the given file assignment. (Since for finite N there are only a finite number of different file assignments irrespective of P and T , we can fix the assignment.) The following limiting behaviour must hold.

Lemma 3.5 *Consider two disjoint sets \mathcal{T} and \mathcal{R} of same size*

$$|\mathcal{T}| = |\mathcal{R}|, \quad (3.121)$$

and define $\mathcal{F} \triangleq [\mathcal{K}] \setminus (\mathcal{R} \cup \mathcal{T})$. Let $\mathcal{M} \subseteq [N]$ be the set of files known only to nodes \mathcal{T} but not to any other node and partition the set of all IVAs \mathcal{A} it into the following disjoint subsets:

$$\mathcal{W}_r \triangleq \{a_{j,m}\}_{\substack{j \in \mathcal{R} \\ m \in [N] \setminus \mathcal{M}_j}}, \quad (3.122)$$

$$\mathcal{W}_t \triangleq \{a_{j,m}\}_{\substack{j \in \mathcal{T} \\ m \in \mathcal{M} \setminus \mathcal{M}_j}}. \quad (3.123)$$

For any sequence of distributed computing systems:

$$d \triangleq \overline{\lim}_{P \rightarrow \infty} \overline{\lim}_{T \rightarrow \infty} \frac{A}{T \log P} \leq \frac{|\mathcal{T}|}{|\mathcal{W}_t| + |\mathcal{W}_r|} \quad (3.124)$$

(Notice that \mathcal{W}_r denotes the set of all IVAs intended to nodes in \mathcal{R} and \mathcal{W}_t the set of IVAs deduced from files in \mathcal{M} and intended for nodes not in \mathcal{R} .)

Proof: Denote by \mathcal{H} the set of all channel coefficients to all nodes in the system and define $\mathcal{W}_c \triangleq \mathcal{A} \setminus (\mathcal{W}_r \cup \mathcal{W}_t)$. Since channel coefficients and IVAs are independent, we have

$$H(\mathcal{W}_t, \mathcal{W}_r) = H(\mathcal{W}_t, \mathcal{W}_r | \mathcal{W}_c, \mathcal{H}) \quad (3.125)$$

$$\begin{aligned} &= I(\mathcal{W}_t, \mathcal{W}_r; \mathbf{Y}_{\mathcal{R}} | \mathcal{W}_c, \mathcal{H}) \\ &\quad + H(\mathcal{W}_t, \mathcal{W}_r | \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) \end{aligned} \quad (3.126)$$

$$\begin{aligned} &= h(\mathbf{Y}_{\mathcal{R}} | \mathcal{W}_c, \mathcal{H}) - h(\mathbf{Z}_{\mathcal{R}}) \\ &\quad + H(\mathcal{W}_r | \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) \end{aligned} \quad (3.127)$$

$$\quad + H(\mathcal{W}_t | \mathcal{W}_r, \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) \quad (3.128)$$

$$\begin{aligned} &\leq h(\mathbf{Y}_{\mathcal{R}} | \mathcal{W}_c, \mathcal{H}) - h(\mathbf{Z}_{\mathcal{R}}) \\ &\quad + T\epsilon_T + H(\mathcal{W}_t | \mathcal{W}_r, \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}), \end{aligned} \quad (3.129)$$

where we defined $\mathbf{Y}_{\mathcal{A}} \triangleq [\mathbf{Y}_j]_{j \in \mathcal{A}}$ for a set $\mathcal{A} \subseteq [\mathsf{K}]$ and ϵ_T is a vanishing sequence as $T \rightarrow \infty$. Here the inequality holds by Fano's inequality, because \mathcal{W}_r is decoded from $\mathbf{Y}_{\mathcal{R}}$ and \mathcal{W}_c , and because we impose vanishing probability of error (2.33).

Again by Fano's inequality and by (2.33), there exists a vanishing sequence ϵ'_T such that

$$\begin{aligned} &H(\mathcal{W}_t | \mathcal{W}_r, \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) \\ &\leq I(\mathcal{W}_t; \mathbf{Y}_{\mathcal{F}} | \mathcal{W}_r, \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) + T\epsilon'_T \end{aligned} \quad (3.130)$$

$$= h(\mathbf{Y}_{\mathcal{F}} | \mathcal{W}_r, \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) \quad (3.131)$$

$$\quad - h(\mathbf{Y}_{\mathcal{F}} | \mathcal{W}_r, \mathcal{W}_t, \mathcal{W}_c, \mathbf{Y}_{\mathcal{R}}, \mathcal{H}) + T\epsilon'_T \quad (3.132)$$

$$\leq h(\bar{\mathbf{Y}}_{\mathcal{F}} | \bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}) - h(\mathbf{Z}_{\mathcal{F}}) + T\epsilon'_T, \quad (3.133)$$

where $\bar{\mathbf{Y}}_{\mathcal{A}} \triangleq [\bar{\mathbf{Y}}_j]_{j \in \mathcal{A}}$ and $\bar{\mathbf{Y}}_j$ denotes Node j 's "cleaned" signal without the inputs that do not depend on files in \mathcal{M} but only on IVAs $\mathcal{W}_r \cup \mathcal{W}_c$:

$$\bar{\mathbf{Y}}_j \triangleq \mathbf{H}_{j, \mathcal{T}} \mathbf{X}_{\mathcal{T}} + \mathbf{Z}_j, \quad j \in \mathcal{T} \cup \mathcal{F}.$$

Here, $\mathbf{H}_{\mathcal{A}, \mathcal{B}}$ denotes the channel matrix from set \mathcal{B} to set \mathcal{A} .

To bound the first term in (3.133), we introduce a random variable E indicating whether the matrix $\mathbf{H}_{\mathcal{R}, \mathcal{T}}$ is invertible ($E = 1$) or not ($E = 0$). If this matrix is invertible and $E = 1$, then the input vector $\mathbf{X}_{\mathcal{T}}$ can be computed from $\bar{\mathbf{Y}}_{\mathcal{R}}$ up to noise terms. Based on this observation and defining the residual noise terms

$$\bar{\mathbf{Z}}_j \triangleq \mathbf{Z}_j - \mathbf{H}_{j, \mathcal{T}} \mathbf{H}_{\mathcal{R}, \mathcal{T}}^{-1} \mathbf{Z}_{\mathcal{R}}, \quad \text{if } E = 1, \quad (3.134)$$

we obtain:

$$\begin{aligned}
 & h(\bar{\mathbf{Y}}_{\mathcal{F}}|\bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}) \\
 & \leq \mathbb{P}(E = 1) \cdot h(\bar{\mathbf{Z}}_{\mathcal{F}}|\bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}, E = 1) \\
 & \quad + \mathbb{P}(E = 0) \cdot h(\bar{\mathbf{Y}}_{\mathcal{F}}|\bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}, E = 0) \tag{3.135} \\
 & \leq h(\bar{\mathbf{Z}}_{\mathcal{F}}) + \mathbb{P}(E = 0)h(\bar{\mathbf{Y}}_{\mathcal{F}}|\bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}, E = 0). \tag{3.136}
 \end{aligned}$$

Since the channel coefficients follow continuous distribution, $\mathbf{H}_{\mathcal{R}, \mathcal{T}}$ is invertible almost surely, implying $\mathbb{P}(E = 0) = 0$. By the boundedness of the entropy term $h(\bar{\mathbf{Y}}_{\mathcal{F}}|\bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}, E = 0)$ (since power P and channel coefficients are bounded), this implies

$$h(\bar{\mathbf{Y}}_{(\mathcal{F} \cup \mathcal{T})}|\bar{\mathbf{Y}}_{\mathcal{R}}, \mathcal{H}) \leq h(\bar{\mathbf{Z}}_{\mathcal{F}}),$$

which combined with (3.129) and (3.133) yields:

$$\begin{aligned}
 H(\mathcal{W}_t, \mathcal{W}_r) & \leq h(\mathbf{Y}_{\mathcal{R}}|\mathcal{H}) - h(\mathbf{Z}_{\mathcal{R}}) + h(\bar{\mathbf{Z}}_{\mathcal{F}}) \\
 & \quad - h(\mathbf{Z}_{\mathcal{F}}) + \mathbb{T}(\epsilon_{\mathbb{T}} + \epsilon'_{\mathbb{T}}) \\
 & \leq \mathbb{T}|\mathcal{R}| \log(P) + \mathbb{T}C_{\mathbb{T}, \mathcal{H}}, \tag{3.137}
 \end{aligned}$$

where $C_{\mathbb{T}, \mathcal{H}}$ is a function that is uniformly bounded over all realizations of channel matrices and powers P . Noticing

$$H(\mathcal{W}_t, \mathcal{W}_r) = A(|\mathcal{W}_t| + |\mathcal{W}_r|), \tag{3.138}$$

dividing (3.137) by $\mathbb{T} \log(P)$, and letting $P \rightarrow \infty$, establishes the lemma because $|\mathcal{R}| = |\mathcal{T}|$ and $\mathbb{T}C_{\mathbb{T}, \mathcal{H}}$ is bounded. \blacksquare

For each subset $\mathcal{T} \subseteq [\mathbb{K}]$, let $\mathcal{B}_{\mathcal{T}}^j$ denote the set of IVAs that are computed exclusively at nodes in set \mathcal{T} and intended for reduce function j . Define $b_{\mathcal{T}} = |\mathcal{B}_{\mathcal{T}}^j|$, which does not depend on the index of the reduce function $j \in [\mathbb{K}] \setminus \mathcal{T}$.

Choose two disjoint subsets \mathcal{T} and \mathcal{R} of same size $|\mathcal{T}| = |\mathcal{R}|$. By Lemma 3.5, and rewriting the sets \mathcal{W}_t and \mathcal{W}_r in the lemma in terms of the sets $\{\mathcal{B}_{\mathcal{T}}^j\}$, we obtain:

$$\frac{|\mathcal{T}|}{d} \geq \sum_{\mathcal{T} \subseteq [\mathbb{K}]} \sum_{j \in \mathcal{R} \setminus \mathcal{T}} |\mathcal{B}_{\mathcal{T}}^j| + \sum_{\mathcal{G} \subseteq \mathcal{T}} \sum_{j \in [\mathbb{K}] \setminus (\mathcal{R} \cup \mathcal{T})} |\mathcal{B}_{\mathcal{G}}^j| \tag{3.139}$$

$$= \sum_{\mathcal{T} \subseteq [\mathbb{K}]} |\mathcal{R} \setminus \mathcal{T}| \cdot b_{\mathcal{T}} + \sum_{\mathcal{G} \subseteq \mathcal{T}} (\mathbb{K} - |\mathcal{R}| - |\mathcal{T}|) \cdot b_{\mathcal{G}}. \tag{3.140}$$

Summing up Equality (3.140) over all sets \mathcal{T} and \mathcal{R} of constant size $t \leq \mathsf{K}/2$, we obtain:

$$\begin{aligned} & \binom{\mathsf{K}}{t} \cdot \binom{\mathsf{K}-t}{t} \cdot \frac{t}{d} \\ & \geq \sum_{\mathcal{T} \in \llbracket [\mathsf{K}] \rrbracket^t} \sum_{\mathcal{R} \in \llbracket [\mathsf{K}] \setminus \mathcal{T} \rrbracket^t} \sum_{\mathcal{T} \subseteq [\mathsf{K}]} |\mathcal{R} \setminus \mathcal{T}| \cdot b_{\mathcal{T}} \\ & \quad + \sum_{\mathcal{T} \in \llbracket [\mathsf{K}] \rrbracket^t} \sum_{\mathcal{R} \in \llbracket [\mathsf{K}] \setminus \mathcal{T} \rrbracket^t} \sum_{\mathcal{G} \subseteq \mathcal{T}} (\mathsf{K} - 2t) \cdot b_{\mathcal{G}} \end{aligned} \quad (3.141)$$

$$\begin{aligned} & = \sum_{\mathcal{T} \subseteq [\mathsf{K}]} \binom{\mathsf{K}-t}{t} \sum_{\ell \in [\mathsf{K}] \setminus \mathcal{T}} \sum_{\mathcal{R} \in \llbracket [\mathsf{K}] \rrbracket^t} \mathbb{1}_{\ell \in \mathcal{R}} b_{\mathcal{T}} \\ & \quad + \sum_{\substack{\mathcal{G} \subseteq [\mathsf{K}]: \\ |\mathcal{G}| \leq t}} \binom{\mathsf{K}-|\mathcal{G}|}{t-|\mathcal{G}|} \cdot \binom{\mathsf{K}-t}{t} \cdot (\mathsf{K}-2t) b_{\mathcal{G}} \end{aligned} \quad (3.142)$$

$$\begin{aligned} & = \sum_{\mathcal{T} \subseteq [\mathsf{K}]} \binom{\mathsf{K}}{t} \cdot \binom{\mathsf{K}-t}{t} \cdot (\mathsf{K}-|\mathcal{T}|) \frac{t}{\mathsf{K}} b_{\mathcal{T}} \\ & \quad + \sum_{\substack{\mathcal{G} \subseteq [\mathsf{K}]: \\ |\mathcal{G}| \leq t}} \binom{\mathsf{K}-|\mathcal{G}|}{t-|\mathcal{G}|} \cdot \binom{\mathsf{K}-t}{t} \cdot (\mathsf{K}-2t) b_{\mathcal{G}} \end{aligned} \quad (3.143)$$

$$\begin{aligned} & = \binom{\mathsf{K}}{t} \cdot \binom{\mathsf{K}-t}{t} \cdot t \left(\mathsf{N} - \frac{r\mathsf{N}}{\mathsf{K}} \right) \\ & \quad + \sum_{\substack{\mathcal{G} \subseteq [\mathsf{K}]: \\ |\mathcal{G}| \leq t}} \binom{\mathsf{K}-|\mathcal{G}|}{t-|\mathcal{G}|} \cdot \binom{\mathsf{K}-t}{t} \cdot (\mathsf{K}-2t) b_{\mathcal{G}}, \end{aligned} \quad (3.144)$$

where we define $\binom{a}{0} = 1$ for any positive integer a . The second equality holds because for a given element ℓ , there are $\binom{\mathsf{K}-1}{t-1} = \binom{\mathsf{K}}{t} \frac{t}{\mathsf{K}}$ admissible sets \mathcal{R} and the last equality holds because

$$\sum_{\mathcal{T} \subseteq [\mathsf{K}]} b_{\mathcal{T}} = \mathsf{N}, \quad \sum_{\mathcal{T} \subseteq [\mathsf{K}]} |\mathcal{T}| \cdot b_{\mathcal{T}} \leq r \cdot \mathsf{N}. \quad (3.145)$$

Dividing both sides of (3.144) by $\binom{\mathsf{K}}{t} \binom{\mathsf{K}-t}{t} t$, and defining $b_i \triangleq \sum_{\mathcal{T} \in \llbracket [\mathsf{K}] \rrbracket^i} b_{\mathcal{T}}$, for any $t \in \llbracket [\mathsf{K}/2] \rrbracket$ we obtain:

$$\frac{1}{d} \geq \mathsf{N} - \frac{r \cdot \mathsf{N}}{\mathsf{K}} + \min_{\substack{b_1, \dots, b_{\mathsf{K}} \in \mathbb{Z}^+: \\ \sum_{i=1}^{\mathsf{K}} b_i = \mathsf{N} \\ \sum_{i=1}^{\mathsf{K}} i b_i \leq r\mathsf{N}}} \sum_{i=1}^t C_t(i) b_i, \quad t \in \llbracket [\mathsf{K}/2] \rrbracket, \quad (3.146)$$

where $C_t(i)$ is defined in (3.12).

For any $t \in \llbracket [\mathsf{K}/2] \rrbracket$, the sequence of coefficients $C_t(1), C_t(2), \dots, C_t(t)$ is convex and non-increasing, see Appendix B.1. Based on this convexity, it can be shown (see Appendix B.2) that for any $r < t + 1$ there exists a solution to the minimization problem in (3.146) putting only positive masses on $b_{\lfloor r \rfloor}^*$ and $b_{\lceil r \rceil}^*$ in the unique way satisfying

$$b_{\lfloor r \rfloor}^* + b_{\lceil r \rceil}^* = \mathsf{N} \quad (3.147)$$

$$\lfloor r \rfloor b_{\lfloor r \rfloor}^* + \lceil r \rceil b_{\lceil r \rceil}^* = r\mathsf{N}. \quad (3.148)$$

For $r \geq t+1$ an optimal solution consists of setting $b_{\lfloor r \rfloor}^* = N$, in which case the minimization in (3.146) evaluates to 0.

For $r \geq 2$, the lower bound on the NDT in the theorem is then obtained by plugging these optimum values into bound (3.146) for the choice $t = \lfloor K/2 \rfloor$. For $r = 1$ we choose $t = 1$, and for $r \in (1, 2)$ we maximize over the value of t .

3.8 Conclusion

This chapter presents an improved upper bound and the first information-theoretic lower bound on the computation-NDT tradeoff of full-duplex wireless MapReduce systems. The upper bound is obtained by zero-forcing and a novel IA scheme that is tailored to the information cancellation capabilities of the nodes in a MapReduce system. As a conclusion of this work, we observe that linear beamforming, zero-forcing, and interference cancellation are optimal when each node can store at least half of the files, but suboptimal otherwise. It's worth noting that IA algorithms require large precoding matrices, leading to significant storage and computational costs. The design of a practical IA algorithm for a MapReduce system could be an interesting topic for future research.

Table 3.6: Codewords precoded by matrix \mathbf{U}_7 when $K = 7$ and $r = 3$. Each codeword is zero-forced at the two nodes not belonging to the transmit \mathcal{T} and not equal to the receive node j or to 7.

$\mathcal{T} \setminus j$	1	2	3	4	5	6	7
{1, 2, 3}	x	x	x	\mathbf{U}_7	o	o	o
{1, 2, 4}	x	x	\mathbf{U}_7	x	o	o	o
{1, 2, 5}	x	x	\mathbf{U}_7	o	x	o	o
{1, 2, 6}	x	x	\mathbf{U}_7	o	o	x	o
{1, 2, 7}	x	x	o	o	o	o	x
{1, 3, 4}	x	o	x	x	\mathbf{U}_7	o	o
{1, 3, 5}	x	o	x	o	x	o	o
{1, 3, 6}	x	\mathbf{U}_7	x	o	o	x	o
{1, 3, 7}	x	o	x	o	o	o	x
{1, 4, 5}	x	o	o	x	x	\mathbf{U}_7	o
{1, 4, 6}	x	\mathbf{U}_7	o	x	o	x	o
{1, 4, 7}	x	o	o	x	o	o	x
{1, 5, 6}	x	\mathbf{U}_7	o	o	x	x	o
{1, 5, 7}	x	o	o	o	x	o	x
{1, 6, 7}	x	o	o	o	o	x	x
{2, 3, 4}	o	x	x	x	\mathbf{U}_7	o	o
{2, 3, 5}	o	x	x	\mathbf{U}_7	x	o	o
{2, 3, 6}	o	x	x	\mathbf{U}_7	o	x	o
{2, 3, 7}	o	x	x	o	o	o	x
{2, 4, 5}	o	x	o	x	x	\mathbf{U}_7	o
{2, 4, 6}	o	x	o	x	o	x	o
{2, 4, 7}	o	x	o	x	o	o	x
{2, 5, 6}	\mathbf{U}_7	x	o	o	x	x	o
{2, 5, 7}	o	x	o	o	x	o	x
{2, 6, 7}	o	x	o	o	o	x	x
{3, 4, 5}	o	o	x	x	x	\mathbf{U}_7	o
{3, 4, 6}	o	o	x	x	\mathbf{U}_7	x	o
{3, 4, 7}	o	o	x	x	o	o	x
{3, 5, 6}	\mathbf{U}_7	x	o	x	x	o	o
{3, 5, 7}	o	o	x	o	x	o	x
{3, 6, 7}	o	o	x	o	o	x	x
{4, 5, 6}	\mathbf{U}_7	o	o	x	x	x	o
{4, 5, 7}	o	o	o	x	x	o	x
{4, 6, 7}	o	o	o	x	o	x	x
{5, 6, 7}	o	o	o	o	x	x	x

Multi-bit Quantizer Optimization for Distributed Estimation

4.1 Introduction

In numerous surveillance applications, sensors are positioned in various locations with the aim of measuring a common phenomenon and, consequently, the same parameter [43]–[45]. The sensors do not individually carry out the final parameter estimation due to the fact that the local data often suffer from noises. Instead, they transmit their data after quantization through a propagation channel to a FC, which conducts the estimation. Discovering multi-bit quantizers constitutes a key issue in estimation.

More precisely, most works consider that *i)* the quantized version of a new arrival sample is sent at the FC according to a sequential Round-Robin technique, *ii)* then the FC collects all the quantized samples during one round to perform the estimation, and *iii)* so the performance is evaluated for one round. In that context, Fisher information and the CRB have been calculated under different assumptions [46]–[53]. In [52], [53], information-theoretic tools are used and the conclusion is independent of any quantizer but finally, the provided bounds are loose. In other above-mentioned papers, only performances with specified quantizers are considered but without quantizer optimization.

In [54], [55], [57]–[59], the authors propose quantizer optimization for different configurations and assumptions. For instance, [54] proposes an optimal deterministic multi-bit quantizer for one sensor in low SNR case. In [55], Bayesian CRB and a dynamic programming approach are considered to exhibit the optimal multi-bit deterministic quantizer in a single sensor context. In [56], the authors propose a deterministic quantizer which is obtained by minimizing the CRB when the parameter vanishes (i.e., at low SNR) with a particle swarm optimization algorithm. In [57], multi-sensors are considered but each is equipped with a one-bit quantizer. The quantizer is assumed to be random at each sensor but the related threshold cumulative density function is linear piece-wise and data-dependent. The criterion is minimax in the sense they minimize the worst CRB with respect to the parameter range. In [59], the same setup as [57] is considered but they find the best threshold distribution without the linear-piecewise structure assumption. Finally, in [58], multiple-sensor scenario is considered. Mathematically, the authors propose to

optimize the worst case CRB in an iterative way with respect to the thresholds.

In the chapter, we consider multi-bit quantizers in the multiple-sensor scenario for the min-max approach. We assume that each sensor has a random quantizer coming from a common distribution between sensors. And this common distribution is optimized when the number of sensors is large enough. In [59], the same approach was considered but for one-bit quantizer. Here, the challenge is to extend [59] to multi-bit quantizer. This extension is not straightforward for two reasons: *i*) expressing our CRB in closed-form requires order statistics, *ii*) the obtained optimization problem is not convex anymore. In the asymptotic regime with respect to the number of sensors, we propose a framework based on SGP to obtain an optimized quantizer for the worst-case of the target parameter. Under the assumption that all thresholds of quantizers are generated according to a distribution, we express the CRB using order statistics, and the goal of the framework is to explore the optimal distribution for the CRB. After discretizing, we convert the optimization problem into a SGP and can be solved by the algorithm proposed in [81]. The obtained quantizer outperforms uniformly distributed, regular deterministic quantizers and those proposed in [55], [56] in the mid-to-high SNR regime. Another interesting observation is that the quantizer performs well even with a limited number of sensors, despite being designed under the assumption of a large number of sensors.

