# Efficient Methods for Optimizing Decentralized Multi-Agent Systems

by

Hsu Kao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2022

Doctoral Committee:

Associate Professor Vijay Subramanian, Chair
Professor Mingyan Liu
Associate Professor Aditya Mahajan, McGill University
Associate Professor Dimitra Panagou
Professor Lei Ying

Hsu Kao

hsukao@umich.edu

ORCID iD: 0000-0001-7694-5869

我們貢獻這個大學于宇宙的精神

We contribute to this university the spirit of the universe

# ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor Vijay Subramanian for giving me the opportunity to study at UM and for all the guidance he gave me over these years. I am particularly grateful for his patience with me, his always putting my education at the highest priority, and shaping my rigorous attitude towards scholarship. I would also like to acknowledge my other committee members: Mingyan Liu, Aditya Mahajan, Dimitra Panagou, and Lei Ying, for providing valuable and constructive feedback on the thesis. A big Thank You to all the faculty members at UM who have helped me in my PhD journey. Among all, Demosthenis Teneketzis has been a great mentor and an academic role model for me. From two of his excellent courses, I learnt the foundational basics for my research work.

I want to thank Chen-Yu for all the helpful discussions when I was stuck in technical details, and the collaboration that gave rise to Chapter V of the thesis. I would also like to thank my roommate and colleague Shih-Tang for putting up with a night owl roommate and the dining table jibber-jabbers. Thanks are also due to my outstanding colleagues Daniel, Dengwang, and Mehrdad, and the two newcomers, Nouman and Saghar. I benefited a lot from the reading groups and taking the courses with them.

Thank You to all those who had inspired me to start this journey, my parents, my sister, and my high school and undergraduate instructors: Mei-Li Wang, Chung-Hsien Li, Ching-Liang Lin, Lin-Shan Lee, Borching Su, Tian-Li Yu, and Wanjiun Liao. My warmest thanks to my family, especially my sister, and my dearest friends Chen-Yu and Fu-Yao who have always been supportive of me during my PhD life.

For all people whom I owe thanks to but not mentioned above, I will quote what Zhifan Chen wrote in his prose: “The people that I have to express gratitude to are so many, so I thank heaven (Xie Tian).”

# TABLE OF CONTENTS

**CHAPTER**

# LIST OF FIGURES

# LIST OF TABLES

**Table**

# LIST OF APPENDICES

**Appendix**

# ABSTRACT

Applications of decentralized multi-agent systems are ubiquitous in the present day, including autonomous driving, gaming AI, sensor networks, etc. Depending on the setting of an application problem, the theoretical framework behind the problem commonly used is either distributed optimization, decentralized control, or multi-agent reinforcement learning (MARL). However, existing methods may not be scalable, may require strong but limiting assumptions, or may need solutions to open problems that remain unresolved when put into practice. In this thesis, we aim to develop more general and efficient computational algorithms for optimizing such systems that better fit with the applications, where the efficiency we are concerned with is in terms of memory storage, communication overhead, convergence speed, and low regret. On the distributed optimization side, we propose a localization scheme that exploits partial dependency structure in the objective functions to save memory and communication required, along with a convergence rate analysis of the proposed scheme. We also study an approximation framework that allows us to relax the Lipschitz gradient assumption commonly made in the literature, and investigate nonlinear consensus schemes from a stochastic approximation point of view. To overcome the non-stationarity issue and the curse of dimensionality in MARL problems, we devise algorithms that achieve near-optimal regret for a class of problems with a hierarchical information structure; for even more general settings, we extend the notion of an approximate information state to certain multi-agent cases that can be used to design scalable and low regret learning algorithms without the knowledge of the models.