The rest of this chapter is organized as follows. In Section 4.2, we review the framework of Geometric Programming (GP) and SGP. In Section 4.3, we introduce the system model for distributed estimation system with a FC. In Section 4.4, we derive the CRB for the general case. Then, the asymptotic CRB is derived in Section 4.5. The discretizing methods and the optimization framework with SGP are also given in the section. The experiment results are presented in Section 4.6 . Finally, Section 4.7 concludes this chapter.

4.2 Preliminaries

4.2.1 Geometric Programming (GP)

In this section, we present of the definition of GP, which is the method to solve a class of optimization problem. Let x_1, \dots, x_N denote N positive variables, and the vector $\mathbf{x} = (x_1, \dots, x_N)$. We introduce two definitions

Definition 4.1 (Monomial functions) *A monomial $g(\mathbf{x})$ is a function which takes the form,*

$$g(\mathbf{x}) = c \cdot \prod_{n=1}^N x_n^{a_n}, \quad (4.1)$$

with the coefficient c is an strict positive real number and a_n are real numbers.

For example, $2 \cdot x_1^{1.5} x_2^{-1/3}$ is a monomial.

Definition 4.2 (Signomial functions) *A signomial $f(\mathbf{x})$ is a function which takes the form,*

$$f(\mathbf{x}) = \sum_{m=1}^M \left(c_m \cdot \prod_{n=1}^N x_n^{a_{m,n}} \right), \quad (4.2)$$

with c_m and $a_{m,n}$ are real numbers.

Signomials can be thought as the multivariable version of polynomials, and a signomial is called a posynomial if all coefficients c_m are positive, which imply a posynomial is the sum of monomials. It is easy to deduce that any signomial can be written as the difference between two posynomials. Posynomials are also closed under addition, multiplication, and nonnegative scaling. If a posynomial is multiplied or divided by a monomial, the result is a posynomial.

A GP is an optimization problem of the following form

Problem 4.1 (GP)

$$\min_{\mathbf{x}} f_0(\mathbf{x}) \quad (4.3)$$

$$s.t. \quad f_i(\mathbf{x}) \leq 1, \quad i \in \{1, \dots, p_1\} \quad (4.4)$$

$$g_i(\mathbf{x}) = 1, \quad i \in \{1, \dots, p_2\} \quad (4.5)$$

where $f_i(\mathbf{x})$ are posynomials and $g_i(\mathbf{x})$ are monomials.

GP problems are not generally convex. However, solving a GP is much easier comparing to other nonlinear programming as they can be converted into convex optimization problem by logarithmic transformation of the objective function and constraints, as well as by the change of variables. The positivity assumption on \mathbf{x} , as well as on $f_i(\mathbf{x})$ and $g_i(\mathbf{x})$, guarantees that this logarithmic transformation is valid. The converted problem has the form

Problem 4.2

$$\min_{\mathbf{y}} \log f_0(e^{\mathbf{y}}) \quad (4.6)$$

$$s.t. \quad \log f_i(e^{\mathbf{y}}) \leq 1, \quad i \in \{1, \dots, p_1\} \quad (4.7)$$

$$\log g_i(e^{\mathbf{y}}) = 1, \quad i \in \{1, \dots, p_2\} \quad (4.8)$$

where $y_n = \log x_n$, $\mathbf{y} = (y_1, \dots, y_N)$, and $e^{\mathbf{y}}$ is element-wise exponentiation.

The new problem is convex due to that fact that $\log f(e^{\mathbf{y}})$ is convex if f is posynomials. Therefore, the problem can be solved efficiently. As the exact method for solving a GP is not the focus of this thesis, we would not delve into detail on the GP solvers. Numerous tutorials and programs on this topic can be found in [94]–[97]. We will explore an extension of GP in the next section.

4.2.2 Signomial Geometric Programming (SGP)

SGP is a extension of GP, which contains signomial inequality constraints and signomial equality constraints. A SGP yields the following form:

Problem 4.3 (SGP)

$$\min_{\mathbf{x}} f_0(\mathbf{x}) \quad (4.9)$$

$$s.t. \quad f_i(\mathbf{x}) \leq 1, \quad i \in \{1, \dots, p_1\}, \quad (4.10)$$

$$f_i(\mathbf{x}) = 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.11)$$

where $f_i(\mathbf{x})$ are signomials.

We recall that each signomial can be rewritten as the difference between two posynomials. It follows that $\forall i \in \{1, \dots, p_2\}$, we have $f_i(\mathbf{x}) = f_i^+(\mathbf{x}) - f_i^-(\mathbf{x})$, where both $f_i^+(\mathbf{x})$ and $f_i^-(\mathbf{x})$ are posynomials. The original SGP has the equivalent form

Problem 4.4

$$\min_{\mathbf{x}} f_0^+(\mathbf{x}) - f_0^-(\mathbf{x}) \quad (4.12)$$

$$s.t. \quad f_i^+(\mathbf{x}) - f_i^-(\mathbf{x}) \leq 1, \quad i \in \{1, \dots, p_1\} \quad (4.13)$$

$$f_i^+(\mathbf{x}) - f_i^-(\mathbf{x}) = 1, \quad i \in \{p_1 + 1, \dots, p_2\} \quad (4.14)$$

We introduce a variable x_0 to bound the original objective function as follows:

$$f_0^+(\mathbf{x}) - f_0^-(\mathbf{x}) + M \leq x_0 + 1,$$

where M is a sufficiently large constant ensuring that $f_0^+(\mathbf{x}) - f_0^-(\mathbf{x}) + M > 1$, which guarantees $x_0 > 0$. This allows us to express the objective function in a linear form and rewrite all constraints as ratios of posynomials. Consequently, the optimization problem can be reformulated in an equivalent form.

Problem 4.5

$$\min_{x_0, \mathbf{x}} x_0 \quad (4.15)$$

$$s.t. \quad \frac{f_0^+(\mathbf{x}) + M}{f_0^-(\mathbf{x}) + x_0 + 1} \leq 1, \quad (4.16)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} \leq 1, \quad i \in \{1, \dots, p_1\}, \quad (4.17)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} = 1, \quad i \in \{p_1 + 1, \dots, p_2\}. \quad (4.18)$$

The constraints in the above optimization problem involves ratios between two posynomials. This type of optimization problems is categorized as as Complementary Geometric Programming (CGP) [98], [99], which is nonconvex optimization problem. In [81], authors proposed a successive convexification framework that converts the original problem into a series of basic GP. It is shown in several experience that a local optimal is found efficiently by this framework. In each iteration, there are two important transformation for the original problem in this framework. First, auxiliary variables t_i are introduced to relax the equality constraints, and define $\mathbf{t} = (t_{p_1+1}, \dots, t_{p_2})$. The problem is rewritten as

Problem 4.6

$$\min_{x_0, \mathbf{x}, \mathbf{t}} \quad x_0 + \sum_{i=p_1+1}^{p_2} w_i \cdot t_i \quad (4.19)$$

$$s.t. \quad \frac{f_0^+(\mathbf{x}) + M}{f_0^-(\mathbf{x}) + x_0 + 1} \leq 1, \quad (4.20)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} \leq 1, \quad i \in \{1, \dots, p_1\}, \quad (4.21)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} \leq 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.22)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} \geq 1 - t_i, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.23)$$

$$0 \leq t_i \leq 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.24)$$

where w_i is sufficiently large weighting coefficient.

The goal is to find solutions with small \mathbf{t} , which is achieved by progressively increasing the weighting coefficients $\{w_i\}_{i=p_1+1}^{p_2}$ while solving the series of optimization problems. Meanwhile, to reduce the number of constraints on \mathbf{t} , variables t_i are replaced by $s_i = 1/(1 - t_i)$, and $\mathbf{s} = (s_{p_1+1}, \dots, s_{p_2})$. Then, we obtain

Problem 4.7

$$\min_{x_0, \mathbf{x}, \mathbf{s}} \quad x_0 + \sum_{i=p_1+1}^{p_2} w_i \cdot s_i \quad (4.25)$$

$$s.t. \quad \frac{f_0^+(\mathbf{x}) + M}{f_0^-(\mathbf{x}) + x_0 + 1} \leq 1, \quad (4.26)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} \leq 1, \quad i \in \{1, \dots, p_1\}, \quad (4.27)$$

$$\frac{f_i^+(\mathbf{x})}{f_i^-(\mathbf{x}) + 1} \leq 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.28)$$

$$\frac{s_i^{-1}(f_i^-(\mathbf{x}) + 1)}{f_i^+(\mathbf{x})} \leq 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.29)$$

$$s_i^{-1} \leq 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.30)$$

The second transformation is to replace all denominators in the constraints by its best local monomial approximation with arithmetic geometric mean approximation [96]. For a posynomial $f(\mathbf{x}) = \sum_m g_m(\mathbf{x})$ with terms $g_k(\mathbf{x})$ being monomials, the best local monomial approximation around point \mathbf{y} is

$$\hat{f}_{\mathbf{y}}(\mathbf{x}) = \prod_m \left(\frac{g_m(\mathbf{x})}{\beta_m(\mathbf{y})} \right)^{\beta_m(\mathbf{y})}, \quad (4.31)$$

with $\beta_m(\mathbf{y}) \triangleq g_m(\mathbf{y})/f(\mathbf{y})$. Finally, we derive the GP approximation of the original problem as follows:

Problem 4.8 (GP Approximation)

$$\min_{x_0, \mathbf{x}, \mathbf{s}} \quad x_0 + \sum_{i=p_1+1}^{p_2} w_i \cdot s_i \quad (4.32)$$

$$s.t. \quad \frac{f_0^+(\mathbf{x}) + M}{\hat{f}_{0,\mathbf{y}}^-(\mathbf{x}, x_0)} \leq 1, \quad (4.33)$$

$$\frac{f_i^+(\mathbf{x})}{\hat{f}_{i,\mathbf{y}}^-(\mathbf{x})} \leq 1, \quad i \in \{1, \dots, p_2\}, \quad (4.34)$$

$$\frac{s_i^{-1}(f_i^-(\mathbf{x}) + 1)}{\hat{f}_{i,\mathbf{y}}^+(\mathbf{x})} \leq 1, \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.35)$$

$$s_i^{-1} \leq 1 \quad i \in \{p_1 + 1, \dots, p_2\}, \quad (4.36)$$

where $\hat{f}_{i,\mathbf{y}}^+(\mathbf{x})$ and $\hat{f}_{i,\mathbf{y}}^-(\mathbf{x})$ are monomial approximations for the corresponding denominators calculated by (4.31), and $\hat{f}_{0,\mathbf{y}}^-(\mathbf{x}, x_0)$ is the monomial approximation for the function $f_0^-(\mathbf{x}) + x_0 + 1$.

The entire process of the successive convexification framework is summarized in Algorithm 1. As the iterative algorithm moves toward the final point, the increasing weighting coefficient w_i will force the auxiliary variables s_i to reach one. Thus the series of solutions of Problem 4.8 converge to a point satisfying the KKT conditions of the original SGP problem [81], [100].

Algorithm 1: Successive convexification framework for solving SGP

Data: $p_1, p_2, \{f_i\}_{i \in \{0, \dots, p_2\}}, w_{step} \geq 0, \epsilon \geq 0$

Result: Optimized solution \mathbf{x}^*

Find a feasible solution of problem \mathbf{x}_1 ;

$r \leftarrow 1$;

$w_i \leftarrow 1$ for $i \in \{p_1 + 1, \dots, p_2\}$;

$E \leftarrow \epsilon + 1$;

Rewrite optimization problem into the form of Problem 4.7;

while $E \geq \epsilon$ **do**

$r \leftarrow r + 1$;

$w_i \leftarrow w_i + w_{step}$ for $i \in \{p_1 + 1, \dots, p_2\}$;

Calculate monomial approximations $\hat{f}_{0,\mathbf{x}_{r-1}}^-(\mathbf{x}, x_0)$, $\hat{f}_{i,\mathbf{x}_{r-1}}^+(\mathbf{x})$ and $\hat{f}_{i,\mathbf{x}_{r-1}}^-(\mathbf{x})$ for denominators in Problem 4.7;

Find the optimal solution of GP Problem 4.8;

$E \leftarrow \|\mathbf{x}_r - \mathbf{x}_{r-1}\|_2$;

$\mathbf{x}^* \leftarrow \mathbf{x}_r$;

end

4.3 System Model

We consider a distributed estimation system with K sensors and one single FC. The goal of the system is to estimate a scalar parameter θ . Sensor $k \in \{1, \dots, K\}$ collects a noisy

sample y_k . This sample is transmitted after a quantization process with B bits (described later). The FC so receives KB bits. We assume that the transmission channels between the sensors and the FC are reliable.

For the sake of simplicity, we consider the following model between θ and y_k :

$$y_k = \theta + w_k, \quad (4.37)$$

where w_k is a zero-mean white Gaussian noise with variance σ_w^2 and θ is assumed to be within the finite support $\mathcal{I} \subset \mathbb{R}$.

The output of the quantizer Q_k at sensor k , denoted by q_k , is

$$q_k = Q_k(y_k). \quad (4.38)$$

This quantizer Q_k transforms the continuous scalar y_k into B bits in such a way

$$Q_k(u) \triangleq \begin{cases} 0 & \text{if } u < \tau_{k,1}, \\ L & \text{if } u \geq \tau_{k,L}, \\ i & \text{if } \tau_{k,i} \leq u < \tau_{k,i+1}, \text{ for } i \in \{1, \dots, L-1\}. \end{cases}$$

where $L = 2^B - 1$ and $\boldsymbol{\tau}_k \triangleq (\tau_{k,1}, \dots, \tau_{k,L})$ is a strictly increasing sequence of thresholds.

The aim of this section of the thesis is to establish an effective method for constructing a threshold sequence for each sensor. To achieve this, we base our approach on the derivation of the CRB for θ at the FC.

4.4 Non-asymptotic Cramer-Rao Bound

The Cramer-Rao Bound for any unbiased estimation at the FC can be written as

$$\mathbb{E}[(\hat{\theta} - \theta)^2] \geq \text{CRB}(\theta) = \frac{1}{F(\theta)}, \quad (4.39)$$

where $F(\theta)$ is the Fisher information associated with quantized bits $\mathbf{q} \triangleq \{q_k\}_{k \in \{1, \dots, K\}}$ and writes as

$$F(\theta) \triangleq \mathbb{E}_{\mathbf{Q}|\theta} \left[\left(\frac{\partial \log p_{\mathbf{Q}|\theta}(\mathbf{q}|\theta)}{\partial \theta} \right)^2 \right] \quad (4.40)$$

with $p_{\mathbf{Q}|\theta}$ the distribution of \mathbf{q} for the parameter value θ . As q_k is independent in k , we have

$$F(\theta) = \sum_{k=1}^K F_k(\theta), \quad (4.41)$$

where $F_k(\theta)$ is the Fisher information provided by one quantized sample for the k -th sensor and

$$F_k(\theta) \triangleq \mathbb{E}_{Q_k|\theta} \left[\left(\frac{\partial \log p_{Q_k|\theta}(q_k|\theta)}{\partial \theta} \right)^2 \right]. \quad (4.42)$$

According to [48], the Fisher information can be expressed as

$$F_k(\theta) = \sum_{i=0}^L \frac{1}{p_{Q|\theta}(q_k = i|\theta)} \cdot \left(\frac{\partial p_{Q|\theta}(q_k = i|\theta)}{\partial \theta} \right)^2 \quad (4.43)$$

where $p_{Q|\theta}(q_k = i|\theta)$ can be decomposed as follows

$$p_{Q|\theta}(q_k = i|\theta) = \int p_{Q|Y}(q_k = i|y_k) p_{Y|\theta}(y_k|\theta) dy_k \quad (4.44)$$

since the value of q_k depends only on y_k according to Eq. (4.38). As $y_k|\theta$ follows a Gaussian distribution with mean θ and variance σ_w^2 , we deduce

$$p_{Q|\theta}(q_k = i|\theta) = \begin{cases} \Psi(\tau_{k,1}, \theta) & \text{if } i = 0 \\ 1 - \Psi(\tau_{k,L}) & \text{if } i = L \\ \Psi(\tau_{k,i+1}, \theta) - \Psi(\tau_{k,i}) & \text{otherwise.} \end{cases}$$

where

$$\Psi(\tau, \theta) \triangleq \Phi\left(\frac{\tau - \theta}{\sigma_w}\right) \quad (4.45)$$

with Φ the cumulative distribution function for standard normal distribution.

Then, we have

$$\frac{\partial p_{Q|\theta}(q_k = i|\theta)}{\partial \theta} = \begin{cases} -\psi(\tau_{k,1}, \theta) & \text{if } i = 0, \\ \psi(\tau_{k,L}, \theta) & \text{if } i = L, \\ -\psi(\tau_{k,i+1}, \theta) + \psi(\tau_{k,i}, \theta) & \text{otherwise.} \end{cases}$$

where

$$\psi(\tau, \theta) \triangleq \frac{1}{\sigma_w} \phi\left(\frac{\tau - \theta}{\sigma_w}\right) \quad (4.46)$$

with ϕ the probability density function for standard normal distribution.

Finally, we obtain that

$$F_k(\theta) = \eta_1(\tau_{k,1}, \theta) + \eta_L(\tau_{k,L}, \theta) + \sum_{i=1}^{L-1} \eta(\tau_{k,i}, \tau_{k,i+1}, \theta) \quad (4.47)$$

with

$$\eta_1(\tau, \theta) \triangleq \frac{(\psi(\tau - \theta))^2}{\Psi(\tau - \theta)}, \quad (4.48)$$

$$\eta_L(\tau, \theta) \triangleq \frac{(\psi(\tau - \theta))^2}{1 - \Psi(\tau - \theta)}, \quad (4.49)$$

$$\eta(\tau, \tau', \theta) \triangleq \frac{(\psi(\tau' - \theta) - \psi(\tau - \theta))^2}{\Psi(\tau' - \theta) - \Psi(\tau - \theta)}. \quad (4.50)$$

4.5 Asymptotic Cramer-Rao Bound

By considering large number of sensors, the CRB can be approximated by

$$\text{CRB}(\theta) \approx \frac{1}{K \cdot \underline{F}(\theta)} \quad (4.51)$$

where

$$\underline{F}(\theta) \triangleq \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K F_k(\theta) \quad (4.52)$$

$$= \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K \left(\eta_1(\tau_{k,1}, \theta) + \eta_L(\tau_{k,L}, \theta) + \sum_{i=1}^{L-1} \eta(\tau_{k,i}, \tau_{k,i+1}, \theta) \right). \quad (4.53)$$

Like in [59], the evaluation of the term $\underline{F}(\theta)$ will be done by assuming that the thresholds $\{\tau_{k,i}\}_{k,i}$ correspond to a realization of a random variable. The realizations on k are iid but the realizations on i have to be ranked since $\tau_{k,i} \leq \tau_{k,i+1}$ by construction. Therefore, we rely on the order statistics [101]. Let λ be the probability distribution for the L ordered threshold for any sensor (the distribution is assumed to be the same whatever the sensor, so we skip the index k). We also define the marginal probability distribution for the first threshold (actually, $\tau_{k,1}$ for any sensor) as λ_1 , the marginal probability distribution for the last threshold (actually, $\tau_{k,L}$ for any sensor) as λ_L . Finally, the joint probability distribution between two consecutive thresholds (actually, $(\tau_{k,i}, \tau_{k,i+1})$ for any sensor with $i \in \{1, \dots, L-1\}$) as λ_i . We thus obtain

$$\begin{aligned} \underline{F}(\theta) &= \int_{-\infty}^{\infty} \eta_1(\tau, \theta) \lambda_1(\tau) d\tau + \int_{-\infty}^{\infty} \eta_L(\tau, \theta) \lambda_L(\tau) d\tau \\ &+ \sum_{i=1}^{L-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\tau'} \eta(\tau, \tau', \theta) \lambda_i(\tau, \tau') d\tau d\tau'. \end{aligned} \quad (4.54)$$

The ordered thresholds are obtained thanks to a unique random variable whose probability density function is g and the cumulative distribution function G . More precisely, for each sensor, we collect L realizations related to the random variable driven by g . Then these variables are ranked in order to provide the ordered thresholds. Consequently, the distributions for the ranked variables are given by

$$\lambda_1(\tau) = c[1 - G(\tau)]^{L-1} g(\tau), \quad (4.55)$$

$$\lambda_L(\tau) = c[G(\tau)]^{L-1} g(\tau), \quad (4.56)$$

$$\lambda_i(\tau, \tau') = c_i [G(\tau)]^{i-1} [1 - G(\tau')]^{L-i} g(\tau) g(\tau'), \quad (4.57)$$

with $c = L$ and $c_i = L(L-1) \binom{L-2}{i-1}$.

4.5.1 Optimized Quantizer with SGP

Our goal is to minimize the worst cast asymptotic CRB with respect to g . According to Eq. (4.51), minimizing the CRB is equivalent to maximizing \underline{F} . Therefore by integrating Eqs. (4.55)-(4.57) into Eq. (4.54) and over the infimum of θ , we obtain the following minimax optimization problem.

Problem 4.9 (Functional optimization problem) Assuming $g(\tau) \geq 0$, we solve

$$\max_g \inf_{\theta} f(g, \theta)$$

s.t. $\int g(\tau) d\tau = 1$, with

$$\begin{aligned} f(g, \theta) = & \int \eta_1(\tau, \theta) \cdot c[1 - G(\tau)]^{L-1} g(\tau) d\tau + \int \eta_L(\tau, \theta) \cdot c[G(\tau)]^{L-1} g(\tau) d\tau \\ & + \sum_{i=1}^{L-1} \iint^{\tau'} \eta(\tau, \tau', \theta) \cdot c_i [G(\tau)]^{i-1} \cdot [1 - G(\tau')]^{L-i} g(\tau) g(\tau') d\tau d\tau'. \end{aligned} \quad (4.58)$$

We propose to transform this functional optimization problem into a vector search by discretizing g . Before going further, we assume that the support of θ is symmetric about the origin, i.e. $\mathcal{I} = [-W_0, W_0]$, with $W_0 > 0$. Then, we discretize the interval \mathcal{I} into $N (> 0)$ regular subintervals $\{\mathcal{J}_j \triangleq [u_{j-1}, u_j]\}_{j \in \{1, \dots, N\}}$ with $u_0 = -W_0$ and $u_N = W_0$ whose the middle of each interval is m_j for $j \in \{1, \dots, N\}$. The discrete version of g at m_j is given by its normalized value i.e.,

$$a_j = \frac{g(m_j)}{\sum_{j'=1}^N g(m_{j'})} \quad \forall j \in \{1, \dots, N\}. \quad (4.59)$$

We now define the cumulative sequence of $\mathbf{a} \triangleq \{a_j\}_{j \in \{1, \dots, N\}}$ as

$$A_\ell = \sum_{j=1}^{\ell} a_j,$$

and the complement cumulative sequence as $R_\ell = 1 - A_\ell$. By convention, we also put $A_0 = 0$. Define also the sequence $\{\theta_j\}_{j \in \{1, \dots, M\}}$ with $\theta_1 = -W_0$ and $\theta_M = W_0$ representing a quantization of the parameter range.

We discretize the function $f(g, \theta)$ into a set of functions $\tilde{f}_j(\mathbf{a})$ as follows

$$\begin{aligned} \tilde{f}_j(\mathbf{a}) = & \sum_{\ell=1}^N d_{\ell,j}^{(1)} R_\ell^{L-1} a_\ell + \sum_{\ell=1}^N d_{\ell,j}^{(L)} A_\ell^{L-1} a_\ell \\ & + \sum_{i=1}^{L-1} \sum_{\ell_2=1}^N \sum_{\ell_1=1}^{\ell_2} d_{\ell_1, \ell_2, j}^{(i)} A_{\ell_1}^{i-1} a_{\ell_1} R_{\ell_2}^{L-i} a_{\ell_2} \end{aligned} \quad (4.60)$$

with $d_{\ell,j}^{(1)} = c \cdot \eta_1(u_\ell, \theta_j)$, $d_{\ell,j}^{(L)} = c \cdot \eta_L(u_\ell, \theta_j)$, and $d_{\ell_1, \ell_2, j}^{(i)} = c_i \cdot \eta(u_{\ell_1}, u_{\ell_2}, \theta_j)$. The discretized version of Problem 4.9 is straightforward. But in terms of optimization, we have the following objective function to maximize $\min_j \tilde{f}_j(\mathbf{a})$. In order to handle the minimum operator easily, we introduce a new variable x bounding all functions $\tilde{f}_j(\mathbf{a})$. We then obtain the following optimization problem.

Problem 4.10 (Discretized optimization problem) Assuming $a_\ell \geq 0$ for $\ell \in \{1, \dots, N\}$ and $x \geq 0$, we solve

$$\max_{\mathbf{a}, x} x \quad (4.61a)$$

$$\text{s.t. } \tilde{f}_j(\mathbf{a}) \geq x \quad \forall j \in \{1, \dots, M\}, \quad (4.61b)$$

$$\sum_{\ell=1}^N a_\ell = 1. \quad (4.61c)$$

We can find a stationary point of Problem 4.10 by SGP. However, as we mentioned in Section 4.2.2, extra variables are needed to relax equality constraints in the original optimization problem. We thus change the optimization variables from \mathbf{a} to $\mathbf{A} \triangleq \{A_1, A_2, \dots, A_{N-1}\}$ (A_0 and A_N are excluded from the variable vector as they are always equal to 0 and 1 respectively) to bypass the equality constraint (4.61c). Functions in (4.60) are rewritten as

$$f_j(\mathbf{A}) = \sum_{\ell=1}^N d_{\ell,j}^{(1)} R_{\ell}^{L-1} (A_{\ell} - A_{\ell-1}) + \sum_{\ell=1}^N d_{\ell,j}^{(L)} A_{\ell}^{L-1} (A_{\ell} - A_{\ell-1}) \\ + \sum_{i=1}^{L-1} \sum_{\ell_2=1}^N \sum_{\ell_1=1}^{\ell_2} d_{\ell_1, \ell_2, j}^{(i)} A_{\ell_1}^{i-1} (A_{\ell_1} - A_{\ell_1-1}) \times R_{\ell_2}^{L-i} (A_{\ell_2} - A_{\ell_2-1}). \quad (4.62)$$

The function f_j is a signomial and can be written as a difference of two posynomials [102]. Let's write this decomposition below

$$f_j(\mathbf{A}) \triangleq f_{j,d}(\mathbf{A}) - f_{j,n}(\mathbf{A}), \quad (4.63)$$

where $f_{j,d}$ and $f_{j,n}$ are two posynomials.

Thanks to Eqs. (4.62)-(4.63), Problem 4.10 can be rewritten as Problem 4.11 which is a SGP and so can be solved easily.

Problem 4.11 (Final optimization problem) *Assuming $A_{\ell} \geq 0$ for $\ell \in \{1, \dots, N\}$ and $x \geq 0$, we solve*

$$\min_{\mathbf{A}, x} x^{-1} \quad (4.64a)$$

$$\text{s.t. } \frac{x + f_{j,n}(\mathbf{A})}{f_{j,d}(\mathbf{A})} \leq 1, \forall j \in \{1, \dots, M\}, \quad (4.64b)$$

$$\frac{A_{\ell-1}}{A_{\ell}} \leq 1, \quad \forall \ell \in \{2, \dots, N-1\}, \quad (4.64c)$$

$$A_{\ell} \leq 1, \quad \forall \ell \in \{1, \dots, N-1\}. \quad (4.64d)$$

This optimization problem can be solved by the recurrence algorithm given in Algorithm 1. A more detailed description of the algorithm specified for Problem 4.10 is provided below.

- *Initialization* Choose $A_i^{(0)} = i/n, \forall i \in [n-1]$. Calculate $x^{(0)} = \min_j \{\tilde{f}_j(\mathbf{A}^{(0)})\}$ and $x_0^{(0)} = -x^{(0)} + C$. Write each functions $\tilde{f}_{j,d}(\mathbf{A})$ into a sum of monomial functions $\{m_{j,i}(\mathbf{A})\}_i$, i.e.

$$\tilde{f}_{j,d}(\mathbf{A}) = \sum_i m_{j,i}(\mathbf{A}) \quad (4.65)$$

- *Step 1* At round $r \geq 1$, for each $j \in [M]$, calculate

$$\beta^{(r)} = \frac{x^{(r-1)}}{x^{(r-1)} + x_0^{(r-1)}}, \quad (4.66)$$

$$\beta_0^{(r)} = \frac{x_0^{(r-1)}}{x^{(r-1)} + x_0^{(r-1)}}, \quad (4.67)$$

$$\beta_{j,i}^{(r)} = \frac{m_{j,i}(\mathbf{A}^{(r-1)})}{\tilde{f}_{j,d}(\mathbf{A}^{(r-1)})}, \quad (4.68)$$

- *Step 2* Solve the following optimization problem with GP by replacing functions on the denominator in Problem 4.11 by its local monomial approximation.

$$\min_{\mathbf{A}, x_0, x} x_0 \quad (4.69a)$$

$$\text{s.t.} \quad \frac{C}{\hat{f}^{(r)}(x, x_0)} \leq 1 \quad (4.69b)$$

$$\frac{x + \hat{f}_{j,n}(\mathbf{A})}{\hat{f}_{j,d}^{(r)}(\mathbf{A})} \leq 1, \quad \forall j \in [M] \quad (4.69c)$$

$$\frac{A_{\ell-1}}{A_\ell} \leq 1, \quad \forall \ell \in [N-1] \setminus \{1\} \quad (4.69d)$$

$$0 \leq A_\ell \leq 1, \quad \forall \ell \in [N-1]. \quad (4.69e)$$

where

$$\hat{f}^{(r)}(x, x_0) \triangleq (x/\beta^{(r)})^{\beta^{(r)}} \cdot (x_0/\beta_0^{(r)})^{\beta_0^{(r)}}, \quad (4.70)$$

$$\hat{f}_{j,d}^{(r)}(\mathbf{A}) \triangleq \prod_i \left(\frac{m_{j,i}(\mathbf{A})}{\beta_{j,i}^{(r)}} \right)^{\beta_{j,i}^{(r)}}, \quad (4.71)$$

- *Step 3* Repeat Step 1 and Step 2 until the required precision is achieved.

Once Problem 4.11 is solved, we obtain the points $\{a_\ell^*\}_{\ell=1,\dots,N}$. Then we compute $g(\tau)$ by doing a Lagrangian-polynomial interpolation on the set $\{(m_\ell, a_\ell^*)\}_{\ell=1,\dots,N}$ followed by a normalization step. Finally, each sensor builds its quantizer as follows: it obtains $(L-1)$ realizations from the random variable whose distribution is $g(\tau)$. Each sensor sorts its realizations to be employed as its thresholds.

4.5.2 Integral-preserving Discretization

Note that Problem 4.11 includes fractional posynomial constraints in (4.64b), which increases the complexity of the problem. These fractional posynomial constraints arise when converting the maximization problem in Problem 4.10 into the minimization problem in Problem 4.11. This conversion inevitably introduces negative terms, regardless of whether variables \mathbf{a} or \mathbf{A} are applied. In this subsection, we propose an alternative discretization method as an attempt to overcome this challenge.

To gain insight, we revisit the objective function in (4.58) and observe that constants can be added to the coefficients without altering the optimization problem. This is because λ_1 , λ_L , and λ_i represent a probability distribution (as shown in (4.55)-(4.57)). For instance, when $L=3$, (4.58) can be rewritten into four terms, where the second term is:

$$\int_{\tau} \eta_3(\tau, \theta) \cdot 3[G(\tau)]^2 g(\tau) d\tau.$$

The optimization problem remains unchanged if this term is replaced by:

$$\int_{\tau} (\eta_3(\tau, \theta) + c_0) \cdot 3[G(\tau)]^2 g(\tau) d\tau,$$

where c_0 is a constant. All negative coefficients arising from converting the maximization problem into a minimization problem can be transformed into positive ones by adding a sufficiently large constant. We would like the discrete analogue of $f(g, \theta)$ still conserve this integral property so that we are allowed to adjust coefficients.

Discrete analogue related to λ_L We first would like to discretize the term

$$T_L \triangleq \int \eta_L(\tau, \theta) \cdot c[G(\tau)]^{L-2} g(\tau) d\tau.$$

Actually, on T_L , we have two integrals since G is an integral of g . For calculating T_L , we may directly discretize both integrals with two discretization uniform schemes and then

$$\hat{T}_L \triangleq \sum_{\ell=1}^N d_{\ell}^{(L)} A_{\ell}^{L-2} a_{\ell} \quad (4.72)$$

with $d_{\ell,j}^{(L)} = c \cdot \eta_L(u_{\ell}, \theta_j)$, and we recall that the sequences $\{u_{\ell}\}$, $\{a_{\ell}\}$, $\{A_{\ell}\}$ and $\{\theta_j\}$ are defined in Section 4.5.1. As shown in Fig. 4.1, it requires a large amount of discretization levels N for the discrete analogue to converge. In the rest of paragraph, we attempt another way to proceed to work with smaller N and so less variables to optimize.

Let us focus on the case $\eta_L(\tau, j) = 1$ and so $d_{\ell,j}^{(L)} = 1$ for the explanation. Then, by applying integral by parts, we have $T_L = 1/(L-1)$. Usually the value of \hat{T}_L is different from $1/(L-1)$ when N is different from $+\infty$ due to the presence of the term A_{ℓ}^{L-1} . Therefore the idea is to modify the term related to the cumulative function by designing another one $\tilde{A}_{\ell}^{(L-2)}$ such that $\tilde{T}_L \triangleq \sum_{\ell=1}^N \tilde{A}_{\ell}^{(L-2)} a_{\ell} = 1/(L-1)$. Thanks to the so-called Abel's lemma on the sequences, we get

$$\sum_{\ell=1}^N A_{\ell-1} (A_{\ell}^{L-2} - A_{\ell-1}^{L-2}) + \sum_{\ell=1}^N a_{\ell} A_{\ell}^{L-2} = A_N A_{N+1} - A_0^2$$

with $A_0 = 0$ and $A_N = A_{N+1} = 1$. Consequently, by regrouping the sums and dividing by $L-1$, we obtain

$$\sum_{\ell=1}^N \tilde{A}_{\ell}^{(L-2)} a_{\ell} = \frac{1}{L-1} \quad (4.73)$$

with

$$\tilde{A}_{\ell}^{(L-2)} = \frac{1}{L-1} \sum_{i=0}^{L-2} A_{\ell}^{L-2-i} A_{\ell-1}^i.$$

Therefore, we have $\int G(\tau)^{L-2} g(\tau) d\tau \approx \sum_{\ell=1}^N \tilde{A}_{\ell}^{(L-2)} a_{\ell}$ and by extension we consider that

$$T_L \approx \sum_{\ell=1}^N d_{\ell}^{(L)} \tilde{A}_{\ell}^{(L-2)} a_{\ell}. \quad (4.74)$$

Discrete analogue related to λ_1 Similarly, we would like to discretize

$$T_1 \triangleq \int \eta_1(\tau, \theta) [1 - G(\tau)]^{L-2} g(\tau) d\tau.$$

By following the same approach, we obtain that

$$T_1 \approx \sum_{\ell=1}^N d_{\ell,j}^{(1)} \tilde{R}_{\ell}^{(L-2)} a_{\ell} \quad (4.75)$$

with $d_{\ell,j}^{(1)} = \eta_1(u_{\ell}, \theta_j)$ and

$$\tilde{R}_{\ell}^{(L-2)} \triangleq \frac{1}{L-1} \sum_{i=0}^{L-2} R_{\ell}^{L-2-i} R_{\ell-1}^i.$$

Discrete version related to λ_i We now focus on the following term which is more complicated than the previous ones due to the double integrals.

$$T_i = \iint^{\tau'} \eta(\tau, \tau', \theta) \cdot c_i [G(\tau)]^{i-2} \cdot [1 - G(\tau')]^{L-i-1} g(\tau) g(\tau') d\tau d\tau'$$

To provide an insight on how discretizing the previous integrals in T_i , we look at the term with the integral on τ and without C_i . Therefore, we consider

$$U_i = \int^{\tau'} [G(\tau)]^{i-2} [1 - G(\tau')]^{L-i-1} g(\tau) g(\tau') d\tau. \quad (4.76)$$

After integration by parts, we have

$$U_i = [G(\tau')^{i-1} - \sum_{j=1}^{L-i-1} \binom{L-i-1}{j} (-1)^{i+j} G(\tau')^{j+i-1}] g(\tau'),$$

which leads to the following discrete version by replacing $G(\tau')^i$ and $g(\tau')$ with $\tilde{A}_\ell^{(i)}$ and a_ℓ respectively

$$U_i(\ell) \approx [\tilde{A}_\ell^{(i-1)} - \sum_{j=1}^{L-i-1} \binom{L-i-1}{j} (-1)^{j+i} \tilde{A}_\ell^{(j+i-1)}] a_\ell,$$

or equivalently,

$$U_i(\ell) \approx \tilde{A}_\ell^{(i-1)} \tilde{R}_\ell^{(L,i)} a_\ell,$$

with

$$\tilde{R}_\ell^{(L,i)} \triangleq 1 - \sum_{j=1}^{L-i-1} \binom{L-i-1}{j} (-1)^{j+i} \frac{\tilde{A}_\ell^{(j+i-1)}}{\tilde{A}_\ell^{(i-1)}}. \quad (4.77)$$

We can deduce that the sum of the discrete version of $U_i(\ell)$ over ℓ is a constant, since it can be written as the sum of terms $\sum_\ell \tilde{A}_\ell^{(i-1)} a_\ell$ which are constants as we have shown in (4.73).

For the rest of the derivations, we prefer to write $U_i(\ell)$ as a sum since it should mimic an integral. For doing that, we sum increments of the primitive function. Therefore, we have

$$\begin{aligned} U_i(\ell) &= \sum_{\ell_2=1}^{\ell} (\tilde{A}_{\ell_2}^{(i-1)} \tilde{R}_{\ell_2}^{(L,i)} a_{\ell_2} - \tilde{A}_{\ell_2-1}^{(i-1)} \tilde{R}_{\ell_2}^{(L,i)} a_{\ell_2}) \\ &= \sum_{\ell_2=1}^{\ell} [\tilde{A}_{\ell_2}^{(i-1)} - \tilde{A}_{\ell_2-1}^{(i-1)}] \tilde{R}_{\ell_2}^{(L,i)} a_{\ell_2}. \end{aligned} \quad (4.78)$$

By going back to the approximation of T_i , we finally obtain

$$T_i = \sum_{\ell_1=1}^N \sum_{\ell_2=1}^{\ell_1} d_{\ell_1, \ell_2}^{(i)} U_i(\ell_1) \quad (4.79)$$

with $d_{\ell_1, \ell_2, j}^{(i)} = c_i \cdot \eta(u_{\ell_1}, u_{\ell_2}, \theta_j)$ and $U_i(\ell_1)$ given by Eq. (4.78).

Final result Plugging Eqs. (4.74), (4.75) and (4.79) into Eq. (4.58) leads to the discrete optimization problem which becomes the main goal of the paper.

Problem 4.12 (Equivalent vector optimization problem)

$$\min_{\mathbf{a}} \max_j \tilde{f}_j(\mathbf{a})$$

s.t. $a_\ell \geq 0$, and $\sum_{\ell=1}^N a_\ell = 1$, with

$$\begin{aligned} \tilde{f}(\mathbf{a}) = & \sum_{\ell=1}^N \gamma_\ell^{(1)} \tilde{R}_\ell^{(L-2)} a_\ell + \sum_{\ell=1}^N \gamma_\ell^{(L)} \tilde{A}_\ell^{(L-2)} a_\ell \\ & + \sum_{i=2}^{L-1} \sum_{\ell_1=1}^N \sum_{\ell_2=1}^{\ell_1} \gamma_{\ell_1, \ell_2}^{(i)} \left(\tilde{A}_{\ell_1}^{(i-1)} - \tilde{A}_{\ell_1-1}^{(i-1)} \right) \tilde{R}_{\ell_2}^{(L,i)} a_{\ell_2} \end{aligned} \quad (4.80)$$

where $\gamma_\ell^{(1)} \triangleq -d_\ell^{(1)} + c_0$, $\gamma_\ell^{(L)} \triangleq -d_\ell^{(L)} + c_0$, $\gamma_{\ell_1, \ell_2}^{(i)} \triangleq -d_{\ell_1, \ell_2}^{(i)} + c_0$ with c_0 chosen large enough s.t. $\gamma_\ell^{(1)}, \gamma_\ell^{(L)}, \gamma_{\ell_1, \ell_2}^{(i)}$ are strictly positive.

We can verify the improvement of our discretizing method over the intuitive method by a simple example. Choose $g(\tau) = 2\tau \cdot \mathbb{1}_{\{0 \leq \tau \leq 1\}}$, and $\int \eta_1(\theta, \tau) p_\Theta(\theta) d\theta = \tau$. It is easy to calculate accurately the integral

$$\int_{\tau} \left(\int_{\theta} \eta_1(\theta, \tau) d\theta \right) \lambda_L(\tau) p_\Theta(\theta) d\tau = \frac{2}{2L-1}. \quad (4.81)$$

The numerical results represented in Fig. 4.1 show that our discretizing method produces a better approximation than the intuitive discretization.

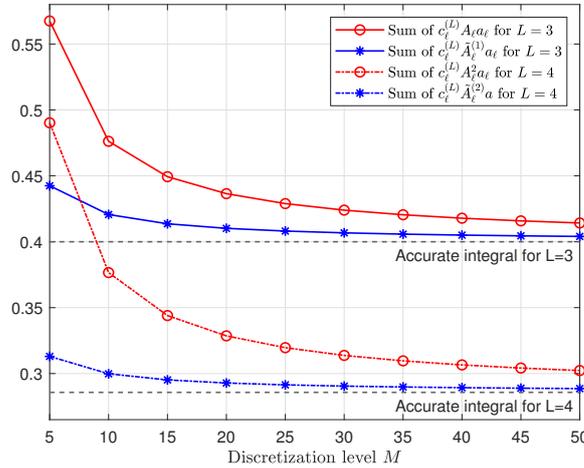


Figure 4.1: Comparison between the intuitive quantization and our method of quantization with $L = 3$ and $L = 4$

Unfortunately, it is not possible to eliminate all fractions when $L \geq 4$, as shown in (4.77). For $L = 3$ (and similarly for $L = 2$), we can derive an optimization problem without fractional polynomial constraints, using discrete analogues as follows:

$$\tilde{A}_\ell^{(2)} \triangleq \frac{1}{3} (A_{\ell-1}^2 + A_{\ell-1} A_\ell + A_\ell^2),$$

$$\tilde{R}_\ell^{(2)} \triangleq \frac{1}{3} (R_{\ell-1}^2 + R_{\ell-1} A_\ell + R_\ell^2).$$

However, since this discretization method does not simplify the optimization problem in general cases, we will not use it in our simulations.

4.6 Numerical Results

We assume that each sensor sends 2 bits ($L = 3$), and $M = 8$. We also assume the support of the parameter $\mathcal{I} = [-1, 1]$. Problem 4.11 has been numerically solved by algorithm described in [81]. Two cases for discretization problem have been computed: $N = 3$ or $N = 10$. We also examine the performance of the uniformly-distributed (i.e. $a_\ell = 1/N, \forall \ell \in \{1, \dots, N\}$) and the regular deterministic quantizers.

In Fig. 4.2, we plot the optimized continuous distribution $g(\tau)$ by doing a polynomial fitting of discrete solution for Problem 4.11. Two cases have been considered: $N = 3$ and $N = 10$. We observe that the solutions are different from the uniform distribution, which implies the necessity of the optimization. Moreover, $N = 3$ does not match with $N = 10$ exhibiting better performance, as shown later in Fig. 4.3 and Fig. 4.4. This implies that $N = 3$ is insufficient.