# CHAPTER I

# Introduction

## 1.1 Motivation

Application of decentralized multi-agent systems (MASs) are ubiquitous in the present day, including autonomous driving [99], Internet of Things (IoT) [75], resource scheduling and allocation in telecommunication [45], wireless sensor networks [36], and designing artificial intelligence (AI) algorithms for strategic games [26], just to name a few. Consisting of autonomous entities known as agents, decentralized MASs are capable of achieving complex tasks with great flexibility, as the agents interact with neighbors and environments and make decisions [38]. Such a concept is made possible not only due to the explosive growth of the number of computational devices [106] led by the breakthroughs in embedded electronics, but also the recent huge success in the field of statistical and computational learning theory, including distributed optimization, machine learning (ML), reinforcement learning (RL), and deep learning (DL).

Depending on the setting of an application problem, the theoretical framework behind the problem commonly falls into one of distributed optimization, decentralized stochastic control, multi-agent reinforcement learning (MARL), or federated learning; some of the problems also belong to the game-theoretic framework, especially problems in decentralized control and MARL [48]. In distributed optimization framework, the agents knowing only their own objective functions collectively optimize the sum of the objective functions of all the agents. The theory of decentralized stochastic control concerns finding the optimal policies of a multi-stage optimization problem when the model, which is usually Markovian, is known [82]; on the other hand, in an MARL setting, the agents try to find the same optimal policies without the knowledge of the model, by learning and interacting with the environment which the model characterizes. Federated learning can be thought of as a decentralized ML scheme that minimizes an appropriate loss function without sharing private data; in the scheme, the agents try to find a centralized model that best fits the training data which is stored at the individual agents' sides. In a game-theoretic framework, the agents may have different objectives and may move strategically against each other to maximize their welfare; however, this framework will not be considered in this thesis as we focus on cooperating agents that work towards a common goal.

All of these frameworks share many similarities. First, all of them involve decision making to reach an appropriate optimual value for a common objective (when the focus in on the fully-cooperative cases as in this thesis): in distributed optimization, the agents choose the decision variable to minimize the sum of their objective functions; in decentralized stochastic control, the agents decide a joint policy that maximizes the long-term reward; in MARL, the planner designs an algorithm whose output converges to an optimal joint policy asymptotically while minimizing the regret; and in federated learning, the agents search for a model parameter that minimizes the loss function whilst adhering to privacy requirements. Second, most computations are carried out on the agents' side, with the agents coordinating or cooperating, and with certain information sharing schemes. Third, in these frameworks, each agent holds a dynamic state, and updates the state in each time step with the incoming information received from the neighbors or interactions with the environment [131]. Federated learning is conceptually a distributed optimization problem if the local loss function defined by the local training data (sometimes along with a loss of privacy penalty) is viewed as the objective function, and the model parameter is viewed as the decision variable [124]. If the policies are parametrized, which is a popular approach in RL [115], then finding an optimal policy is equivalent to deciding the parameter that maximizes the (long-term) reward [44]; thus, distributed optimization and decentralized stochastic control are also related in this sense.

Although these frameworks are widely studied in the literature, gaps still exist when they are put into practice. There are two fundamental distinct features in decentralized MASs that make the setting much more complicated than their centralized counterparts: one is information asymmetry [88], and another is the prevalence of signaling [127, 41]. The former refers to the fact that agents possess private information unseen by other agents; this creates a non-classical information structure where information no longer builds on itself over time, and hinders the agents from using existing centralized sequential decomposition methods [127] such as dynamic programming. On the other hand, signaling refers to agents using observable information from other agents, usually communications or actions, to infer their unobserved information; agents may reason either through complex multi-level beliefs and rationality driven concerns [127, 6] or the theory of mind when observing others' actions [42], both of which are hard for analysis and also computation. These fundamental aspects lead to several challenges for decentralized MASs. One of the crucial challenges is the scalability issue, both in the dimensions of time and number of agents; this issue manifests via its impact on memory storage, communication overhead, computational complexity, etc. A notable example of an infeasible requirement of computational complexity is the doubly exponential growth complexity in time for MARL problems, which basically forbids practical implementations of most of the existing results in decentralized stochastic control or its MARL counterpart; this is an important open problem in the field. Moreover, some of the existing methods may require strong but limiting assumptions that do not exactly hold in practice. For example, in real-world applications, the communications between the agents or the timing of their actions may be subject to delay and inaccuracies, which gives rise to the synchronization issue and the quantization issue [138, 90, 105]; further, transmission errors can also result in information asymmetry between the agents. Some good properties such as Markovian structure or smoothness assumptions also may not hold in applications. Consensus formation is yet another core topic in decentralized MASs. The agents may have difficulties reaching a consensus due to information asymmetry, which leads to the non-stationarity issue in the MARL setting when they try to learn concurrently (which will be explained in detail in Section 4.2.2); even in frameworks of distributed optimization and federated learning, the means to reach consensus efficiently and methods to alternate consensus with the optimization step at the right frequency are active research topics [13, 14, 125].