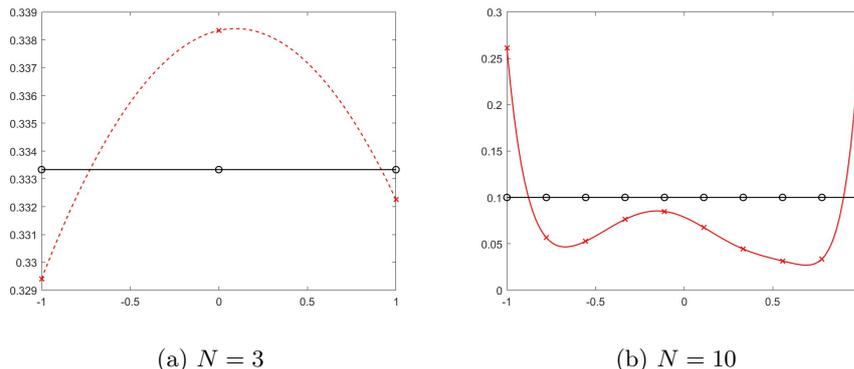
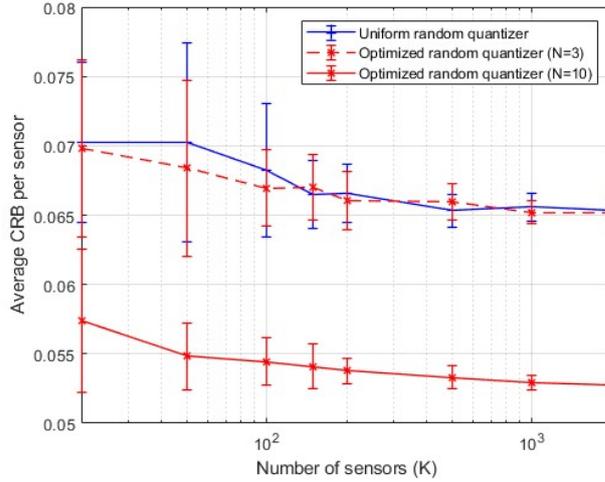


Figure 4.2: Optimized distribution $g(\tau)$ obtained by polynomial fitting based on discrete solution of Problem 4.11.

In Fig. 4.3, we plot the average CRB per sensor (obtained as the inverse of the average Fisher information $\min_{j'} \frac{1}{K} \sum_{k=1}^K F_k(\theta'_{j'})$ where the set $\{\theta'_{j'}\}_{j' \in \{1, \dots, M'\}}$ with $M' = 100$ covers the parameter range well, and actually much more than the set chosen for optimization) versus K . The SNR (defined as $1/\sigma_w^2$) is put at 8dB. For each sensor, we build its quantizer as follows: we obtain 3 realizations from the random variable whose distribution is $g(\tau)$. Then they are ranked to be employed as the three thresholds. This evaluation is executed 20 times for each configuration. Then we plot the average and the standard deviation of each configuration. We observe that even for small values of K , our optimization obtained through an asymptotic approach is still valid. The information provided by each sensor at the system is higher with the optimized version of the random quantizer. Only the standard deviation decreases with K .

We also compare our quantization method with existing works in [55], [56]. In [56], a heuristic algorithm for quantizer optimization in detection problem is proposed based on

Figure 4.3: Average CRB vs K (SNR=8dB).

Particle Swarm Optimization Algorithm (PSOA). This algorithm evaluates large amount of randomly generated quantizers to approximate locate the optimal quantizer. While the objective in [56] differs from ours, we utilize an adapted version of the original algorithm for comparison, making minimal changes. The PSOA gives an identical quantizer for all sensors, and authors in [56] assumed θ is around zero, the objective function is thus changed as

$$\max_{\tau_1 < \tau_2 < \dots < \tau_L} f_{psoa}(\tau_1, \dots, \tau_L) \quad (4.82)$$

with

$$f_{psoa}(\tau_1, \dots, \tau_L) \triangleq \eta_1(\tau_1, 0) + \eta_L(\tau_L, 0) + \sum_{i=1}^{L-1} \eta(\tau_i, \tau_{i+1}, 0). \quad (4.83)$$

Actually, this is the same Fisher information defined in (4.53), but with the assumptions that $\theta = 0$ and a deterministic quantizer is applied. We give a brief description of the algorithm. Initially, a sequence of thresholds τ_k^0 are independently initialized for each sensor according to a uniform distribution on the support $\mathcal{I} = [-1, 1]$. The initial velocity vector \mathbf{v}_k^0 are generated according to a uniform distribution in $[-1, 1]$. To follow the order constraint in (4.83), we reorder τ_k with the mechanism described as follows:

$$\text{If } \tau_{k,i-1} > \tau_{k,i}, \text{ then } \tau_{k,i} = \tau_{k,i} + \epsilon, \quad i \in \{2, \dots, L\}. \quad (4.84)$$

where ϵ is an arbitrary small positive real number. Based on the initialization $\{\tau_k^0\}_{k \in [K]}$, the initial personal best quantizer pbest_k^0 of the k -th sensor to be

$$\text{pbest}_k^0 = \tau_k^0. \quad (4.85)$$

Substituting the initial quantizer $\{\tau_k^0\}_{k \in [K]}$ into the objective function in (4.83), we obtain a set of values $\{f_{psoa}(\tau_k^0)\}_{k \in [K]}$, and then set the initial global best quantizer gbest^0 to be

$$\text{gbest}^0 = \arg \max_{\tau \in \{\tau_k^0\}_{k \in [K]}} \{f_{psoa}(\tau)\}. \quad (4.86)$$

At the t -th iteration, the velocity vector \mathbf{v}_k^t and quantizer τ_k^t of the k -th sensor is updated respectively as,

$$\mathbf{v}_k^t = c_0 [\mathbf{v}_k^{t-1} + c_1 r_{k,1}^t (\text{pbest}_k^{t-1} - \tau_k^{t-1}) + c_2 r_{k,2}^t (\text{gbest}^{t-1} - \tau_k^{t-1})], \quad (4.87)$$

and

$$\tau_k^t = \tau_k^{t-1} + \mathbf{v}_k^t, \quad (4.88)$$

where $r_{i,1}^k$ and $r_{i,2}^k$ are random numbers uniformly distributed within $[0, 1]$; the positive constants c_1 and c_2 represent the acceleration coefficients. The algorithm then reorders each quantizer with the same mechanism in (4.84), and updates personal best quantizer for this iteration, and evaluates the global best quantizer. The algorithm repeats this process until the norm of the velocity vector is sufficiently small. The entire algorithm is presented in Algorithm 2.

Algorithm 2: Particle Swarm Optimization Algorithm (PSOA) [56]

Input: $K, L, v_{tol}, \mathcal{I}, f_{psoa}(\boldsymbol{\tau}), c_0, c_1, c_2$
Result: Optimized solution $\boldsymbol{\tau}^* = \{\tau_1^*, \dots, \tau_L^*\}$
 Randomly initialize $\tau_k^0 \in \mathcal{I}^L$ and $\mathbf{v}_k^0 \in \mathcal{I}^L$ for $k \in [K]$;
 Reorder the initial quantizer τ_k^0 by (4.84);
 $\text{pbest}_k^0 \leftarrow \tau_k^0$;
 $\text{gbest}^0 \leftarrow \arg \max_{\boldsymbol{\tau} \in \{\tau_k^0\}_{k \in [K]}} \{f_{psoa}(\boldsymbol{\tau})\}$;
 $t \leftarrow 0$;
while $\|(\mathbf{v}_1^t, \dots, \mathbf{v}_K^t)\|_2 > v_{tol}$ **do**
 $t \leftarrow t + 1$;
 for $k \in [K]$ **do**
 Calculate the velocity \mathbf{v}_k^t by (4.87);
 $\tau_k^t \leftarrow \tau_k^{t-1} + \mathbf{v}_k^t$;
 Reorder the quantizer τ_k^t by (4.84);
 $\text{pbest}_k^t \leftarrow \arg \max_{\boldsymbol{\tau} \in \{\text{pbest}_k^{t-1}, \tau_k^t\}} \{f_{psoa}(\boldsymbol{\tau})\}$;
 end
 $\text{gbest}^t \leftarrow \arg \max_{\boldsymbol{\tau} \in \{\text{pbest}_k^t\}_{k \in [K]}} \{f_{psoa}(\boldsymbol{\tau})\}$;
end
 $\boldsymbol{\tau}^* \leftarrow \text{gbest}^t$;

In [55], the authors proposed a quantizer optimization algorithm based on dynamic programming called Interval Design for Enhanced Accuracy (IDEA)¹. The authors assumed that a prior knowledge of the parameter θ was available, and similar to our method, a grid (u_1, \dots, u_N) was applied to discretize the continuous feasible region of each threshold τ_i . The grid was assumed to be the same for all thresholds for simplicity.

We rewrite the algorithm with our notations. The objective is

$$\max_{\tau_1 < \tau_2 < \dots < \tau_L \in \{u_1, \dots, u_N\}} f_{dp}(\tau_1, \dots, \tau_L) \quad (4.89)$$

¹The authors proposed two algorithms in [55]. Here, we refer specifically to the algorithm presented in Section V.

with

$$f_{dp}(\tau_1, \dots, \tau_L) \triangleq \int_{\theta} \left(\eta_1(\tau_1, \theta) s(\theta) + \eta_L(\tau_L, \theta) s(\theta) + \sum_{i=1}^{L-1} \eta(\tau_i, \tau_{i+1}, \theta) s(\theta) \right) d\theta. \quad (4.90)$$

As each term only depends on at most two thresholds, it is straightforward to convert the optimization problem into a dynamic programming. To initiate, the following function is calculated

$$DP_1(\tau_1) \triangleq \int_{\theta} \eta_1(\tau_1, \theta) \cdot s(\theta) d\theta \quad (4.91)$$

For $i \in \{2, \dots, L-1\}$, the previous results $\{DP_{i-1}(\tau)\}_{\tau \in \{u_1, \dots, u_L\}}$ is used to calculate

$$DP_i(\tau_i) \triangleq \max_{\substack{\tau_{i-1} \in \{u_1, \dots, u_N\} \\ \tau_{i-1} < \tau_i}} \left\{ \int_{\theta} \eta(\tau_{i-1}, \tau_i, \theta) \cdot s(\theta) d\theta + DP_{i-1}(\tau_{i-1}) \right\} \quad (4.92)$$

Define also $\hat{\tau}_{i-1}(\tau_i)$ the solution for $DP_i(\tau_i)$, i.e.

$$\hat{\tau}_{i-1}(\tau_i) \triangleq \arg \max_{\substack{\tau_{i-1} \in \{u_1, \dots, u_N\} \\ \tau_{i-1} < \tau_i}} \left\{ \int_{\theta} \eta(\tau_{i-1}, \tau_i, \theta) \cdot s(\theta) d\theta + DP_{i-1}(\tau_{i-1}) \right\} \quad (4.93)$$

When $i = L$, calculate

$$DP_L(\tau_L) \triangleq \max_{\substack{\tau_{L-1} \in \{u_1, \dots, u_N\} \\ \tau_{L-1} < \tau_L}} \left\{ \int_{\theta} \eta_L(\tau_L, \theta) \cdot s(\theta) d\theta + DP_{L-1}(\tau_{L-1}) \right\} \quad (4.94)$$

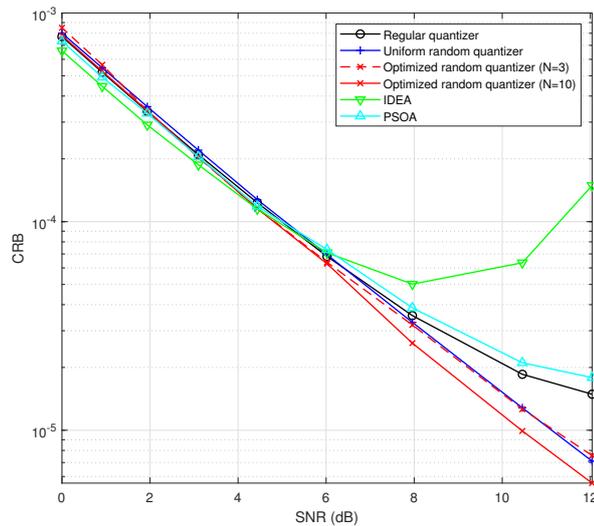
$$\hat{\tau}_{L-1}(\tau_L) \triangleq \arg \max_{\substack{\tau_{L-1} \in \{u_1, \dots, u_N\} \\ \tau_{L-1} < \tau_L}} \left\{ \int_{\theta} \eta_L(\tau_L, \theta) \cdot s(\theta) d\theta + DP_{L-1}(\tau_{L-1}) \right\} \quad (4.95)$$

The final result is obtained by searching $\tau_L^* = \arg \max_{\tau_L \in \{u_1, \dots, u_L\}} \{DP_L(\tau_L)\}$ and then backtracking using functions $\hat{\tau}_{i-1}(\tau_i^*)$ for $i \in \{2, \dots, L\}$. The algorithm is summarized in Algorithm 3.

In Fig. 4.4, we plot the CRB (obtained as the inverse of the Fisher information $\min_j \sum_{k=1}^K F_k(\theta'_j)$) versus SNR for six quantizers including those of [55], [56]. The optimized random quantizer with $N = 10$ gives a significant improvement compared to the other ones. For instance, at mid/high SNR, the gain is around 2dB.

4.7 Conclusion

Our work proposed a worst-case optimized random quantizer in the context of quantized communications of observations to a FC in order to estimate a common parameter. Following an asymptotic approach with respect to the number of sensors, we obtained after a few steps a vectorial optimization problem which writes as signomial programming. The proposed quantizer outperformed uniformly distributed, regular deterministic and previously proposed quantizers for mid/high SNR scenarios.

Algorithm 3: Interval Design for Enhanced Accuracy (IDEA) [55]**Data:** $L, s(\theta), \eta_1(\tau, \theta), \eta(\tau, \tau', \theta), \eta_1(\tau, \theta), \{u_1, \dots, u_N\}$ **Result:** Optimized solution $\{\tau_1^*, \dots, \tau_L^*\}$ Compute and store $DP_1(\tau_1)$ in (4.91) for all $\tau_1 \in \{u_1, \dots, u_N\}$; $i \leftarrow 2$;**while** $i \leq L$ **do** Compute and store $DP_i(\tau_i)$ defined in (4.92) and (4.94) for all $\tau_i \in \{u_0, \dots, u_N\}$; Store $\hat{\tau}_{i-1}(\tau_i)$ for all $\tau_i \in \{u_0, \dots, u_N\}$; $i \leftarrow i + 1$;**end**Find $\tau_L^* \leftarrow \arg \max_{\tau_L} \{DP_L(\tau_L)\}$; $i \leftarrow L$;**while** $i \geq 2$ **do** $\tau_{i-1}^* \leftarrow \hat{\tau}_{i-1}(\tau_i^*)$;**end**Figure 4.4: CRB versus SNR ($K = 2000$).

Stochastic Activation Broadcast Sum-weight for Distributed Estimation

5.1 Introduction

In this chapter, we continue to explore distributed estimation systems. Unlike the previous chapter, we concentrate on the systems that operate without a fusion center, and the sensors are connected by an undirected self-loop-free graph. Similar to the system discussed in the previous chapter, each node observes a noisy version of the target parameter θ , and nodes exchange data with their neighbors to obtain a more precise estimate of θ and achieve consensus.

The problem of achieving consensus on the average of initial sensor measurements is one of the most critical challenges in wireless distributed systems. Regarding node activation, there are two primary approaches for algorithm design: the synchronous approach and the asynchronous approach. In the synchronous approach, multiple sensors can be activated simultaneously to exchange local data. In contrast, the asynchronous approach limits activation to a single sensor or, at most, a pair of sensors at any given time. Early research primarily focused on the asynchronous approach due to its simplicity and stability. For instance, one of the earliest asynchronous algorithms was introduced in [71] and later extended in [61]. This problem is addressed using the *Random Gossip* algorithm, where a randomly selected sensor communicates with one of its neighbors at each iteration. The two sensors exchange their values and update them by averaging the received and previous values. Given the broadcast nature of wireless channels, it is also promising to design algorithms that exploit this property. One example is the *Broadcast Gossip* algorithm [62], where an active node broadcasts its value to all neighbors, which then updates their estimates by averaging the received value with their previous estimates. Moreover, asynchronous algorithms have been studied extensively in various network settings, including directed graphs [72], [73], link failures [74], [75], and unstable sensors [76]. However, as network size increases, the asynchronous approach reveals its drawback of slow convergence. At the same time, advancements in synchronous transmission techniques have mitigated many of their limitations [77]–[79], leading to a renewed interest in the synchronous approach.

In this chapter, we thus focus on the synchronous approach and assume that nodes operate in full-duplex mode. For a given timeslot, each node can be in one of two states: active or inactive. When a node is active during a time slot, it broadcasts its local data to its neighbors; when inactive, it remains silent. Regardless of state, nodes always attempt to receive data from their neighbors. However, nodes may fail to receive data due to potential collisions if too many neighboring nodes are active simultaneously during a timeslot. In the special case of a collision-free system, as shown in [103] and [104], no activation policy is required. Under general assumption, a stochastic activation policy becomes necessary to balance data transmission with the risk of collisions. The objective of this chapter is to design an algorithm that minimizes the MSE of the estimations efficiently and achieves consensus among the nodes.

To address this issue, we propose a synchronous scheme that integrates stochastic activation, where each node can become active with a probability of $1 - \gamma$, as determined by the initial algorithm. For simplicity, we currently assume that γ is uniform across all nodes, with the potential for future refinements. To determine the optimal activation probability, we establish a theoretical relationship between the convergence rate and γ . This analysis builds on the upper bound for convergence rates derived in [82] for the asynchronous approach and later generalized in [83], which, to the best of our knowledge, represents the tightest available bound. Although numerical tests reveal a gap between the actual convergence rate and the upper bound, their monotonic behavior is consistent. This insight enables us to efficiently determine the optimal γ using simple one-dimensional search in a short amount of time. The numerical results confirm the effectiveness of the optimized γ through the upper bound. Additionally, they demonstrate that the performance of applying the averaged optimized γ is comparable to that of applying a graph-dependent γ . This is particularly advantageous in practice, as the algorithm is allowed to operate without the precise knowledge of the graph.

The rest of this chapter is organized as follows: Section 5.2 introduces useful definitions for this chapter. Section 5.3 presents the system model for distributed estimation without a fusion center. In Section 5.4, we propose our synchronous algorithm with stochastic activation. Section 5.5 analyzes the algorithm's convergence rate. Experimental results are discussed in Section 5.6. The chapter concludes in Section 5.7.

5.2 Preliminaries

Let \mathbf{A} be a $K \times K$ matrix. The matrix \mathbf{A} is referred to as *non-negative* if $\forall (k, \ell) \in [K]^2, \mathbf{A}[k, \ell] \geq 0$, and as *positive* if $\forall (k, \ell) \in [K]^2, \mathbf{A}[k, \ell] > 0$.

A non-negative $K \times K$ matrix \mathbf{A} is called *row-stochastic* if the sum of each row equals one, that is:

$$\mathbf{A} \cdot \mathbf{1} = \mathbf{1}.$$

Similarly, \mathbf{A} is called *column-stochastic* if the sum of each column equals one, expressed as:

$$\mathbf{1}^T \cdot \mathbf{A} = \mathbf{1}^T.$$

If a non-negative $K \times K$ matrix \mathbf{A} is both row-stochastic and column-stochastic, it is called *doubly-stochastic*.

Finally, the *spectral radius* of a square matrix is defined as the largest absolute value of its eigenvalues.

5.3 System Model

5.3.1 Graph-based Network

Consider a network of K sensors modeled as a graph $\mathcal{G} = (\mathcal{K}, \mathcal{E})$ where

- \mathcal{K} represents the set of sensors, which correspond to the vertices in terms of graph theory;
- $\mathcal{E} = \{(k, \ell) \in \mathcal{K}^2 \text{ and } k \neq \ell\}$ is the set of links between sensors, and the edges of the graph represented as an ordered pair (k, ℓ) if there is a link from sensor k to sensor ℓ .

This type of object may be called precisely a *directed simple graph*. If a graph satisfies the above definition and additionally meeting the condition that $(k, \ell) \in \mathcal{E} \Rightarrow (\ell, k) \in \mathcal{E}$, it is called *undirected simple graph*. A loop is an edge that links a sensor to itself, and multiple edges are edges that link the same sensors. Under the above definitions, neither loop nor multiple edges are allowed, which corresponds to the nature of the sensor network. In this chapter, we only focus on simple graph and we will use the term "graph" and "simple graph" interchangeably. As the main objective of this chapter is to study the performance related to the structure of graphs, we assume that each link is *error-free*. Let $\mathcal{K}_k = \{\ell : (k, \ell) \in \mathcal{E}\}$ denote the set of neighbors of sensor k , and $K_k = |\mathcal{K}_k|$.

In directed graphs, we define a graph as *weakly connected* if, for any pair $(k, \ell) \in \mathcal{K}^2$, there exists either a path from k to ℓ or a path from ℓ to k . A graph is *strongly connected* if, for any pair $(k, \ell) \in \mathcal{K}^2$, there exists both a path from k to ℓ and a path from ℓ to k . In undirected graphs, these two concepts are equivalent, so we simply refer to them as connected. Define the *adjacency matrix* \mathbf{A} for the graph \mathcal{G} as

$$a_{k,\ell} \triangleq \mathbf{A}[k, \ell] = \begin{cases} 1 & \text{if } (k, \ell) \in \mathcal{E} \\ 0 & \text{if } (k, \ell) \notin \mathcal{E} \end{cases} \quad (5.1)$$

As we assume the graphs are loop-free, all diagonal entries are null. It is easy to deduce that the adjacency matrix is symmetric if the graph is undirected. For undirected graphs, we also define the *degree vector*

$$\mathbf{d} \triangleq (d_1, d_2, \dots, d_K), \quad (5.2)$$

with

$$d_k = \sum_{\ell=1}^K a_{k,\ell}.$$

The *degree matrix* \mathbf{D} is defined as the $K \times K$ diagonal matrix such that

$$\mathbf{D} = \text{diag}(\mathbf{d}).$$

Finally, we define the *Laplacian matrix* of the graph as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \quad (5.3)$$

The eigenvalues of this matrix play a fundamental role in algebraic graph theory, if \mathcal{G} is connected, then the second smallest eigenvalue of the Laplacian λ_2 is strictly positive [105, Lemma 1.7].

5.3.2 Node Activation

The network operates synchronously. In each time slot, a subset of nodes becomes active while others are censored, and active sensors communicate with their neighbors. Details of the communication scheme will be discussed later in this chapter. It is important to note that all nodes estimate the parameter θ using their own data and data received only from neighbors, regardless of whether they are active or not. We define the *activation matrix* $\mathbf{C}_t \in \{0, 1\}^{K \times K}$ as

$$\mathbf{C}_t \triangleq \text{diag}(\mathbf{c}_t), \quad (5.4)$$

where, $\mathbf{c}_t \in \{0, 1\}^K$ is a vector where the k -th entry equals 1 if sensor k is activated at time t , and 0 otherwise. For a sensor k , we denote the $\tilde{\mathcal{K}}_{k,t} \subseteq \mathcal{K}_k$ the activated neighbor of the sensor, and $\tilde{K}_{k,t} = |\tilde{\mathcal{K}}_{k,t}|$.

5.3.3 Communication Failure

Different from the asynchronous setting where only one node is activated for a given time slot, multiple nodes are communication simultaneously in our model. Nodes may experience data reception failures due to potential collisions if too many neighboring nodes are active within a single time slot. We denote $p_{k,t}$ the probability of this communication failure for sensor k at the timeslot t . We define the *collision-free matrix* $\mathbf{S}_t \in \{0, 1\}^{K \times K}$

$$\mathbf{S}_t \triangleq \text{diag}(\mathbf{s}_t), \quad (5.5)$$

where $\mathbf{s}_t \in \{0, 1\}^K$ is a vector where the k -th entry equals 1 if sensor k is collision-free at time t , and 0 otherwise. Thus, the k -th diagonal entry of \mathbf{S}_t follows a Bernoulli distribution with success probability $1 - p_{k,t}$. We also assume that each entry of \mathbf{S}_t is generated independently and the probability $p_{k,t}$ is determined by a function of the active neighborhood size $f(\tilde{K}_{k,t})$, and thus

$$p_{k,t} = 1 - f(\tilde{K}_{k,t}). \quad (5.6)$$

In general, $f(\tilde{K}_{k,t})$ is a monotonically decreasing function bounded in $[0, 1]$, and we also assume perfect transmission if a node only has one active neighbor, i.e. $f(0) = f(1) = 1$. The exact form of $f(\tilde{K}_{k,t})$ depends on the applied protocol, which is not the focus of this chapter.

To obtain an representation of the actual data exchange at time t , we define a new matrix $\tilde{\mathbf{A}}_t$ based on \mathbf{A} , \mathbf{C}_t and \mathbf{S}_t as

$$\tilde{\mathbf{A}}_t \triangleq \mathbf{S}_t \mathbf{A} \mathbf{C}_t. \quad (5.7)$$

We also assume that a sensor experiencing a collision broadcasts a special message to all its neighboring nodes. This message enable its neighbors to become aware of the communication failure. Define $\mathcal{K}'_{k,t} \subseteq \mathcal{K}_k$ the set of collision-free nodes neighboring to node k at time t , and $K'_{k,t} = |\mathcal{K}'_{k,t}|$.

5.3.4 Sensor Observation and Estimation

Similar to the previous chapter, we consider K sensor having one noised measurement

$$y_k = \theta + \varepsilon_k \quad (5.8)$$

However, there is no fusion center in the system. We assume that ε_k are iid and follows a zero-mean normal distribution with variance σ^2 . The goal of the estimation algorithm is to evaluate the parameter θ with linear operations. Let $x_{k,t}$ be the estimation given by the sensor k at the t -th iteration with $x_{k,0} = y_k$. The MSE of the estimation is defined as

$$\text{MSE}_t \triangleq \mathbb{E} \left\{ \frac{1}{K} \|\mathbf{x}_t - \theta \cdot \mathbf{1}\|^2 \right\}, \quad (5.9)$$

where $\mathbf{x}_t \triangleq (x_{1,t}, \dots, x_{K,t})^T$. We introduce the average value of all estimation

$$x_{avg,t} \triangleq 1/K \sum_{k=1}^K x_{k,t}$$

as an intermediate measurement to simplify the analysis of MSE_t . Then, MSE_t can be rewritten as

$$\text{MSE}_t = \mathbb{E} \left\{ \frac{1}{K} \|\mathbf{x}_t - (x_{avg,t} + x_{avg,t} - \theta) \cdot \mathbf{1}\|^2 \right\} \quad (5.10)$$

$$= \mathbb{E} \left\{ \frac{1}{K} \|\mathbf{x}_t - x_{avg,t} \cdot \mathbf{1}\|^2 \right\} + \mathbb{E} \left\{ \frac{1}{K} \sum_{k=1}^K 2(x_{k,t} - x_{avg,t})(x_{avg,t} - \theta) \right\} \\ + \mathbb{E} \left\{ |x_{avg,t} - \theta|^2 \right\} \quad (5.11)$$

$$= \mathbb{E} \left\{ \frac{1}{K} \|\mathbf{x}_t - x_{avg,t} \cdot \mathbf{1}\|^2 \right\} + \mathbb{E} \left\{ |x_{avg,t} - \theta|^2 \right\} \quad (5.12)$$

The cross term in (5.11) is null as $\sum_{k=1}^K (x_{k,t} - x_{avg,t}) = 0$ by the definition of $x_{avg,t}$. To simplify the discussion, we assume the applied estimation algorithm conserve the average for each iteration, i.e.

$$x_{avg,t} = x_{avg,0}, \quad \forall t > 0.$$

We recall that ε_k follows normal distribution. Therefore, we deduce that

$$\mathbb{E} \left\{ |x_{avg,t} - \theta|^2 \right\} = \mathbb{E} \left\{ \left| \frac{1}{K} \sum_{k=1}^K \varepsilon_k \right|^2 \right\} = \frac{\sigma^2}{K}. \quad (5.13)$$

Meanwhile, it is crucial for the estimation algorithm to reach consensus, meaning that the first term in (5.12) tends to zero as t becomes large. The estimation algorithm should thus satisfy two properties: i) Average conservation; ii) Consensus. Next, we will introduce an estimation algorithm that satisfies these two properties and analyze its convergence ratio in detail.

5.4 Broadcast Sum-weight Framework with Stochastic Activation

5.4.1 Average Conservation and Consensus

One approach to design the estimation algorithm involves exchanging local data between sensors and performing linear operations. Specifically, this can be expressed as

$$\mathbf{x}_{t+1} = \mathbf{G}_t \cdot \mathbf{x}_t.$$

The requirements in the previous section can be fulfilled by carefully selecting the matrix \mathbf{G}_t as an doubly-stochastic matrix. This design is feasible because the graphs considered in these studies are undirected, making the adjacency matrix \mathbf{A} symmetric. Since \mathbf{G}_t shares the same support as \mathbf{A} , the doubly-stochastic property of \mathbf{G}_t can be achieved by appropriately selecting its diagonal entries and normalizing them. However, in our setup, \mathbf{G}_t shares the support of $\tilde{\mathbf{A}}$, as defined in (5.7), rather than that of \mathbf{A} . Unlike \mathbf{A} , $\tilde{\mathbf{A}}$ is not necessarily symmetric, as communication failures only impact message reception. As a result, we can only ensure that \mathbf{G}_t is either row-stochastic or column-stochastic, but not doubly-stochastic.

Therefore, we introduce a two variables framework called *sum-weight framework*. The sum-weight framework, originally introduced in [106] and later adapted for wireless networks in [107], relies on the joint update of two variables per node. Specifically, at each node k and time step t , two variables, $s_{k,t}$ (representing the sum, initialized with $x_{k,0}$) and $w_{k,t}$ (representing the weight, initialized with 1), are maintained and updated. At each iteration, the two variables $s_{k,t}$ and $w_{k,t}$ are updated using the same update matrix. Then, the estimation $x_{k,t}$ is obtained by dividing $s_{k,t}$ by $w_{k,t}$, assuming $w_{k,t} \neq 0$. In matrix form, we define

$$\mathbf{s}_t = (s_{1,t}, s_{2,t}, \dots, s_{K,t})^T \quad (5.14)$$

$$\mathbf{w}_t = (w_{1,t}, w_{2,t}, \dots, w_{K,t})^T. \quad (5.15)$$

Then, the update process of the sum-weight framework can be expressed as

$$\mathbf{s}_{t+1} = \mathbf{G}_t \cdot \mathbf{s}_t, \quad (5.16)$$

$$\mathbf{w}_{t+1} = \mathbf{G}_t \cdot \mathbf{w}_t, \quad (5.17)$$

$$\mathbf{x}_{t+1} = \mathbf{s}_{t+1} / \mathbf{w}_{t+1}. \quad (5.18)$$

The division in (5.18) is element-wise.

To ensure sum conservation and consensus, the algorithm requires the update matrices \mathbf{G}_t to satisfy the following conditions [82].

Property 5.1 *The update matrices \mathbf{G}_t must verify:*

- i. The update matrices are non-negative, column-stochastic, and have positive diagonal entries;*

ii. The update matrices are chosen through an independent and identically distributed process;

iii. $\mathbb{E}[\mathbf{G}_t]$ is primitive.

The column-stochasticity in Property 5.1-*i* ensures that the sum conservation. Property 5.1-*ii* can be easily achieved by selecting \mathbf{G}_t independently and identically distributed between iterations. Property 5.1-*iii* holds if $\text{Supp}(\mathbb{E}\{\mathbf{G}\}) = (\mathbf{Id}_K + \mathbf{A})$ and \mathbf{A} is the adjacency matrix of the graph. The detailed design of the update matrix will be given later in this section.

5.4.2 Framework Design

Similar to sum-weight framework, sensors exchange and update two variables $s_{k,t}$ and $w_{k,t}$ in every timeslot. In addition to considering the convergence of the algorithm, it is crucial to address potential communication failures caused by collisions. As mentioned at the beginning of this chapter, one common approach to mitigate such failures is the use of asynchronous algorithms, which allow only one node to transmit local data during a given time slot. While these methods are collision-free, they typically result in slower convergence. To overcome this limitation, we implement a more flexible scheme where each node is active with probability $1 - \gamma$, a parameter authorized by the initial algorithm. For simplicity, we assume that γ is independent of the individual nodes.

At time 0, we have $s_{k,0} = y_k$ and $w_{k,0} = 1$. For time slot t ,

- With probability $(1 - p_{k,t})$, node k update the local data as

$$s_{k,t+1} = \sum_{\ell \in \tilde{\mathcal{K}}_{k,t} \cup \{k\}} g_{k,\ell,t} \cdot s_{\ell,t}, \quad (5.19)$$

$$w_{k,t+1} = \sum_{\ell \in \tilde{\mathcal{K}}_{k,t} \cup \{k\}} g_{k,\ell,t} \cdot w_{\ell,t}. \quad (5.20)$$

- With probability $p_{k,t}$ (which corresponds to communication failure), node k updates the local data as

$$s_{k,t+1} = g_{k,k,t} \cdot s_{k,t}, \quad w_{k,t+1} = g_{k,k,t} \cdot w_{k,t}. \quad (5.21)$$

We recall that K_k is the number of neighbors of Node k , $\mathcal{K}'_{\ell,t}$ the set of collision-free nodes neighboring to node k and $K'_{\ell,t} = |\mathcal{K}'_{\ell,t}|$. For a given $\ell \in [K]$ and $k \in \mathcal{K}'_{\ell,t}$, the parameter $g_{k,\ell,t}$ is defined as

$$g_{k,\ell,t} \triangleq \begin{cases} 1/(K_\ell + 1) & \text{if } k \neq \ell, \\ (K_k - K'_{k,t} + 1)/(K_k + 1) & \text{if } k = \ell, \text{ and node } k \text{ activated} \\ 1 & \text{if } k = \ell, \text{ and node } k \text{ non-activated} \end{cases} \quad (5.22)$$

Also define the matrix \mathbf{G}_t with $\mathbf{G}_t[k, \ell] = g_{k,\ell,t}$. We recall that the diagonal matrix \mathbf{C}_t defined in (5.4) is the activation matrix, the diagonal matrix \mathbf{S}_t defined in (5.5) is the

collision-free matrix, and in (5.7), we define $\tilde{\mathbf{A}}_t = \mathbf{S}_t \mathbf{A} \mathbf{C}_t$. With these matrices, we obtain the expression in matrix form of \mathbf{G}_t as

$$\mathbf{G}_t \triangleq \left(\text{diag}(\mathbf{1}^T \mathbf{A} + \mathbf{1}^T - \mathbf{1}^T \tilde{\mathbf{A}}_t) + \tilde{\mathbf{A}}_t \right) \mathbf{N}, \quad (5.23)$$

where \mathbf{N} is the diagonal normalization matrix that ensures the matrix \mathbf{G}_t is column-stochastic, and the k -th entry on the diagonal is

$$\mathbf{N}[k, k] = \frac{1}{K_k + 1}, \forall k \in [K].$$

Notice that $\mathbf{1}^T \mathbf{A}$ is the degree vector of the adjacency matrix \mathbf{A} , and $\mathbf{1}^T \cdot \tilde{\mathbf{A}}$ is the out-degree vector of the matrix $\tilde{\mathbf{A}}$.

Example 5.1 Consider the graph in Fig. 5.1. The adjacency matrix is given by

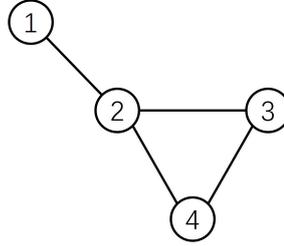


Figure 5.1: Example graph

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (5.24)$$

We assume that only node 3 does not transmit in the time slot t , so $\mathbf{C}_t = \text{diag}([1, 1, 0, 1])$. The numbers of activated neighbors for each sensor are

$$\tilde{K}_{1,t} = 1, \quad \tilde{K}_{2,t} = 2, \quad \tilde{K}_{3,t} = 2, \quad \tilde{K}_{4,t} = 1.$$

The communication failure probabilities for each sensor are

$$p_{1,t} = 1 - f(1) = 0, \quad p_{2,t} = 1 - f(2), \quad p_{3,t} = 1 - f(2), \quad p_{4,t} = 1 - f(1) = 0.$$

In this case, nodes 2 and 3 may experience potential communication failures. Suppose only node 2 encounters collisions, represented as $\mathbf{S}_t = \text{diag}([1, 0, 1, 1])$. The resulting message flow graph is depicted in Fig. 5.2, with the corresponding matrix given as:

$$\tilde{\mathbf{A}}_t = \mathbf{S}_t \mathbf{A} \mathbf{C}_t = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (5.25)$$

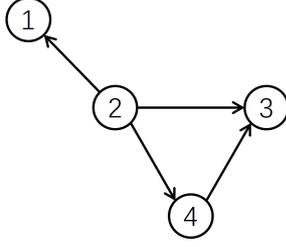


Figure 5.2: Actual message flow graph

The numbers of collision-free neighbors for each sensor are

$$K'_{1,t} = 0, \quad K'_{2,t} = 3, \quad K'_{3,t} = 1, \quad K'_{4,t} = 1.$$

Substituting the above results into (5.22), we obtain

$$\mathbf{G}_t = \begin{bmatrix} 1 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 1 & \frac{1}{3} \\ 0 & \frac{1}{4} & 0 & \frac{2}{3} \end{bmatrix} \quad (5.26)$$

It is straightforward to observe that the matrix \mathbf{G}_t satisfies Property 5.1-i, as the update matrices are non-negative, column-stochastic, and have positive diagonal entries.

5.5 Convergence Rate Analysis

In this section, we analyze the convergence rate of MSE_t with the algorithm proposed in the previous section. Since the update matrix \mathbf{G}_t is column-stochastic, and is thus average-conserving for each iteration, the expectation of the 2nd term in (5.12) is a constant as it is the variance of $1/K \sum \varepsilon_k$. We thus focusing on the first term of MSE_t , and denote

$$\text{MSE}'_t \triangleq \mathbb{E} \left\{ \|\mathbf{x}_t - x_{avg,t}\|^2 \right\}.$$

The goal is to identify an optimal probability γ of not transmitting that is adapted to the failure error, thereby maximizing the convergence rate. To achieve this, we establish an relation between the probability γ and MSE'_t in this section. According to the Proposition 3.16 in [82], the important factor for the convergence speed of MSE'_t is

$$\omega \triangleq -\log \left(\rho \left(\mathbb{E} [\mathbf{G}_t \otimes \mathbf{G}_t] \cdot (\mathbf{J}^\perp \otimes \mathbf{J}^\perp) \right) \right), \quad (5.27)$$

where $\rho(\cdot)$ is the spectral radius of a matrix, and

$$\mathbf{J}^\perp \triangleq \mathbf{Id}_K - \frac{1}{K} \mathbf{1}\mathbf{1}^T.$$

As the majority of the expression in (5.27) is deterministic except the term $\mathbb{E} [\mathbf{G}_t \otimes \mathbf{G}_t]$, we focus on the calculation of this expectation. For simplicity, we omit the index t as we

assume all matrices are independent between different timeslots. We obtain

$$g_{k,\ell} = \begin{cases} \frac{\tilde{a}_{k,\ell}}{K_\ell + 1} & \text{if } k \neq \ell \\ 1 - \frac{\sum_{k_1=1}^K \tilde{a}_{k_1,\ell}}{K_\ell + 1} & \text{if } k = \ell \end{cases}, \quad (5.28)$$

with $\tilde{a}_{k,\ell} = \tilde{\mathbf{A}}[k, \ell] = s_k a_{k,\ell} c_\ell$, and we recall that the communication failure probability $p_k = 1 - f(\tilde{K})$ with $\tilde{K}_k = \sum_{\ell=1}^K a_{k,\ell} c_\ell$.

By the definition of Kronecker product, we have $(\mathbf{G} \otimes \mathbf{G})[K \cdot k + k', K \cdot \ell + \ell'] = g_{k,\ell} \cdot g_{k',\ell'}$. There are 3 different cases for $\mathbb{E}[g_{k,\ell} \cdot g_{k',\ell'}]$.

- When $k \neq \ell$ and $k' \neq \ell'$, we have

$$\mathbb{E}[g_{k,\ell} \cdot g_{k',\ell'}] = \frac{1}{(K_\ell + 1)(K_{\ell'} + 1)} \cdot \mathbb{E}[\tilde{a}_{k,\ell} \cdot \tilde{a}_{k',\ell'}] \quad (5.29)$$

- When $k = \ell$ and $k' \neq \ell'$ (or the case when $k \neq \ell$ and $k' = \ell'$ is similar), we have

$$\mathbb{E}[g_{\ell,\ell} \cdot g_{k',\ell'}] = \frac{1}{K_\ell + 1} \cdot \mathbb{E}[\tilde{a}_{k',\ell'}] - \frac{1}{(K_\ell + 1)(K_{\ell'} + 1)} \cdot \sum_{k_1=1}^K \mathbb{E}[\tilde{a}_{k_1,\ell} \cdot \tilde{a}_{k',\ell'}] \quad (5.30)$$

- When $k = \ell$ and $k' = \ell'$, we have

$$\begin{aligned} \mathbb{E}[g_{\ell,\ell} \cdot g_{\ell',\ell'}] &= 1 - \frac{1}{K_\ell + 1} \cdot \sum_{k_1=1}^K \mathbb{E}[\tilde{a}_{k_1,\ell}] - \frac{1}{K_{\ell'} + 1} \cdot \sum_{k'_1=1}^K \mathbb{E}[\tilde{a}_{k'_1,\ell'}] \\ &\quad + \frac{1}{(K_\ell + 1)(K_{\ell'} + 1)} \cdot \sum_{k_1=1}^K \sum_{k'_1=1}^K \mathbb{E}[\tilde{a}_{k_1,\ell} \cdot \tilde{a}_{k'_1,\ell'}] \end{aligned} \quad (5.31)$$

According to (5.29)-(5.31), we need to calculate $\mathbb{E}[a_{k,\ell}]$ and $\mathbb{E}[a_{k,\ell} \cdot a_{k',\ell'}]$. Let's begin with the term $\mathbb{E}[a_{k,\ell}]$. For a given pair of $(k, \ell) \in [K]^2$, we have

$$\mathbb{E}[\tilde{a}_{k,\ell}] = a_{k,\ell} \cdot \mathbb{E}[s_k | c_\ell = 1] \cdot \mathbb{P}(c_\ell = 1) \quad (5.32)$$

We recall that s_k follow the Bernoulli distribution with a success probability determined by the number of neighboring active nodes \tilde{K}_k and the function in (5.6). We only calculate $\mathbb{E}[s_k | c_\ell = 1]$ for the case where $a_{k,\ell} = 1$ as $\mathbb{E}[\tilde{a}_{k,\ell}]$ is non-zero only when $a_{k,\ell} = 1$, we thus obtain

$$\mathbb{E}[s_k | c_\ell = 1] = \sum_{m=1}^{K_k} \mathbb{E}[s_k | c_\ell = 1, \tilde{K}_k = m] \cdot \mathbb{P}(\tilde{K}_k = m | c_\ell = 1).$$

The summation starts with 1 because we assume that node ℓ is active and that node k and ℓ are connected. The random variable $s_k | \tilde{K}_k = m$ follows a Bernoulli distribution with success probability $f(m)$, so $\mathbb{E}[s_k | c_\ell = 1, \tilde{K}_k = m] = f(m)$. Also recall that for $\ell \in [K]$ the activation of node ℓ follows the Bernoulli distribution with success probability $1 - \gamma$. Therefore, \tilde{K}_k follows the binomial distribution $B(K_k, 1 - \gamma)$, i.e.

$$\mathbb{P}(\tilde{K}_k = m | c_\ell = 1) = \binom{K_k - 1}{m - 1} (1 - \gamma)^{(m-1)} \gamma^{(K_k - m)}, \quad \text{for } 1 \leq m \leq K_k.$$

Then, we deduce that

$$\mathbb{E}[s_k | c_\ell = 1] = \sum_{m=1}^{K_k} \binom{K_k - 1}{m - 1} f(m) (1 - \gamma)^{(m-1)} \gamma^{(K_k - m)} \quad (5.33)$$

To simplify the equation, we define

$$F_1(K) \triangleq \sum_{m=1}^K \binom{K - 1}{m - 1} f(m) (1 - \gamma)^{(m-1)} \gamma^{(K - m)}. \quad (5.34)$$

By combining (5.32) and (5.33), we obtain

$$\mathbb{E}[\tilde{a}_{k,\ell}] = a_{k,\ell} \cdot F_1(K_k) \cdot (1 - \gamma). \quad (5.35)$$

Then, we calculate the term $\mathbb{E}[\tilde{a}_{k,\ell} \tilde{a}_{k',\ell'}]$. For two pairs $(k, \ell) \in [K]^2$ and $(k', \ell') \in [K]^2$, we have

$$\mathbb{E}[\tilde{a}_{k,\ell} \tilde{a}_{k',\ell'}] = a_{k,\ell} a_{k',\ell'} \cdot \mathbb{E}[s_k s_{k'} | c_\ell c_{\ell'} = 1] \cdot \mathbb{P}(c_\ell c_{\ell'} = 1). \quad (5.36)$$