In this thesis, our goal is to develop more general and efficient computational algorithms for optimizing decentralized MASs that better fit with the applications, where the efficiency we are concerned with is in terms of memory storage, communication overhead, convergence speed, and low regret. Intuitively, the theoretical frameworks are concerned with algorithms that solve certain optimization problems for the original objectives, and we are then optimizing these optimization and control algorithms for another set of metrics so that they address more concerns of practitioners.

## 1.2 Background and Scope of the Thesis

This thesis is organized into two parts. The first part, constituting Chapter II and Chapter III, considers the distributed optimization framework; the second part, containing Chapter IV and Chapter V, considers the decentralized stochastic control and the MARL frameworks.

### 1.2.1 Distributed Optimization

In a distributed optimization problem, $N$ agents work collaboratively to solve the optimization problem $F(\mathbf{x}) \triangleq \sum_{i=1}^{N} f_i(\mathbf{x})$ subject to certain constraint set; the variable $\mathbf{x}$ is called the decision variable, and agent $i$'s objective function $f_i$ is only known to itself but not other agents. This setup dates back to the seminal work of [122] and has received extensive attention since then.

Although there are innumerable variants of distributed optimization algorithms that have been proposed, most of them can be categorized as either primal-based methods or dual-based methods [94, 80]. In primal-based methods, each agent usually keeps a local copy of the decision variable and performs a local optimization or descent step followed by a consensus step iteratively. In the local optimization step, the agent only uses the local objective function $f_i$ plus information gathered from the neighbors to update its local copy of the decision variable; some examples of this step include gradient-based or sub-gradient-based schemes as local gradient descent [91, 55, 17], second-order-based schemes [76, 83], and successive convex approximation (SCA) [80]. Then in the consensus step, the agents exchange their local copies of the decision variable to reach a consensus through a gossip-type averaging scheme. On the other hand, dual-based methods including the renowned Alternating Direction Method of Multipliers (ADMM) [24], first introduce the (augmented) Lagrangian of the original optimization problem, and update the local copies of the decision variables as well as the Lagrangian multipliers. The original constraint and the consensus constraint are taken care of through the updates of the multipliers, and under suitable regularity assumptions, the optimal value of the dual problem will be the same as that of the original optimization problem [94]. In some algorithms, mostly primal-based, each agent maintains another variable copy $\mathbf{y}$ that tracks the current value of the total gradient $\sum_{i=1}^{N} \nabla f_i(\cdot)$ [80, 94], which can be used to construct a more accurate surrogate function of $F$ and reportedly enhance the convergence rate [114]. In this thesis, for concreteness and tractability, we only focus on two primal-based algorithms, which is the Distributed Gradient Descent (DGD) algorithm [91], and the in-Network succEssive conveX approximaTion (NEXT) algorithm [80].