Since s_k and $s_{k'}$ are independent when $k \neq k'$, and c_ℓ and $c_{\ell'}$ are independent when $\ell \neq \ell'$, we have

$$\mathbb{E}[\tilde{a}_{k,\ell} \tilde{a}_{k',\ell'}] = \begin{cases} a_{k,\ell} a_{k',\ell'} \cdot \mathbb{E}[s_k | c_\ell = 1] \mathbb{E}[s_{k'} | c_{\ell'} = 1] \cdot (1 - \gamma) & \text{if } k \neq k', \ell = \ell' \\ a_{k,\ell} \cdot \mathbb{E}[s_k | c_\ell = 1] \cdot (1 - \gamma) & \text{if } k = k', \ell = \ell' \\ a_{k,\ell} a_{k',\ell'} \cdot \mathbb{E}[s_k | c_\ell c_{\ell'} = 1] \mathbb{E}[s_{k'} | c_\ell c_{\ell'} = 1] \cdot (1 - \gamma)^2 & \text{if } k \neq k', \ell \neq \ell' \\ a_{k,\ell} a_{k',\ell'} \cdot \mathbb{E}[s_k | c_\ell c_{\ell'} = 1] \cdot (1 - \gamma)^2 & \text{if } k = k', \ell \neq \ell' \end{cases} \quad (5.37)$$

As $\mathbb{E}[s_k | c_\ell = 1]$ is calculated in (5.33), we only need to calculate $\mathbb{E}[s_k | c_\ell c_{\ell'} = 1]$. Similarly, we only calculate for the case $a_{k,\ell} = 1$ and $a_{k',\ell'} = 1$. We first consider the case that $k \neq k'$ and $\ell \neq \ell'$. If node k and ℓ' are not connected, i.e. $a_{k,\ell'} = 0$, we obtain

$$\mathbb{E}[s_k | c_\ell c_{\ell'} = 1] = \mathbb{E}[s_k | c_\ell = 1] = F_1(K_k). \quad (5.38)$$

Otherwise, node k has at least two active neighbors, then

$$\mathbb{E}[s_k | c_\ell c_{\ell'} = 1] = \sum_{m=2}^{K_k} \binom{K_k - 2}{m - 2} f(m) (1 - \gamma)^{(m-2)} \gamma^{(K_k - m)}. \quad (5.39)$$

And define

$$F_2(K) \triangleq \sum_{m=2}^K \binom{K - 2}{m - 2} f(m) (1 - \gamma)^{(m-2)} \gamma^{(K - m)}. \quad (5.40)$$

The two expression can be combined and rewritten as

$$\mathbb{E}[s_k | c_\ell c_{\ell'} = 1] = F_{1+a_{k,\ell'}}(K_k). \quad (5.41)$$

When $k = k'$ and $\ell \neq \ell'$, we have $a_{k,\ell'} = a_{k',\ell'} = 1$, so

$$\mathbb{E}[s_k | c_\ell c_{\ell'} = 1] = F_2(K_k). \quad (5.42)$$

By (5.33), (5.37), (5.41) and (5.42), we obtain

$$\mathbb{E}[\tilde{a}_{k,\ell} \cdot \tilde{a}_{k',\ell'}] = \begin{cases} a_{k,\ell} a_{k',\ell'} \cdot F_1(K_k) F_1(K_{k'}) \cdot (1 - \gamma) & \text{if } k \neq k', \ell = \ell' \\ a_{k,\ell} \cdot F_1(K_k) \cdot (1 - \gamma) & \text{if } k = k', \ell = \ell' \\ a_{k,\ell} a_{k',\ell'} \cdot F_{1+a_{k,\ell}}(K_k) F_{1+a_{k',\ell'}}(K_{k'}) \cdot (1 - \gamma)^2 & \text{if } k \neq k', \ell \neq \ell' \\ a_{k,\ell} a_{k,\ell'} \cdot F_2(K_k) \cdot (1 - \gamma)^2 & \text{if } k = k', \ell \neq \ell' \end{cases} \quad (5.43)$$

Finally, we can calculate the term $\mathbb{E}[\mathbf{G}_t \otimes \mathbf{G}_t]$ by applying the results from (5.35) and (5.43) to (5.29)-(5.31), and subsequently derive ω as defined in (5.27).

Example 5.2 *To better illustrate the application of Proposition 3.16 in [82], we outline the method for computing ω in the case study presented in Example 5.1. We assume an activation probability of $1 - \gamma = \frac{1}{2}$ and a communication success probability function given by $f(K) = \min(\frac{1}{2^K}, 1)$. Using these values, we compute F_1 from (5.34) and F_2 from (5.40), yielding:*

$$F_1(K) = \frac{3^{K-1}}{2^{2K-2}}, \quad F_2(K) = \frac{3^{K-2}}{2^{2K-2}}.$$

As stated in the example, the values of K_k are:

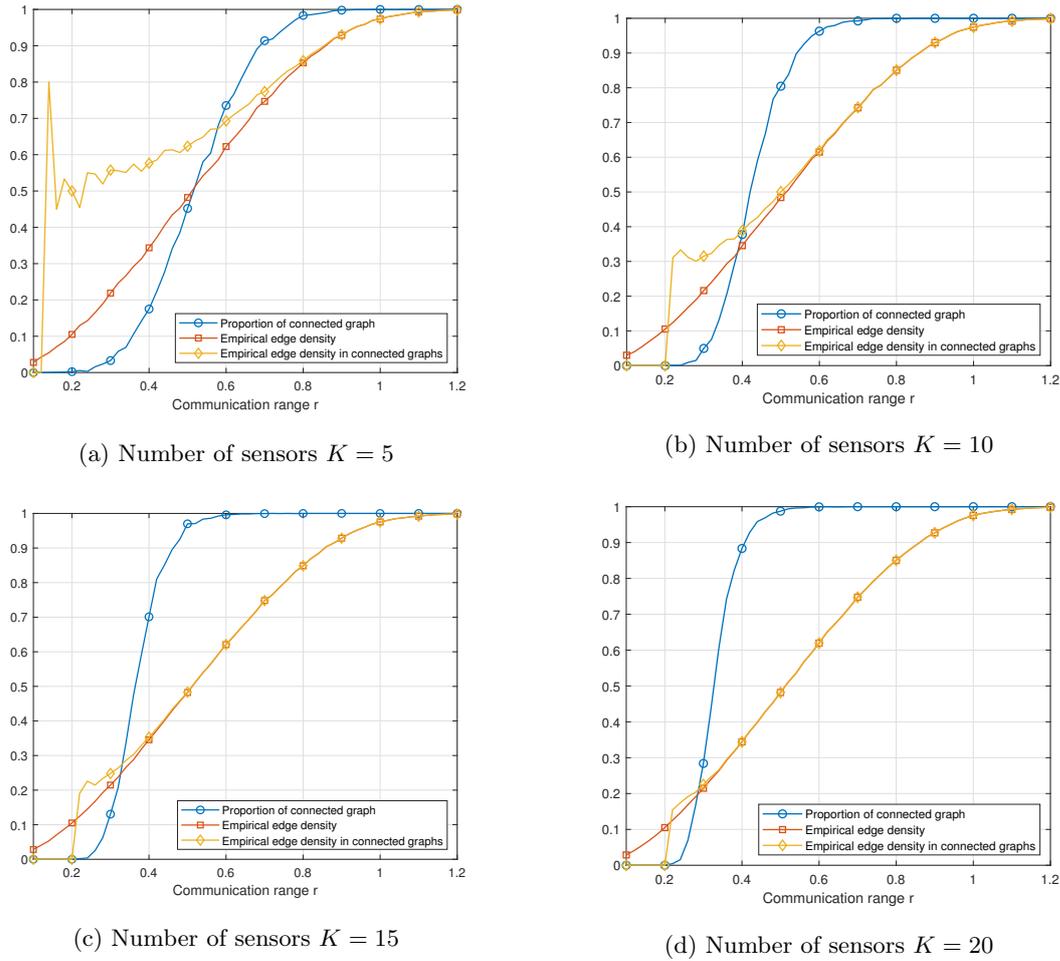
$$K_1 = 1, \quad K_2 = 3, \quad K_3 = 2, \quad K_4 = 2.$$

Next, we determine $\mathbb{E}[\tilde{a}_{k,\ell}]$ and $\mathbb{E}[\tilde{a}_{k,\ell} \cdot \tilde{a}_{k',\ell'}]$ using (5.35) and (5.42), respectively. Additionally, we compute $\mathbb{E}[\tilde{g}_{k,\ell} \cdot \tilde{g}_{k',\ell'}]$ based on (5.29)-(5.31), incorporating $\mathbb{E}[\tilde{a}_{k,\ell}]$ and $\mathbb{E}[\tilde{a}_{k,\ell} \cdot \tilde{a}_{k',\ell'}]$. Finally, the lower bound on the convergence speed of MSE'_t is derived using (5.27).

5.6 Numerical Results

The graphs applied in the simulation are Random Geometric Graph (RGG), which are generated as follows: first, select K points uniformly within the unit square $[0, 1] \times [0, 1]$ to represent the locations of the sensors. Then, connect any two sensors with an undirected edge if they are within a specified radius r , which defines the communication range. This radius determines which sensors are close enough to communicate and thus have an edge between them. RGG is suitable for modeling wireless sensor networks and aligns closely with our setup. In Fig. 5.3, we present three curves in each sub-figure: i) The proportion of connected graph in all graphs; ii) The empirical edge density, defined as the average ratio between the number of connected sensor and $K - 1$; iii) The empirical edge density for connected graphs. Each sub-figure presents the results for a specific number of sensors.

Notice that the connectivity is not guaranteed in RGG. Simulations are conducted with the number of nodes $K = 10$ and communication range $r = 0.4$, as this setting yields a sufficiently large proportion of connected graphs according to the results in Fig. 5.3b. We generate RGGs in the way described in the previous sub-section, discarding non-connected graphs until we obtain 100 connected RGG. For each graph, simulations are conducted for 2000 iterations, carried out 100 times, with sensors initialized each time using newly generated noisy observation vectors \mathbf{x}_0 as defined in (5.8). We choose the parameter $\theta = 2$


 Figure 5.3: Connectivity and edge density with different r and K

and the variance of noise $\sigma^2 = 1$. For the function of communication success probability in (5.6), we choose

$$f(\tilde{K}) = \exp\left(-\alpha \max\{\tilde{K} - 1, 0\}\right) \quad (5.44)$$

The parameter α is a coefficient to quantify the collision strength of the system (or the low quality of the receiver). The higher α is, the higher the failure at the receiver occurs. We obtain the numerical form of (5.35) as follows:

$$\mathbb{E}[\tilde{a}_{k,\ell}] = a_{k,\ell} \cdot \beta^{(K_k-1)}(1 - \gamma), \quad (5.45)$$

where

$$\beta \triangleq \exp(-\alpha) \cdot (1 - \gamma) + \gamma. \quad (5.46)$$

And the numerical for of (5.35) is given as follows:

$$\mathbb{E}[\tilde{a}_{k,\ell} \cdot \tilde{a}_{k',\ell'}] = \begin{cases} a_{k,\ell} a_{k',\ell'} \cdot \beta^{(K_k+K_{k'}-2)} \cdot (1 - \gamma) & \text{if } k \neq k', \ell = \ell' \\ a_{k,\ell} \cdot \beta^{(K_k-1)} \cdot (1 - \gamma) & \text{if } k = k', \ell = \ell' \\ a_{k,\ell} a_{k',\ell'} \cdot \exp(-\alpha(a_{k,\ell'} + a_{k',\ell})) \cdot \beta^{(K_k+K_{k'}-2-a_{k,\ell'}-a_{k',\ell})} \cdot (1 - \gamma)^2 & \text{if } k \neq k', \ell \neq \ell' \\ a_{k,\ell} a_{k',\ell'} \cdot \exp(-\alpha) \cdot \beta^{(K_k-2)} \cdot (1 - \gamma)^2 & \text{if } k = k', \ell \neq \ell' \end{cases}, \quad (5.47)$$

In the simulations, we focus on the following metrics:

- **Slope:** The experimental convergence rate for each simulation is determined by performing linear regression on $\log(\|\mathbf{x}_t - x_{avg} \cdot \mathbf{1}\|^2)$ for the points with values below -2 . We refer to this value as the *Slope*.
- ω : The value of ω is computed using the results from (5.27), (5.28), (5.45) and (5.47), which do not require any simulation results.
- **Empirical ω :** The empirical value of ω is calculated by applying (5.27) with the empirical mean of the matrices \mathbf{G}_t and $\mathbf{G}_t \otimes \mathbf{G}_t$ obtained in each iteration of simulations.

We compare the slope and ω for different (α, γ) pairs in Fig. 5.4. The figure shows scatter plots for each combination of $\alpha \in \{0, 0.5, 1\}$ and $\gamma \in \{0, 0.2, \dots, 0.8\}$ with $K = 10$ nodes. Each point in the scatter plots represents a (slope, ω) pair obtained from one graph. The diagonal line $y = x$ is added to each plot for reference.

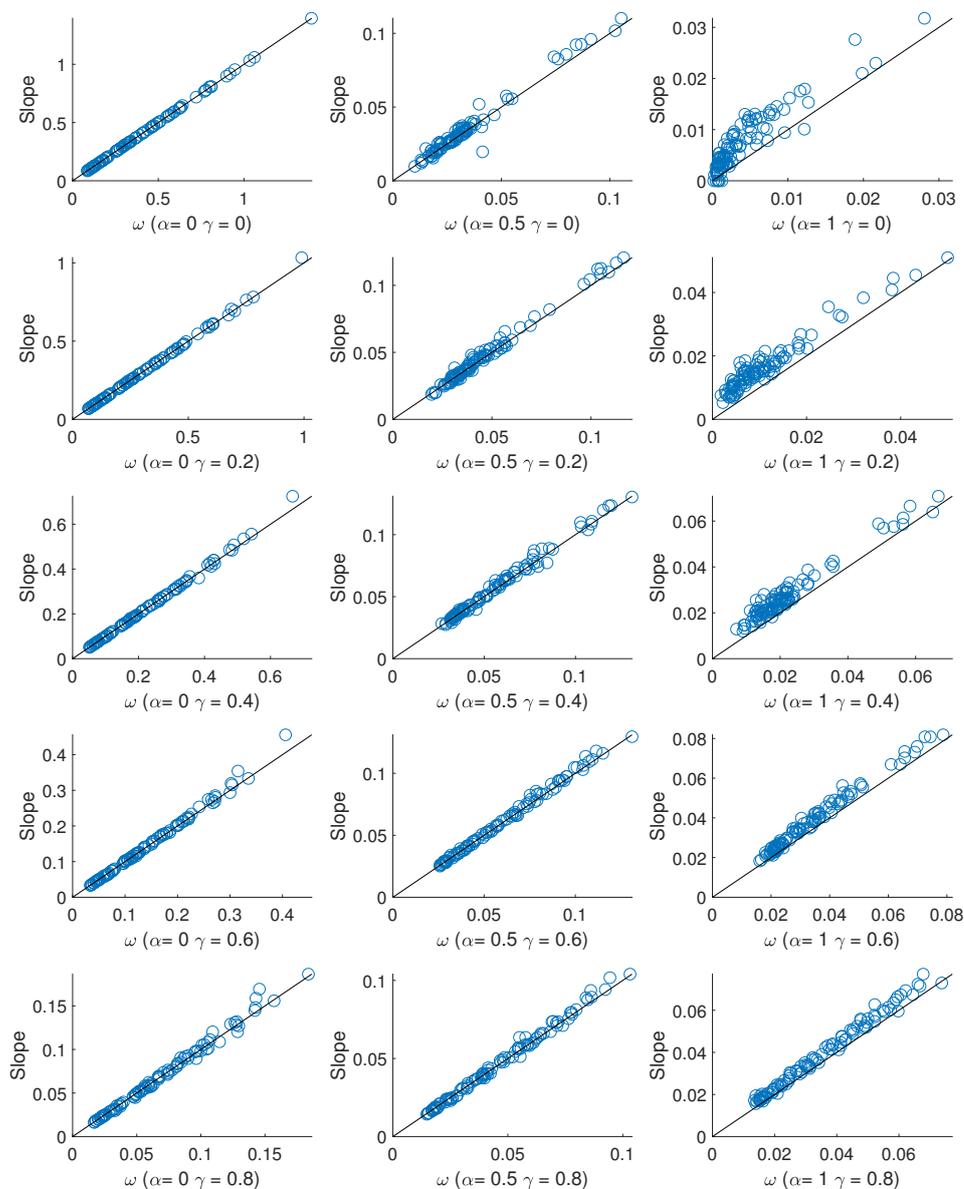
We observe that all points are close to the line $y = x$, suggesting that ω is a good approximation for the slope. Furthermore, for $\alpha \in \{0.5, 1\}$, most points lie above the $y = x$ line, indicating that the slope is generally greater than ω in these cases. This is expected since ω serves as the lower bound for the convergence rate. For $\alpha = 0$, the points are closer to the $y = x$ line, implying that the difference between ω and the slope is influenced by α . This observation could provide insight into how to improve the bound on the convergence rate.

Then, we compare the average slope, the average ω and the average empirical ω over all graphs for different value of γ in Fig 5.5. The curves of ω and of the empirical ω well coincide, which validates our calculation of $\mathbb{E}[\mathbf{G} \otimes \mathbf{G}]$. Additionally, we observe similar effect in Fig. 5.4 that ω coincides with the slope when $\alpha = 0$ and positioned slightly below the curve of slope when $\alpha \in \{0.5, 1\}$. Even though the existence of the gap, ω effectively indicates the variance of convergence speed with γ . Therefore, the optimal γ for a given graph and α can be obtained by one-dimensional search.

In Fig 5.6 and Fig 5.7, we present MSE'_t and MSE_t respectively with different values of γ . The optimal line is obtained by applying the optimized γ for each graph, rather than using a fixed value. The "Average Optimal γ " line is obtained by applying the average of the optimized γ values across all graphs.

The optimal lines in both figures highlight the advantages of using a dynamic approach. However, the performance of employing the averaged optimized γ is comparable to that of applying a dynamic γ . Moreover, using a fixed γ enables the algorithm to function without requiring precise knowledge of the graph's structure. Consequently, the averaged optimized γ is preferred from a practical standpoint. Additionally, as shown in Fig. 5.7, all MSE values converge to σ^2/K , which aligns with the result in (5.13).

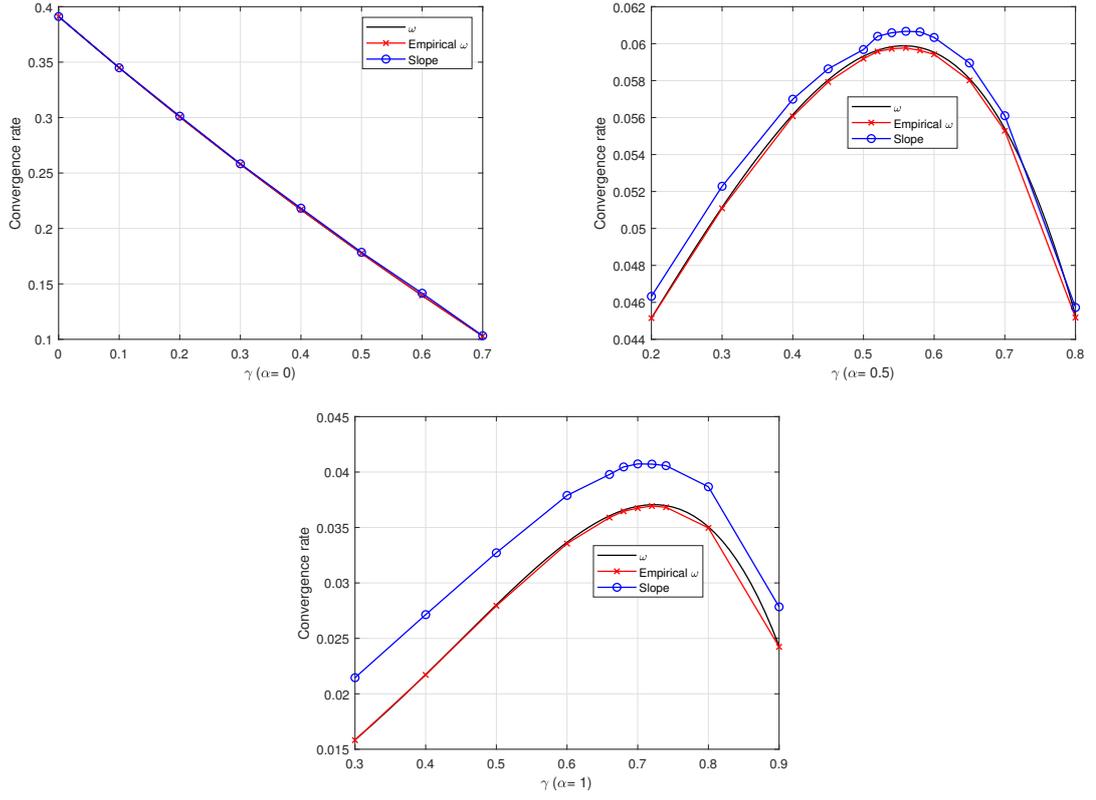
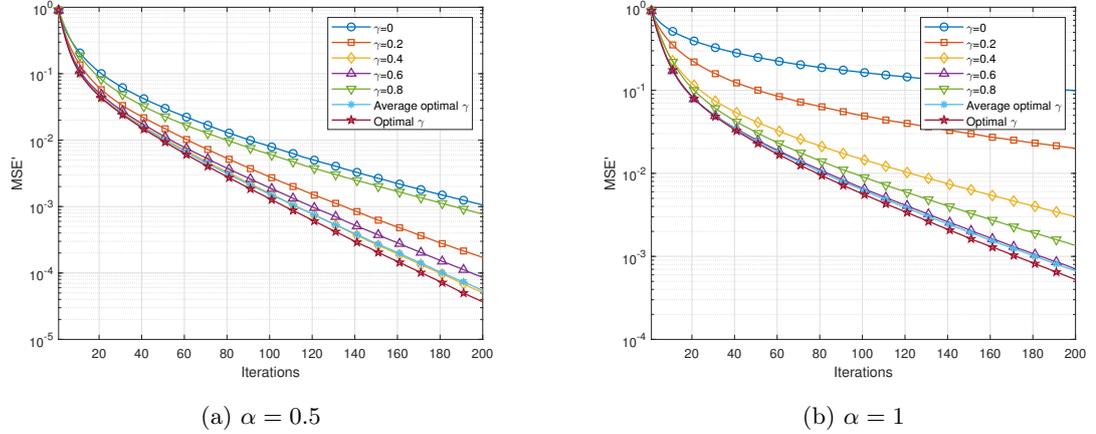
In Fig. 5.8, we present the average optimal γ over all graphs versus communication range r . As edge density is nearly "proportional" to r . Figure 5.8 highlights the impact of the communication range r on the average optimal γ under varying α values. The two curves illustrate that the optimal γ grows with increasing r , and the effect is more

Figure 5.4: Slope versus ω for different γ and α with $K = 10$

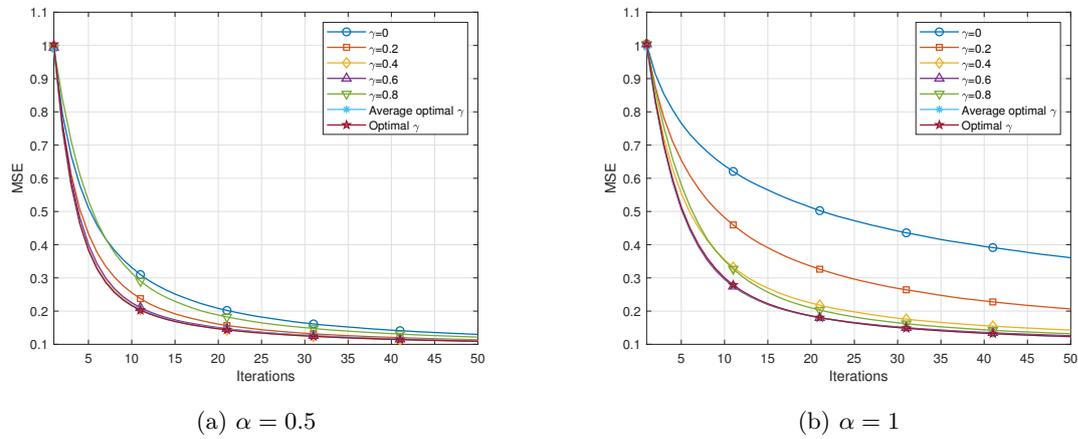
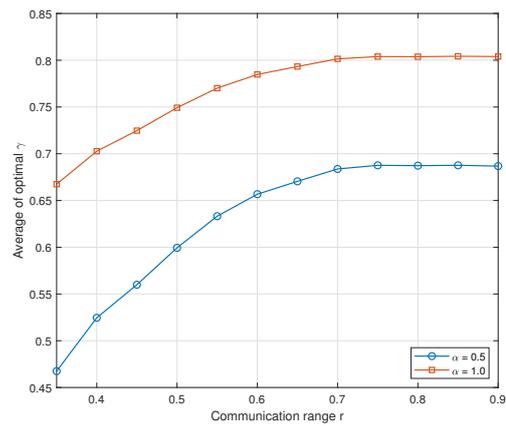
pronounced when $\alpha = 1.0$, indicating a stronger preference for balancing tradeoffs in systems with higher α .

5.7 Conclusion

In conclusion, this chapter addresses the challenge of achieving consensus in distributed estimation systems operating without a fusion center, where nodes are connected by an


 Figure 5.5: Average slope, ω and empirical ω over graphs versus γ for different α with $K = 10$

 Figure 5.6: Simulation results of Broadcast Sum-Weight on MSE'_t with different α , γ with $K = 10$.

undirected, self-loop-free graph. By focusing on the synchronous approach, we assume nodes work in full-duplex mode, allowing them to either broadcast data or remain silent during each time slot. Although node collisions can occur when too many nodes are active simultaneously, we introduce a stochastic activation scheme to mitigate this issue. By allowing each node to become active with a probability of $1 - \gamma$, we establish a theoretical relationship between the convergence rate and γ , leveraging prior research on upper bounds to guide algorithm design. This approach allows for efficient optimization of γ , ensuring minimal MSE in the estimation process and enabling consensus.

Figure 5.7: Simulation results of Broadcast Sum-Weight on MSE_t for different α, γ with $K = 10$.Figure 5.8: The average of the optimal γ versus the communication range r .

Conclusions and Perspectives

This thesis investigated distributed computation and estimation algorithms, along with their theoretical limits, to enable networks to achieve consensus on values of interest. By addressing different systems and objectives such as SDoF, NDT, CRB, and convergence rate, various algorithms were proposed and analyzed, providing insights into the fundamental and practical aspects of these systems.

In Chapter 2, we derived and analyzed SDoF bounds for three subclasses of partially connected channels: NPC, CPXC, and CPIC. The results include a new lower bound for NPC, improved bounds for CPXC, and the exact SDoF for CPIC. These findings were applied to improve the NDT performance of wireless distributed MapReduce systems.

Chapter 3 extended this analysis by presenting the first information-theoretic lower bound and an improved upper bound on the NDT tradeoff in full-duplex wireless MapReduce systems. Using zero-forcing and a novel interference alignment scheme, we demonstrated that linear beamforming, zero-forcing, and interference cancellation are optimal when nodes can store at least half the files. However, the suboptimality of these methods in other cases and the high computational cost of interference alignment highlight avenues for future research, such as designing practical IA algorithms for MapReduce systems.

In Chapter 4, we addressed the challenge of quantized communication for distributed estimation. By proposing a worst-case optimized random quantizer, we formulated a vector optimization problem, solved as signomial programming. The resulting quantizer outperformed existing methods in mid/high SNR scenarios, offering a robust solution for quantized communications to a fusion center.

Finally, Chapter 5 tackled consensus in distributed estimation systems without a fusion center, focusing on synchronous approaches in undirected, self-loop-free graphs. A stochastic activation scheme was introduced to mitigate node collisions during simultaneous activation. By relating the convergence rate to the activation probability, γ , we optimized this parameter to minimize the MSE in estimation, facilitating efficient consensus across the network.

Overall, this work contributes to the theoretical understanding and practical design of distributed computation and estimation systems. It provides a foundation for future advancements, such as more efficient algorithms for interference alignment and refined techniques for consensus in large-scale networks.

Perspectives

We present in the sequel some future perspectives to extend the work of this thesis.

A Unified Lower Bound for PC In Chapter 2, we derived the lower bound for NPC, a specific subclass of PC that restricts each group to a single user. The methodology used to obtain the lower bound for NPC could potentially be generalized to establish a unified and extended bound for PC, encompassing all lower bounds presented in Chapter 2. Specifically, for a given receiver, interference from other groups can be managed using IA, while interference within the group can be managed using ZF. The precoding matrices assignment could follow the approach developed for NPC. The main challenge lies in demonstrating the independence of the useful signal subspace and the interference subspace.

Reevaluating the Assumption of Full CSI In Chapter 2 and Chapter 3, we developed IA schemes based on the assumption of full CSI. While full CSI provides an idealized scenario for theoretical analysis, its practicality in real-world networked systems is questionable. Given the inherent grouping structure in CPXC and CPIC, a more realistic assumption would be partial CSI, where each node only has access to the channel information within the group. This raises important considerations regarding the trade-off between performance and the overhead required to obtain and distribute CSI. Specifically, maintaining full CSI necessitates extensive signaling and feedback mechanisms, which may introduce significant communication overhead and delay, particularly in large-scale or dynamically changing networks. A deeper investigation into the feasibility of partial CSI and its impact on system performance could provide valuable insights into designing more efficient and scalable communication protocols.

Develop Heuristic Approaches for NPC Lower Bound The lower bound for NPC presented in Theorem 2.1 is implicit. To derive a practical bound for NPC, integer programming can be employed for specific wireless system configurations. By formulating the problem as an integer program, the allocation of precoding matrices can be optimized to achieve the best possible performance.

Furthermore, numerical results obtained through integer programming can serve as benchmarks for developing heuristic algorithms. These algorithms can approximate the optimal solutions with significantly reduced computational complexity, making them more suitable for large-scale systems.

Extended Distributed Estimation Systems and Algorithms In Chapters 4 and 5, we focused on estimation models where a single parameter was estimated from sensor data. However, more complex scenarios—such as those involving non-linear parameters, multiple parameters, or sensor networks divided into clusters—require the development of new distributed estimation algorithms. In these cases, the system must not only estimate multiple parameters simultaneously but also handle the fact that sensors within each cluster may

be responsible for estimating different objective parameters, adding a layer of complexity to the estimation process.

Additionally, the assumptions underlying the algorithms presented, such as those related to consensus, provided a solid foundation for solving optimization problems in distributed estimation. Nevertheless, relaxing these assumptions could improve the practicality and scalability of the methods, particularly in real-world applications where network conditions are often less ideal and more variable. Future work could focus on exploring more general or flexible assumptions, which could unlock new possibilities for enhancing distributed estimation systems, making them more adaptable to a wider range of settings and network conditions.

Theoretical Analysis of Distributed Estimation Systems In the numerical results presented in Chapter 5, it was observed that the parameter ω , defined in (5.27), effectively reflects the relationship between the variance of convergence speed and the activation probability γ . A promising direction for future work would be to formally establish that ω and the convergence speed exhibit the same monotonicity. Furthermore, demonstrating that ω is a convex function of γ could further simplify the search for the optimal activation rate by enabling the use of convex optimization techniques. Additionally, recent findings on Markov Random Geometric Graphs [108] may offer valuable theoretical insights for analyzing and understanding the properties of the system.

Appendix

APPENDIX A

Supplementary Proof of Theorem 2.2: Full-Rank Property for Submatrices of $\hat{\mathbf{D}}_p$

We aim to prove that submatrices of $\hat{\mathbf{D}}_p$, for $p \in \mathcal{T}_2$, are full-rank with probability 1. Consider any square sub-matrix $\hat{\mathbf{D}}'_p$ of $\hat{\mathbf{D}}_p$ and define the function

$$F \left(\left\{ \mathbf{S}_{p \bmod r}^{(2,k)} : k \in [\tilde{\mathcal{K}}] \setminus \{2\} \right\}, \mathcal{G}_2 \right) \triangleq \det \left(\hat{\mathbf{D}}'_p \right), \quad (\text{A.1})$$

which is a polynomial in the entries of the matrices $\left\{ \mathbf{S}_{p \bmod r}^{(2,k)} : k \in [\tilde{\mathcal{K}}] \setminus \{2\} \right\}$ and \mathcal{G}_2 . According to (2.79), (2.87), (2.88), and (2.92), F is a rational function in the entries of the matrices $\{\mathbf{H}_{p,q}\}$ and $\{\mathbf{S}_\ell^{(i,k)}\}$, where the polynomial in the denominator (which consists of products of determinants of matrices $\tilde{\mathbf{H}}^{(1,2)}$ and $\{\tilde{\mathbf{H}}^{(2,k)} : k \in [\tilde{\mathcal{K}}] \setminus \{2\}\}$) is bounded and non-zero by our assumption that all channel matrices $\tilde{\mathbf{H}}^{(i,k)}$ are invertible. The zero-set of the rational function F is thus of Lebesgue measure 0 unless F is equal to the all-zero function. (This can be seen by noting that the zeros of F are the zeros of the polynomial in its numerator, which have Lebesgue measure 0 except when the polynomial is the all-zero polynomial, i.e., when F is the all-zero function.) Since real and imaginary parts of all entries of matrices $\{\mathbf{H}_{p,q}\}$ and $\{\mathbf{S}_\ell^{(i,k)}\}$ are drawn independently from continuous distributions, we conclude that the function F evaluates to 0 with probability 0 (over the matrices $\{\mathbf{H}_{p,q}\}$ and $\{\mathbf{S}_\ell^{(i,k)}\}$), except for the case where it is the all-zero function.

In the rest of this section, we show that F is not the all-zero function, or equivalently that the determinant of $\hat{\mathbf{D}}'_p$ is non-zero for at least one realization of the random matrices. In fact, we show the stronger statement that for the realizations

$$\tilde{\mathbf{H}}^{(1,2)} = \mathbf{Id}_{\mathcal{T}_r} \quad (\text{A.2a})$$

$$\tilde{\mathbf{H}}^{(2,k)} = \mathbf{Id}_{\mathcal{T}_r}, \quad k \in [\tilde{\mathcal{K}}] \setminus \{2\}, \quad (\text{A.2b})$$

the determinant of $\hat{\mathbf{D}}'_p$ is non-zero with probability 1. To this end, notice that for the realizations in (A.2), for any distinct triple $(\bar{i}, \bar{j}, \bar{k}) \in [\tilde{\mathcal{K}}]^3$ with either $(\bar{i}, \bar{k}) = (1, 2)$ or $\bar{i} = 2$:

$$\tilde{\mathbf{G}}_{\bar{j}}^{(\bar{i}, \bar{k})} = \tilde{\mathbf{H}}^{(\bar{j}, \bar{k})} \mathbf{S}^{(\bar{i}, \bar{k})}, \quad (\text{A.3})$$

which implies that for any $\bar{p} = (\bar{i} - 1)r + \bar{\ell}$ in group $\mathcal{T}_{\bar{i}}$ and $p' = (\bar{j} - 1)r + \bar{\ell}$ in group $\mathcal{T}_{\bar{j}}$, for \bar{i} and \bar{j} as above:

$$\mathbf{G}_{p'}^{(\bar{p}, \bar{k})} = \mathbf{H}_{p', (\bar{k}-1)r + \bar{\ell}} \mathbf{S}_\ell^{(\bar{i}, \bar{k})}, \quad (\text{A.4})$$

because $\mathbf{S}^{(\bar{i}, \bar{k})}$ is diagonal and $\tilde{\mathbf{H}}^{(\bar{j}, \bar{k})}$ consists of r^2 blocks of \mathbf{T} -dimensional block matrices. As a consequence, for the realizations in (A.2), the matrix $\hat{\mathbf{D}}_p$ is given by (A.6) on top of the next page. In the following, we explain in detail that the matrix in (A.6) has the same

$$\begin{aligned} & \hat{\mathbf{D}}_p \Big|_{(A.2)} \\ &= \left[\mathbf{S}_p^{(2,k)} \cdot \prod_{\substack{(\bar{k}, \bar{p}, p') : \bar{p} \in \mathcal{T}_2 \\ \bar{k} \in [\mathbf{K}] \setminus \{2\}, \\ p' \in [\mathbf{K}] \setminus (\mathcal{T}_2 \cup \mathcal{T}_{\bar{k}})}} \left(\mathbf{G}_{p'}^{(\bar{p}, \bar{k})} \right)^{\alpha_{2, (\bar{k}, \bar{p}, p')}}} \right. \\ & \quad \cdot \left. \prod_{\substack{\bar{p} \in \mathcal{T}_1, \\ p' \in [\mathbf{K}] \setminus \{\mathcal{T}_1 \cup \mathcal{T}_2\}}} \left(\mathbf{G}_{\bar{p}}^{(p', 2)} \right)^{\alpha_{2, (\bar{p}, p')}}} \cdot \mathbf{1} : k \in [\tilde{\mathbf{K}}] \setminus \{2\}, \{\alpha_{2, \bar{p}, p'}\}, \{\alpha_{2, (\bar{k}, \bar{p}, p')}\} \in [\eta] \right] \end{aligned} \quad (A.5)$$

$$\begin{aligned} &= \left[\mathbf{S}_p^{(2,k)} \cdot \prod_{\substack{(\bar{k}, \bar{p}, p') : \bar{p} \in \mathcal{T}_2 \\ \bar{k} \in [\mathbf{K}] \setminus \{2\}, \\ p' \in [\mathbf{K}] \setminus (\mathcal{T}_2 \cup \mathcal{T}_{\bar{k}})}} \left(\mathbf{H}_{p', (\bar{k}-1)r + (\bar{p} \bmod r)} \cdot \mathbf{S}_{p' \bmod r}^{(2, \bar{k})} \right)^{\alpha_{2, (\bar{k}, \bar{p}, p')}}} \right. \\ & \quad \cdot \left. \prod_{\substack{\bar{p} \in \mathcal{T}_1, \\ p' \in [\mathbf{K}] \setminus \{\mathcal{T}_1 \cup \mathcal{T}_2\}}} \left(\mathbf{H}_{\bar{p}, r + (p' \bmod r)} \cdot \mathbf{S}_{p' \bmod r}^{[p'/r, 2]} \right)^{\alpha_{2, (\bar{p}, p')}}} \cdot \mathbf{1} : k \in [\tilde{\mathbf{K}}] \setminus \{2\}, \{\alpha_{2, \bar{p}, p'}\}, \{\alpha_{2, (\bar{k}, \bar{p}, p')}\} \in [\eta] \right]. \end{aligned} \quad (A.6)$$

form as matrix \mathbf{A} in Lemma 2.1 at the end of this section. Trivially, then also any square submatrix of $\hat{\mathbf{D}}_p$ has the same form, which by Lemma 2.1 proves that for the realizations in (A.2) the determinant of $\hat{\mathbf{D}}_p'$ is non-zero with probability 1.

To see that $\hat{\mathbf{D}}_p$ is of the form in (2.70), notice that all matrices involved in (A.6) are diagonal, and their multiplications with an all-one vector from the right leads to a column-vector consisting of the non-zero entries of these diagonal matrices. More precisely, the random variables in row t are given by the slot- t channel coefficients $\{H_{q,p}(t)\}$ and the t -th diagonal elements of $\mathbf{S}_\ell^{(i,k)}$, which by definition are independent of each other and of all random variables in the other rows. Therefore, the matrix (A.6) satisfies Condition i) in Lemma 2.1. To see that it also satisfies Condition ii), notice that there is a one-to-one mapping between the columns of $\hat{\mathbf{D}}_p$ and the parameter tuples $\mathbf{v} = (k, \{\alpha_{2, (\bar{k}, \bar{p}, p')}\}, \{\alpha_{2, (\bar{p}, p')}\})$ and that for any two distinct tuples

$$\mathbf{v}^{(1)} = (k^{(1)}, \{\alpha_{2, (\bar{k}, \bar{p}, p')}^{(1)}\}, \{\alpha_{2, (\bar{p}, p')}^{(1)}\}) \text{ and } \mathbf{v}^{(2)} = (k^{(2)}, \{\alpha_{2, (\bar{k}, \bar{p}, p')}^{(2)}\}, \{\alpha_{2, (\bar{p}, p')}^{(2)}\})$$

the exponents in the corresponding columns differ because:

1. If $\alpha_{2, (\bar{p}, p')}^{(1)} \neq \alpha_{2, (\bar{p}, p')}^{(2)}$, then $H_{\bar{p}, r + (p' \bmod r)}$ has different exponents in the two columns.
2. If $\alpha_{2, (\bar{k}, \bar{p}, p')}^{(1)} \neq \alpha_{2, (\bar{k}, \bar{p}, p')}^{(2)}$, then $H_{p', (\bar{k}-1)r + (\bar{p} \bmod r)}$ has different exponents in the two columns.

-
3. If $\alpha_{2,(\bar{p},p')}^{(1)} = \alpha_{2,(\bar{p},p')}^{(2)}$ and $\alpha_{2,(\bar{k},\bar{p},p')}^{(1)} = \alpha_{2,(\bar{k},\bar{p},p')}^{(2)}$, but $k^{(1)} \neq k^{(2)}$, then both $\mathbf{S}_{p \bmod r}^{(2,k^{(1)})}$ and $\mathbf{S}_{p \bmod r}^{(2,k^{(2)})}$ have different exponents in the two columns.

This concludes the proof.

APPENDIX B

Supplementary Proofs of Theorem 3.1

B.1 Proof of Monotonicity and Convexity of values $C_i^{(t)}$

We shall prove monotonicity and convexity of the values

$$D_i^{(t)} \triangleq \frac{C_i^{(t)}}{\binom{K-t}{t}(K-2t)} \quad (\text{B.1})$$

$$= \binom{K-i}{t-i}, \quad i \in [t]. \quad (\text{B.2})$$

The monotonicity can be proven by the recurrence relation of binomial coefficients:

$$D_{i-1}^{(t)} = \binom{K-i+1}{t-i+1} \quad (\text{B.3})$$

$$= \binom{K-i}{t-i} + \binom{K-i}{t-i+1} > D_i^{(t)}. \quad (\text{B.4})$$

To prove convexity, we apply the same recurrence relation to obtain:

$$\begin{aligned} D_{i+1}^{(t)} + D_{i-1}^{(t)} &= 2 \binom{K-i}{t-i} - \binom{K-i-1}{t-i} + \binom{K-i}{t-i+1} \end{aligned} \quad (\text{B.5})$$

$$= 2 \binom{K-i}{t-i} + \binom{K-i-1}{t-i+1} \quad (\text{B.6})$$

$$\geq 2D_i^{(t)}, \quad (\text{B.7})$$

which concludes the proof.

B.2 Proof of Structure of Minimizer

Start with any feasible vector b_1, \dots, b_K and consider two indices $i < j$ with non-zero masses, $b_i > 0$ and $b_j > 0$. Updating this vector as

$$b'_i = b_i - \Delta, \quad \text{and} \quad b'_{i+1} = b_{i+1} + \Delta, \quad (\text{B.8})$$

$$b'_{j-1} = b_{j-1} + \Delta, \quad \text{and} \quad b'_j = b_j - \Delta, \quad (\text{B.9})$$

for any $\Delta \in [0, \min\{b_i, b_j\}]$, results again in a feasible solution vector, which has smaller objective function due to the convexity of the coefficients $\{C_i^{(t)}\}$.

Applying this argument iteratively, one can conclude that there must exist an optimal solution vector where all entries are zero except for two masses $b_k > 0$ and $b_{k+1} \geq 0$. Since $\sum_{i=1}^K ib_i \leq rN$, the index k cannot exceed r . By the decreasing monotonicity of the coefficients $C_i^{(t)}$, the optimal solution must then be to choose $b_{\lfloor r \rfloor} > 0$ and $b_{\lfloor r \rfloor + 1} \geq 0$ and all other masses equal to 0. Since there is a unique such choice satisfying $\sum_{i=1}^K ib_i \leq rN$ and $\sum_{i=1}^K b_i = N$, this concludes the proof.

Bibliography

- [1] J. Wu, *Distributed system design*. CRC press, 2017.
- [2] R. Buyya, J. Broberg, *et al.*, *Cloud computing: Principles and paradigms*. John Wiley & Sons, 2010.
- [3] J. S. Ng, W. Y. B. Lim, *et al.*, “A comprehensive survey on coded distributed computing: fundamentals, challenges, and networking applications”, *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1800–1837, Jun. 2021.
- [4] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters”, *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
- [5] M. Chowdhury, M. Zaharia, *et al.*, “Managing data transfers in computer clusters with orchestra”, *ACM SIGCOMM computer communication review*, vol. 41, no. 4, pp. 98–109, 2011.
- [6] S. Li, M. A. Maddah-Ali, *et al.*, “A fundamental tradeoff between computation and communication in distributed computing”, *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.
- [7] Q. Yan, S. Yang, *et al.*, *Storage-computation-communication tradeoff in distributed computing: fundamental limits and complexity*, 2019.
- [8] F. Xu, S. Shao, *et al.*, “New results on the computation-communication tradeoff for heterogeneous coded distributed computing”, *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2254–2270, Apr. 2021.
- [9] S. Li, M. A. Maddah-Ali, *et al.*, “A unified coding framework for distributed computing with straggling servers”, in *IEEE Globecom Workshops*, Washington DC, USA, Dec. 2016, pp. 1–6.
- [10] J. Zhang and O. Simeone, “Improved latency-communication trade-off for map-shuffle-reduce systems with stragglers”, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8172–8176.
- [11] C. Lin, G. Han, *et al.*, “A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks”, *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5481–5493, 2020.
- [12] M. Mofokeng and T. Matima, “Future tourism trends: virtual reality based tourism utilizing distributed ledger technologies”, *African Journal of Hospitality, Tourism and Leisure*, vol. 7, p. 1, 2018.