A bulk of literature in distributed optimization focuses on convergence analysis, such as studying various acceleration methods [34, 134] and regularity conditions, whereas addressing the scalability issue and the practicability issue mentioned in Section 1.1 is equally crucial for applying these algorithms in real systems. These issues include topics on synchronization [138], communication efficiency [74], memory efficiency, coordination scheme, variance reduction [74], explicit or implicit assumptions made for the algorithms, etc., and some of them are overlooked in the literature. Take memory efficiency as an example, existing algorithms require every agent to keep a copy of the entire decision variable; although memory requirement is less constrained nowadays, this scheme is still not scalable: e.g., the frequent exchange of many variables can congest communication links. In this thesis, we improve the communication and memory efficiency with a "localization" technique (Chapter II). We also relax the smoothness assumptions on objective functions commonly made in the literature, namely that they have Lipschitz gradients, so that the algorithms can be used more broadly (Chapter II). In addition, we look at the coordination scheme by studying a wide range of consensus methods and their convergence implications (Chapter III). Last but not least, while the communication graph is usually assumed to be given, we are the first to propose it can be a design parameter as well, and to investigate its relations to memory requirement, communication requirement, and convergence rate

(Section 2.5).

### 1.2.2 Decentralized Stochastic Control and MARL

In a decentralized stochastic control problem, $N$ agents also known as the controllers in this context, choose control policies $g \triangleq (g^1, \ldots, g^N)$ that generate a control process from an observation process, while the underlying state of the dynamical system evolves [82]. Certain Markovian structure is often assumed in the model; in particular, the decentralized partially observable Markov decision process (Dec-POMDP) is mostly used in decentralized stochastic control and MARL problems. The goal, then, is to find a policy $g$ that maximizes the expected long-term (discounted) reward; such a $g$ is called an optimal policy and is denoted as $g^*$, and the optimal long-term reward $g^*$ achieves is denoted as $V^*$.

In centralized stochastic control, in particular with the POMDP model, it is well known that with the *belief state*, the posterior distribution of the underlying state given the history information, an optimal policy can be found using dynamic programming (DP) [82]. The problem becomes significantly more complex in decentralized cases, due to the lack of a centralized controller and the coupled dynamics of different controllers. The person-by-person (PBP) [104, 50, 16] approach tries to solve the problem in the following manner. At any given time, we fix all the control policies $g^{-n} \triangleq (g^1, \ldots, g^{n-1}, g^{n+1}, \ldots, g^N)$ except that of agent $n$; then, the underlying dynamics along with the given $g^{-n}$ forms an effective POMDP for agent $n$, and it can find its optimal policy that best responds to $g^{-n}$, which is denoted as $g^{n*}(g^{-n})$. We iterate this through the agents in a round-robin manner, each time fixing all except one agent. Then at some point of time, we may find that no agent can further increase the long-term reward by unilaterally deviating from its current policy; then we call the current policy profile $g$ a PBP optimal policy, which results in a Nash Equilibrium (NE) [82]. A PBP optimal policy is only locally optimal, and by no means is it guaranteed to be globally optimal; whereas it is true that the team optimal policy is also PBP optimal, in fact, it is easy to see that the values of other PBP optimal policies can be arbitrarily worse than the global optimal value. The designer's approach [128], on the other hand, solves the open-loop centralized stochastic problem from the designer's perspective who knows the state transition and observation kernels (or equivalently the distribution of the primitive random variables). This approach does yield optimal control policies; however, the solution space grows rapidly with the number of time steps, as the history information or the number of primitive random variables the policy may depend on grows. The common information (CI) approach [88] divides all history information into common information and private information, and incorporates policies' dependence on the private information as a new function called the "prescription function;" using the approach, one can construct an equivalent POMDP and obtain a sequential decomposition that gives the optimal policy. A detailed summary of the CI approach is given in Section 4.2.4.1. At this point, the problem of decentralized stochastic control can be considered solved from a theoretical/conceptual perspective. However, there continues to be significant effort in the research community in solving Dec-POMDPs and even solving POMDPs, as [88] is a structural result and does not really provide an algorithmic framework to solve general practical problems; in particular, how to form the belief state (or equivalent information state), and how to deal with the explosive growth of the prescriptions space remain open problems that are not entirely overcome yet.

The MARL setting is basically the decentralized stochastic control setting without the knowledge of the model; here, the model can be understood as the state transition and observation kernels in the environment. The problem goal is also different – in an MARL setting, the agents are put into the environment commonly modeled by Dec-POMDP, to interact with and earn reward from it; hence, there is an intrinsic urgency in finding a policy $g$ that achieves $V^*$, or near $V^*$. Formally, within a time horizon $T$ or a finite number of "episodes" (each containing time steps equal to the time horizon), the goal of MARL is to find an algorithm that minimizes the "regret," which is the difference between the actual earned long-term reward and the highest reward that could possibly be earned in hindsight, i.e. the long-term reward obtained by $g^*$. It is common that the agents cannot achieve optimality at the end of the learning period. Agents have to strike a balance between exploration and exploitation for regret minimization [115]. Decentralization further introduces several challenges to the problem, and among them are the non-stationarity issue and the curse of dimensionality. The non-stationarity issue refers to the phenomenon that the "effective environment" each agent is facing is varying over time due to *concurrent* *learning* of the policies in the RL setting. Using the concept of the CI approach, the main solution proposed in the literature is that of "centralized learning and distributed execution" [37]. However, how to implement this scheme in practice remains open because of the high complexity involved, which goes to the next challenge: the complexity of the histories grows exponentially in $N$ and $T$, and the complexity of the prescriptions grows exponentially in $N$ and *doubly exponentially* in $T$. Algorithms proposed in the literature usually involves heuristic/approximation methods without knowing the potential loss from the approximation [35, 42, 73, 111], or some ML schemes without theoretical background [133]; some of them even assume the knowledge of the model, and are more similar to the control setting. In summary, the topic is still largely open, and general frameworks that provide theoretically sound practical design guidance are needed.

Besides the non-stationarity issue, which is still an open problem, there is a lack of a unifying analytical framework for decentralized control/MARL problems. The CI approach is a conceptual breakthrough for constructing such a framework, as decentralization is removed from the coordinator's view. However, the private history is left uncompressed in the original CI approach, so that the curse of dimensionality is not solved fundamentally; moreover, the belief state used in the approach cannot be constructed in the RL context. The idea of approximate information state (AIS) [112] provides a way for the agents to construct the state without model knowledge, but cannot be directly applied here due to the growing private histories. On the other hand, in the sufficient private information (SPI) framework [117], private histories are compressed into private states sufficient for optimal decision making; nevertheless, its construction remains unclear. While there is some progress on the deep-MARL front [35, 42, 73, 111], they are lack of a theoretical ground as mentioned above. Also not well understood is the role of information structure. Different settings are considered for the MARL problem but they may not be easily compared; the differences are typically around: which unit designs or learns which part of the policy, which unit can access the common history, and which unit can access the private histories, etc.

In this thesis, we combine the ideas of the CI approach and AIS to propose a general state representation framework for decentralized control/MARL problems (Chapter IV). In essence, we will posit a simpler MARL problem that approximately matches the original problem such that low regret control algorithms can be developed. In the framework, the common history and the private histories are compressed into the common state and the private states in time-invariant spaces for decision making, which alleviates the dimensionality issue. Besides the coordinator's viewpoint, we propose another "supervisor's perspective," which is very useful in both the conceptualization of the simplified MARL problem and also the analysis. The agents can learn the representations from the supervisor's view, then learn the optimal policies from the coordinator's view, so that the non-stationary behavior will not appear. Another approach to deal with the non-stationarity issue is to take advantage of the hierarchical information structure, which is studied in Chapter V, where we propose algorithms that achieve near-optimal regret in multi-agent bandit and MDP settings.

### 1.2.3 Scope of the Thesis

We define the scope of the thesis from many aspects in an MAS.

**Agents relation.** Throughout the thesis, we assume there is a unique utility/objective common for but usually only partially known to all agents. Most literature on distributed optimization and federated learning assume this, with some of them also taking the fairness issue [49] into account (by adding that to the global objective). The literature on decentralized control and MARL, on the other hand, considers a variety of agents relations, ranging from fully competitive, e.g. zero-sum games [78], to fully cooperative. We consider the fully cooperative cases in this thesis, which is also known as the "team problem" in the setting.