-
- [13] P. Ren, X. Qiao, *et al.*, “Edge-assisted distributed dnn collaborative computing approach for mobile web augmented reality in 5g networks”, *IEEE Network*, vol. 34, no. 2, pp. 254–261, Mar. 2020.
- [14] A. S. Hamza, J. S. Deogun, *et al.*, “Wireless communication in data centers: a survey”, *IEEE communications surveys & tutorials*, vol. 18, no. 3, pp. 1572–1595, 2016.
- [15] Y. Xu and H. Qi, “Distributed computing paradigms for collaborative signal and information processing in sensor networks”, *International Journal of Parallel and Distributed Computing*, vol. 64, pp. 945–959, 2004.
- [16] F. J. Ferrández-Pastor, J. M. García-Chamizo, *et al.*, “Precision agriculture design method using a distributed computing architecture on internet of things context”, *Sensors*, vol. 18, no. 6, p. 1731, Jun. 2018.
- [17] K. Fauvel, D. Balouek-Thomert, *et al.*, “A distributed multi-sensor machine learning approach to earthquake early warning”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 403–411, Apr. 2020.
- [18] S. Gault, “Improving MapReduce performance on clusters”, Ph.D. dissertation, Ecole normale supérieure de lyon - ENS LYON, Mar. 2015.
- [19] M. Attia and R. Tandon, “Information theoretic limits of data shuffling for distributed learning”, *arXiv:1609.05181 [cs, math]*, Sep. 2016.
- [20] Q. Yu, M. Maddah-Ali, *et al.*, “Polynomial codes: an optimal design for high-dimensional coded matrix multiplication”, in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [21] K. Lee, M. Lam, *et al.*, “Speeding up distributed machine learning using codes”, *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.
- [22] S. Li, Q. Yu, *et al.*, “A scalable framework for wireless distributed computing”, *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2643–2654, Oct. 2017.
- [23] K. Yang, Y. Shi, *et al.*, “Data shuffling in wireless distributed computing via low-rank optimization”, *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3087–3099, Jun. 2019.
- [24] A. Paris, H. Mirghasemi, *et al.*, “Energy-efficient edge-facilitated wireless collaborative computing using Map-Reduce”, in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jul. 2019, pp. 1–5.
- [25] K. Yuan and Y. Wu, “Coded wireless distributed computing via interference alignment”, in *Proc. IEEE ISIT*, Espoo, Finland, 2022.
- [26] F. Li, J. Chen, *et al.*, “Wireless MapReduce distributed computing”, *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6101–6114, Oct. 2019.
-

- [27] Y. Bi, P. Ciblat, *et al.*, “DoF of a cooperative X-channel with an application to distributed computing”, in *2022 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2022, pp. 566–571.
- [28] J. Hachem, U. Niesen, *et al.*, “Degrees of freedom of cache-aided wireless interference networks”, *IEEE Trans. Inf. Th.*, vol. 64, no. 7, pp. 5359–5380, Jul. 2018.
- [29] S. Ha, J. Zhang, *et al.*, “Wireless Map-Reduce distributed computing with full-duplex radios and imperfect CSI”, *arXiv:1810.10875 [cs, math]*, Oct. 2018.
- [30] S. Jafar and S. Shamai, “Degrees of freedom region of the MIMO X-channel”, *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.
- [31] V. R. Cadambe and S. A. Jafar, “Can feedback, cooperation, relays and full duplex operation increase the degrees of freedom of wireless networks?”, in *IEEE International Symposium on Information Theory*, Toronto, Canada, Jul. 2008, pp. 1263–1267.
- [32] K. Gomadam, V. R. Cadambe, *et al.*, “A distributed numerical approach to interference alignment and applications to wireless interference networks”, *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3309–3322, Jun. 2011.
- [33] Y. Wei and T. Lok, “An iterative interference alignment algorithm for the general MIMO X channel”, *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1847–1859, Mar. 2019.
- [34] M. Guillaud and D. Gesbert, “Interference alignment in the partially connected k-user mimo interference channel”, in *2011 19th European Signal Processing Conference*, 2011, pp. 1095–1099.
- [35] Y. Wang and J. Yuan, “A fast convergence iterative interference alignment algorithm for partially connected mimo interference channel”, in *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, Changchun, China: IEEE, May 2022, pp. 1056–1059.
- [36] J. Chen and P. Elia, “Toward the performance versus feedback tradeoff for the two-user MISO broadcast channel”, *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8336–8356, 2013.
- [37] J. Zhang, D. Slock, *et al.*, “Achieving the DoF limits of the SISO X channel with imperfect-quality CSIT”, in *2015 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2015, pp. 626–630.
- [38] S. Ha, J. Zhang, *et al.*, “Coded federated computing in wireless networks with straggling devices and imperfect csi”, in *IEEE International Symposium on Information Theory*, Paris, France, Jul. 2019, pp. 2649–2653.
- [39] P. de Kerret, D. Gesbert, *et al.*, “Optimal DoF of the K-user broadcast channel with delayed and imperfect current csit”, *IEEE Transactions on Information Theory*, vol. 66, no. 11, pp. 7056–7066, 2020.
- [40] K. Wan, H. Sun, *et al.*, “Distributed linearly separable computation”, *IEEE Transactions on Information Theory*, vol. 68, no. 2, pp. 1259–1278, Feb. 2022.

-
- [41] Y. Yao and S. A. Jafar, “The capacity of 3 user linear computation broadcast”, *IEEE Transactions on Information Theory*, vol. 70, no. 6, pp. 4414–4438, Jun. 2024.
- [42] Y. Yao and S. A. Jafar, “On the generic capacity of K-user symmetric linear computation broadcast”, *IEEE Transactions on Information Theory*, vol. 70, no. 5, pp. 3693–3717, May 2024.
- [43] F. Pasqualetti, R. Carli, *et al.*, “Distributed estimation via iterative projections with application to power network monitoring”, *Automatica*, vol. 48, no. 5, pp. 747–758, May 2012.
- [44] X. Ge, Q.-L. Han, *et al.*, “Distributed event-triggered estimation over sensor networks: a survey”, *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1306–1320, Mar. 2020.
- [45] S. He, H.-S. Shin, *et al.*, “Distributed estimation over a low-cost sensor network: a review of state-of-the-art”, *Information Fusion*, vol. 54, pp. 21–43, Feb. 2020.
- [46] A. Ribeiro and G. Giannakis, “Bandwidth-constrained distributed estimation for wireless sensor networks-part I: Gaussian case”, *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.
- [47] P. Stoica, X. Shang, *et al.*, “The Cramér–Rao bound for signal parameter estimation from quantized data [lecture notes]”, *IEEE Signal Processing Magazine*, vol. 39, no. 1, pp. 118–125, Jan. 2022.
- [48] A. Sani and A. Vosoughi, “On performance limit for bandwidth-constrained distributed parameter estimation systems”, in *Conference on Information Sciences and Systems (CISS)*, Mar. 2022, pp. 78–83.
- [49] Y. Han, A. Ozgur, *et al.*, “Geometric lower bounds for distributed parameter estimation under communication constraints”, *IEEE Transactions on Information Theory*, vol. 67, no. 12, pp. 8248–8263, Dec. 2021.
- [50] E. Meir and T. Routtenberg, “Cramér-Rao bound for estimation after model selection and its application to sparse vector estimation”, *IEEE Transactions on Signal Processing*, vol. 69, pp. 2284–2301, 2021.
- [51] L. Liu, C. Hua, *et al.*, “Greedy sensor selection: leveraging submodularity based on volume ratio of information ellipsoid”, *IEEE Transactions on Signal Processing*, vol. 71, pp. 2391–2406, 2023.
- [52] L. P. Barnes, Y. Han, *et al.*, “Lower bounds for learning distributions under communication constraints via Fisher information”, *Journal of Machine Learning Research*, vol. 21, no. 236, pp. 1–30, 2020.
- [53] J. C. Duchi, M. I. Jordan, *et al.*, “Optimality guarantees for distributed statistical estimation”, *arXiv preprint arXiv:1405.0782*, 2014.
- [54] A. Mezghani, F. Antreich, *et al.*, “Multiple parameter estimation with quantized channel output”, in *ITG Workshop on Smart Antennas (WSA)*, Feb. 2010.
-

- [55] Y. Cheng, X. Shang, *et al.*, “Interval design for signal parameter estimation from quantized data”, *IEEE Transactions on Signal Processing*, vol. 70, pp. 6011–6020, 2022.
- [56] F. Gao, L. Guo, *et al.*, “Quantizer design for distributed GLRT detection of weak signal in wireless sensor”, *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2032–2042, Apr. 2015.
- [57] S. Kar, H. Chen, *et al.*, “Optimal identical binary quantizer design for distributed estimation”, *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3896–3901, Jul. 2012.
- [58] P. Venkitasubramaniam, L. Tong, *et al.*, “Quantization for Maximin ARE in Distributed Estimation”, *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3596–3605, Jul. 2007.
- [59] A. Kipnis and J. Duchi, “Mean estimation from one-bit measurements”, *IEEE Transactions on Information Theory*, vol. 68, no. 9, pp. 6276–6296, Sep. 2022.
- [60] C.-Y. Chong, K.-C. Chang, *et al.*, “A review of forty years of distributed estimation”, in *2018 21st International Conference on Information Fusion (FUSION)*, Jul. 2018, pp. 1–8.
- [61] S. Boyd, A. Ghosh, *et al.*, “Randomized gossip algorithms”, *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [62] T. C. Aysal, M. E. Yildiz, *et al.*, “Broadcast gossip algorithms for consensus”, *IEEE Transactions on Signal processing*, vol. 57, no. 7, pp. 2748–2761, 2009.
- [63] D. Antunes, D. Silvestre, *et al.*, “Average consensus and gossip algorithms in networks with stochastic asymmetric communications”, in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2011, pp. 2088–2093.
- [64] H. Wang, X. Liao, *et al.*, “Average consensus in sensor networks via broadcast multi-gossip algorithms”, *Neurocomputing*, vol. 117, pp. 150–160, Oct. 2013.
- [65] F. Iutzeler, P. Ciblat, *et al.*, “Analysis of sum-weight-like algorithms for averaging in wireless sensor networks”, *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2802–2814, Jun. 2013.
- [66] R. Olfati-Saber, “Distributed kalman filter with embedded consensus filters”, in *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 8179–8184.
- [67] R. Carli, A. Chiuso, *et al.*, “Distributed Kalman filtering based on consensus strategies”, *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, May 2008.
- [68] G. Battistelli and L. Chisci, “Stability of consensus extended kalman filter for distributed state estimation”, *Automatica*, vol. 68, pp. 169–178, 2016.

-
- [69] A. Mohammadi and A. Asif, “Distributed consensus + innovation particle filtering for bearing/range tracking with communication constraints”, *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 620–635, 2014.
- [70] Y. Liu, M. Coombes, *et al.*, “Mesh-based consensus distributed particle filtering for sensor networks”, *IEEE Transactions on Signal and Information Processing over Networks*, vol. 9, pp. 346–356, 2023.
- [71] S. Boyd, A. Ghosh, *et al.*, “Analysis and optimization of randomized gossip algorithms”, in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, IEEE, vol. 5, 2004, pp. 5310–5315.
- [72] S. Wu and M. G. Rabbat, “Broadcast gossip algorithms for consensus on strongly connected digraphs”, *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 3959–3971, 2013.
- [73] D. Silvestre, J. P. Hespanha, *et al.*, “Broadcast and gossip stochastic average consensus algorithms in directed topologies”, *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 474–486, Jun. 2019.
- [74] S. Kar and J. M. Moura, “Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise”, *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2008.
- [75] B. Gerencsér and J. M. Hendrickx, “Push-sum with transmission failures”, *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1019–1033, Mar. 2019.
- [76] H. Wang, X. Liao, *et al.*, “Distributed parameter estimation in unreliable sensor networks via broadcast gossip algorithms”, *Neural Networks*, vol. 73, pp. 1–9, 2016.
- [77] F. Ferrari, M. Zimmerling, *et al.*, “Efficient network flooding and time synchronization with glossy”, in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IEEE, 2011, pp. 73–84.
- [78] M. Zimmerling, L. Mottola, *et al.*, “Synchronous transmissions in low-power wireless: a survey of communication protocols and network services”, *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–39, 2020.
- [79] J. Debadarshini, M. Tummala, *et al.*, “Timecast: real-time many-to-many data-sharing in low-power wireless distributed systems”, *IEEE Systems Journal*, vol. 17, no. 4, pp. 5726–5737, 2023.
- [80] M. Khatiwada and S. W. Choi, “On the interference management for k-user partially connected fading interference channels”, *IEEE Transactions on Communications*, vol. 60, no. 12, pp. 3717–3725, Dec. 2012.
- [81] G. Xu, “Global optimization of signomial geometric programming problems”, *European Journal of Operational Research*, vol. 233, no. 3, pp. 500–510, Mar. 2014.
- [82] F. Iutzeler, “Distributed estimation and optimization in asynchronous networks”, Ph.D. dissertation, Télécom ParisTech, 2013.
-

- [83] B. Gerencsér and L. Gerencsér, “Tight bounds on the convergence rate of generalized ratio consensus algorithms”, *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1669–1684, 2021.
- [84] P. Algoet and J. Cioffi, “The capacity of a channel with Gaussian noise and intersymbol interference”, in *IEEE International Symposium on Information Theory*, Budapest, Hungary, Jun. 1991, p. 16.
- [85] G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment when using multiple antennas”, *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, Mar. 1998.
- [86] A. Lapidoth, S. Shamai, *et al.*, “On cognitive interference networks”, in *IEEE Information Theory Workshop*, Tahoe City, CA, USA, Sep. 2007, pp. 325–330.
- [87] N. Devroye and M. Sharif, “The multiplexing gain of MIMO X-channels with partial transmit side-information”, in *IEEE International Symposium on Information Theory*, Nice, France, Jun. 2007, pp. 111–115.
- [88] V. R. Cadambe and S. A. Jafar, “Interference alignment and degrees of freedom of the K-user interference channel”, *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.
- [89] V. R. Cadambe and S. A. Jafar, “Interference alignment and the degrees of freedom of wireless X networks”, *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3893–3908, Sep. 2009.
- [90] V. S. Annapureddy, A. El Gamal, *et al.*, “Degrees of freedom of interference channels with CoMP transmission and reception”, *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5740–5760, Sep. 2012.
- [91] A. S. Motahari, S. Oveis-Gharan, *et al.*, “Real interference alignment: exploiting the potential of single antenna systems”, *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4799–4810, Aug. 2014.
- [92] M. Zamanighomi and Z. Wang, “Degrees of freedom region of wireless X networks based on real interference alignment”, *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1931–1941, Apr. 2016.
- [93] A. F. Aji and K. Heafield, “Sparse communication for distributed gradient descent”, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, *et al.*, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 440–445.
- [94] R. S. Dembo, “A set of geometric programming test problems and their solutions”, *Mathematical Programming*, vol. 10, no. 1, pp. 192–213, 1976.
- [95] A. Mutapcic, K. Koh, *et al.*, *Ggplab version 1.00: a matlab toolbox for geometric programming*, 2006.
- [96] S. Boyd, S.-J. Kim, *et al.*, “A tutorial on geometric programming”, *Optimization and engineering*, vol. 8, pp. 67–127, 2007.

-
- [97] M. Grant, S. Boyd, *et al.*, “CVX users’ guide”, *online: <http://www.stanford.edu/boyd/software.html>*, 2009.
- [98] M. Avriel and A. C. Williams, “Complementary geometric programming”, *SIAM Journal on Applied Mathematics*, vol. 19, no. 1, pp. 125–141, 1970.
- [99] M. Chiang *et al.*, “Geometric programming for communication systems”, *Foundations and Trends[®] in Communications and Information Theory*, vol. 2, no. 1–2, pp. 1–154, 2005.
- [100] B. R. Marks and G. P. Wright, “A general inner approximation algorithm for non-convex mathematical programs”, *Operations research*, vol. 26, no. 4, pp. 681–683, 1978.
- [101] H. A. David and H. N. Nagaraja, *Order Statistics*. Wiley Series in Probability and Statistics, 2003.
- [102] M. Avriel and A.-C. Williams, “Complementary geometric programming”, *SIAM Journal of Applied Mathematics*, vol. 19, pp. 125–141, 1970.
- [103] H. Wang, X. Liao, *et al.*, “Distributed parameter estimation in unreliable sensor networks via broadcast gossip algorithms”, *Neural Networks*, vol. 73, pp. 1–9, Jan. 2016.
- [104] S. Yang, X. Zhang, *et al.*, “Decentralized gossip-based stochastic bilevel optimization over communication networks”, *Advances in neural information processing systems*, vol. 35, pp. 238–252, 2022.
- [105] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.
- [106] D. Kempe, A. Dobra, *et al.*, “Gossip-based computation of aggregate information”, in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, Cambridge, MA, USA: IEEE Computer. Soc, 2003, pp. 482–491.
- [107] F. Bénézit, V. Blondel, *et al.*, “Weighted gossip: distributed averaging using non-doubly stochastic matrices”, in *2010 IEEE International Symposium on Information Theory*, Jun. 2010, pp. 1753–1757.
- [108] Q. Duchemin and Y. De Castro, “Random geometric graph: some recent developments and perspectives”, *High Dimensional Probability IX: The Ethereal Volume*, pp. 347–392, 2023.

Publications

Journal papers

- J1. **Y. Bi**, M. Wigger, Y. Wu, "Normalized Delivery Time of Wireless MapReduce", in *IEEE Transaction on Information Theory*, vol. 70, no. 10, pp. 7005-7022, Oct. 2024, doi: 10.1109/TIT.2024.3423710.
- J2. Z. Huang, K. Yuan, S. Ma, **Y. Bi** and Y. Wu, "Coded Computing for Half-Duplex Wireless Distributed Computing Systems via Interference Alignment," in *IEEE Transactions on Wireless Communications*, vol. 23, no. 11, pp. 17399-17414, Nov. 2024, doi: 10.1109/TWC.2024.3453403.

International conferences

- C1. **Y. Bi**, P. Ciblat, M. Wigger, and Y. Wu, "DoF of a Cooperative X-Channel with an Application to Distributed Computing," in *2022 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2022.
- C2. **Y. Bi**, M. Wigger, Y. Wu, "A New Interference-Alignment Scheme for Wireless MapReduce", in *2023 IEEE Global Communications Conference (Globecom)*, Dec. 2023.
- C3. **Y. Bi**, Y. Wu, C. Hua, "DoF Analysis for (M, N)-Channels through a Number-Filling Puzzle", in *2024 IEEE International Symposium on Information Theory (ISIT)*, Jul. 2024.
- C4. **Y. Bi**, P. Ciblat, Y. Wu, C. Hua, "Optimal thresholding for multi-bit distributed estimation", submitted to *2025 IEEE Statistical Signal Processing Workshop (SSP)*.
- C5. **Y. Bi**, P. Ciblat, Y. Wu, C. Hua, "Stochastic Activation based Broadcast Push-Sum for Distributed Estimation", submitted to *2025 IEEE Statistical Signal Processing Workshop (SSP)*.

French conferences

- C6. **Y. Bi**, P. Ciblat, "A propos de bornes de Cramer-Rao pour l'estimation distribuée", *XXIXème Colloque Francophone de Traitement du Signal et des Images (GRETSI)*, Aug. 2023.

Titre : Limites fondamentales et algorithmes pratiques pour les systèmes de calcul et d'estimation distribués sans fil.

Mots clés : Théorie de l'information, Traitement distribué, Théorie de l'estimation

Résumé : Les systèmes distribués sont au cœur des applications informatiques modernes, permettant l'exécution collaborative des tâches sur des composants interconnectés. Cependant, leur nature répartie pose des défis majeurs en matière de communication. Cette thèse étudie ces enjeux en analysant les limites théoriques de l'information et en proposant des solutions pour améliorer les performances des systèmes de calcul distribué (DC) et d'estimation distribuée.

Dans les systèmes DC, la parallélisation des tâches réduit les temps d'exécution, mais la phase shuffle reste un goulot d'étranglement, en particulier dans les réseaux sans fil. Cette thèse introduit de nouveaux schémas de codage pour optimiser le compromis calcul-communication dans ces environnements, en exploitant l'alignement des interférences (IA) et en

établissant des bornes théoriques.

Concernant l'estimation distribuée, où plusieurs nœuds collaborent pour estimer un paramètre commun, deux scénarios sont explorés : avec et sans centre de fusion. Dans le premier cas, un cadre optimisant la quantification multi-bits est proposé pour minimiser la borne de Cramér-Rao. Dans le second, un algorithme synchrone avec activation stochastique est développé pour améliorer la convergence tout en réduisant les collisions de données.

En résumé, cette thèse approfondit la compréhension des limites théoriques et propose des stratégies de codage adaptées aux systèmes distribués, améliorant ainsi leur efficacité et leur robustesse dans divers environnements.

Title : Fundamental Limits and Practical Algorithms for Wireless Distributed Computation and Estimation Systems.

Keywords : Information theory, Distributed processing, Estimation theory

Abstract : Distributed systems are at the core of modern computing applications, enabling collaborative task execution across interconnected components. However, their distributed nature presents major challenges in communication efficiency. This thesis addresses these challenges by analyzing the theoretical limits of information and proposing solutions to enhance the performance of distributed computing (DC) and distributed estimation systems.

In DC systems, task parallelization significantly reduces execution time, but the shuffle phase remains a bottleneck, particularly in wireless networks. This thesis introduces new coding schemes to optimize the computation-communication tradeoff in such environ-

ments, leveraging interference alignment (IA) and establishing theoretical bounds.

Regarding distributed estimation, where multiple nodes collaborate to estimate a common parameter, two scenarios are explored: with and without a fusion center. In the first case, a framework is proposed to optimize multi-bit quantization and minimize the Cramér-Rao bound. In the second, a synchronous algorithm with stochastic activation is developed to improve convergence while reducing data collisions.

In summary, this thesis deepens the understanding of theoretical limits and proposes practical coding strategies for distributed systems, enhancing their efficiency and robustness across various environments.