**Interaction among the agents.** The interaction among the agents is commonly modeled by graphs. There are two widely used interaction graphs (directed or undirected): one is the communication graph, and the other is the influence graph [64]. An agent is only allowed to communicate with another agent if there is an edge between the two agents in the communication graph. In an influence graph, an agent's state or decision will influence neighboring agents. Notable special cases of the graph topology include the complete (fully-connected) graph and the star graph. In the distributed optimization setting, we consider undirected communication graphs given by the model, while the concept of influence graph is unclear in our setting and will mostly not be used. In the special case where each agent is associated with a part of decision variable, the influence graph will coincide with what we call the "local network," see Example 2.7. In the decentralized control/MARL setting, we consider no additional communication among the agents unless specified through "communication actions" in the information structure of the model, and we consider potentially fully-connected influence graph given by the model.

**Decision making structure.** We mostly consider synchronous simultaneous decision making in this thesis; in each time step, this includes deciding the local copies of the decision variable in the distributed optimization setting, deciding the actions in the decentralized control setting, and deciding the actions plus updating the policies in the MARL setting. The only exception is in Chapter V, we consider a tree-like decision making structure, where the upper layer agent decides first, and then the lower layer agent decides based on the decision of the upper layer agent.

## 1.3 Organization and Contributions of the Thesis

### 1.3.1 Overview of Chapter II

The first topic we study in this chapter is localization. In real applications, individual objective functions ($f_i$'s) usually do not depend on the whole decision variable $\mathbf{x}$, so that a *partial dependency structure* exists; for example, in a base station (BS) network with the throughput of a BS being its objective function and power transmitted or scheduling being the decision variable, it is usually the case that the throughput only depends on power or scheduling variables from the nearby BSs, which is only part of the entire decision variable tuple. We achieve memory storage and communication efficiency by proposing a localization framework. In the framework, an agent only maintains the variable part its objective depends on, and only communicates with a neighbor the variable part that both of the agents depend on. This framework can work with most of the primal-based algorithms, and we demonstrate its convergence when combined with the NEXT algorithm [80] and the DGD algorithm [91].

With the localization framework, we also demonstrate that the choice of a local network, that is, a subgraph in the network that dictates the decision of a certain variable part, can be a decision variable. The corresponding optimization target would be a combination of memory storage cost, communication cost, and convergence rate. Evidently, including more agents in a local network increases the memory and communication required, but also implies more nodes are helping with relaying the information and faster consensus, which may or may not lead to a faster convergence rate. The choice of local networks affects the convergence rate through the spectral radius of the topology of the graphs, which possesses some interesting properties that we observe.

The second topic develops methods to expand the applicability of some distributed optimization algorithms by relaxing the assumption that $\nabla f_i$'s are Lipschitz continuous. Such assumption limits the usage of the algorithms, as there are common objective functions in practice that have non-Lipschitz gradients, e.g. the throughput function (often characterized by Shannon entropy) in telecommunication networks, or the cross-entropy function in ML. The relaxation is carried out by using a series of functions that *have* Lipschitz gradients to slowly approach the original function, and is demonstrated with the NEXT algorithm (as a version of the DGD algorithm actually does not require the Lipschitz gradient assumption). We provide examples of such approximation functions and show in simulations that such approximations scheme leads to numerically more stable schemes.

At the end, we also have a case study on the application to the resource allocation problem in BS networks, where we give different schemes to apply the NEXT algorithm and our proposed frameworks and simulate their performances.

### 1.3.2 Overview of Chapter III

In this chapter, we study the nonlinear consensus scheme for distributed optimization algorithms. As mentioned in Section 1.2.1, most primal-based methods contain a local optimization step and a consensus step. From a stochastic approximation perspective, these methods interleave the two steps by running the consensus step at the fast time-scale and local optimization at the slow time-scale. The consensus step can be thought of as a projection onto the consensus plane, not unlike the projection onto the constraint set in constrained optimization. We explain this point of view by providing an alternative proof of the convergence of the NEXT algorithm.

We generalize the linear consensus scheme commonly seen in many primal-based algorithms and in particular the NEXT algorithm and the DGD algorithm in two directions. First, one may relax the column stochasticity constraint for NEXT so that the agents may select any point within the shrunk convex hull spanned by the new iterates from all the neighboring agents. Second, when taking the consensus, one may first take a monotonic Lipschitz transformation and transform back after the averaging; the result further enlarges the candidate choices to the cube hull in the consensus step. We then propose an algorithm that aligns the consensus step with the gradient descent direction in combination with the gradient tracking scheme we described in Section 1.2.1. We observe convergence speed-ups in the simulation for certain cases.

### 1.3.3 Overview of Chapter IV

This chapter combines the idea of an approximate information state (AIS) [112] and sufficient private information (SPI) [117], and develops a general framework of approximate sufficient common state (ASCS) and approximate sufficient private state (ASPS) that can be used in solving decentralized stochastic control problems as well as designing practical algorithms in the MARL setting. In brief, in a POMDP setting, [112] gives a set of conditions an information state, a compression of the history, should satisfy for decision purposes without losing optimality, and a corresponding set of conditions an AIS should satisfy based on which the decisions are made, leading to a regret bound depending on the approximate error parameters and the number of time steps remained. In the Dec-POMDP setting, [117] provide another set of conditions an SPI (which we call the sufficient private state (SPS) to distinguish), a compression of the private history, should satisfy so that one can restrict attention to SPI-based prescriptions without losing optimality. Note that the compression in [112] is state compression, while the compression in [117] is *action compression*, as the space of prescriptions, the action space in the CI approach, being shrunk to the space of SPI-based prescriptions.

We begin by reproving the result in [117] using a dynamic programming (DP) induction approach. In the Dec-POMDP setting, we also provide the conditions for a general compression of the public state, which we call the sufficient common state (SCS). With SCSs as states and SPS-based prescriptions as actions, we show that the

DP is equivalent to the DP without the compressions. Then, paralleling the results of [112], we provide conditions of an ASCS and an ASPS, and derive an optimality gap for the resulting DP that still depends on the error parameters and the number of time steps remained. Due to the involvement of action compression, the dependence on the number is quadratic, in contrast to the linear dependence found in [112]. At the end, we propose a change detection (CD) framework that can adapt to new environments in a time-varying MARL setting, and discuss possible combinations with the aforementioned ASCS and ASPS framework.

### 1.3.4 Overview of Chapter V

We explore the MARL problem in a special case, i.e. multi-agent problems with a particular hierarchical information structure; in the structure, the decisions are made sequentially, and any decision maker has all the decision information from all decision makers prior to itself in the sequence but not the decision makers that act after it. We first consider the two-player bandit setting, which is the simplest setting where things are time-invariant. By adding a bonus term related to the regret bound of the lower layer (more informed) agent, we propose a hierarchical algorithm based on Upper Confidence Bound (UCB) that achieves near-optimal regret; in the algorithm, with the greater bonus, the upper layer (less informed) agent explores the arms slower, giving the lower layer agent enough time to find out the optimal arm in the lower layer for any given upper layer arm first. Although one can achieve nearly the same near-optimal regret when applying the CI approach to this bandit problem, this requires the upper layer agent to recover all the computations done in the lower layer including possible randomization, which may be infeasible in practice because the upper layer agent does not have the computation power, or it does not know the random seed, etc. Our algorithm on the other hand exempts the upper layer agent from these additional complications by exploiting the hierarchical information structure.

We extend the algorithm to bandit settings with multiple layers and multiple agents in the same layer, and to Markov decision process (MDP) settings, with the hierarchical information structure. Finally, we provide the case study of solving the "matching problem," which is a Dec-POMDP example first proposed and discussed in [86]. We demonstrate the following solution paradigm in this case study. Suppose in an MARL problem, we know certain structures about its Dec-POMDP model. Then, we can first solve the partially known model using the CI approach to obtain structural results, and use RL methods based on the structural results to achieve good regrets when it comes to the execution phase. In this particular matching problem, we show that with the structural results we obtain from it, the problem is equivalent to a bandit problem with the hierarchical information structure, and can be solved by the algorithm we mentioned in the last paragraph. The crucial characteristic of the problem is the agents can always resolve the state after one step. In general, if the state is resolvable by all agents within a finite number of time steps, then the MARL problem can be solved by bandit-type algorithms.

### 1.3.5 Summary of Contributions

In summary, the major contributions of this thesis are as follows:

- By exploiting the partial dependency structure, we propose a localization scheme that achieves memory storage and communication efficiency. We also show that the localization scheme affects the convergence rate through the spectral radii of the local networks, which turns out to be another degree of freedom one may optimize.

- We enhance the applicability of the NEXT algorithm by relaxing the gradient Lipschitz assumption commonly made in the literature by using a series of approximation functions. We provide many examples of how to construct such function series, including using the Lasry-Lions envelope function [69]. Our proposed scheme exhibits increased numerical stability in simulations.

- We generalize the linear consensus scheme to a nonlinear consensus scheme from a stochastic approximation point of view, so that the agents may choose any point in the cube hull spanned from the results of the local optimization step in the neighbors for possible convergence speed-up.

- We propose a general SCS and SPS framework for Dec-POMDP settings. This framework compresses the common history and private history to time-invariant spaces, so as to escape the curse of dimensionality in Dec-POMDP/MARL problems. We show that the DP based on our framework is equivalent to the original DP using the CI approach without any compression. For its approximate counterpart, we derive a bound on the regret incurred from using the approximate compression framework in DP.

- We design algorithms for MARL problems with hierarchical information structure in bandit settings and in an MDP setting. We show that our proposed algorithms for all the settings achieve near-optimal regret in terms of $A$, $B$, and $T$, where $A$ and $B$ are number of actions of the two agents, and $T$ is the number of episodes. We also investigate their application to a decentralized matching problem with two agents.

## 1.4 Notation

Notation in each chapter is self-contained. Appendices use the same notation as their corresponding chapter, e.g. Appendix A uses the same notation as Chapter 2. There are some common notations and conventions that are used throughout the thesis, which are listed in the following.

- $\mathbb{I}\{\cdot\}$ denotes the indicator function, which is 1 when the proposition inside is true and 0 otherwise.
- $\mathbb{E}[X]$ denotes the expectation over the random variable $X$.
- $\Omega(X)$ denotes the domain space a (random) variable $X$ takes value in.
- For $a \leq b$ and $c \leq d$, $X_{a:b}^{c:d}$ is the short hand notation for the tuple

  $$X_{a:b}^{c:d} \triangleq (X_a^c, X_a^{c+1}, \dots, X_a^d, \dots, X_b^c, \dots, X_b^d);$$

  in the most common usage, the subscript is used as the time index, and the superscript is used as the agent index. Similar notation also works for functions

  $$f_{a:b}^{c:d}(\cdot) \triangleq \left(f_a^c(\cdot), \dots, f_a^d(\cdot), \dots, f_b^c(\cdot), \dots, f_b^d(\cdot)\right).$$

- When the context is clear that there are $N$ agents and superscript is used as agent index, then

  $$X^{-n} \triangleq (X^1, \dots, X^{n-1}, X^{n+1}, \dots, X^N);$$

  similar notation works for functions.

- In a context that involves randomness, random variables are denoted by upper case letters, e.g. $X \in \Omega(X)$, and their realizations are denoted by corresponding lower case letter $x \in \Omega(X)$.

- $\Delta(\mathcal{X})$ denotes the set of all possible distributions on the space $\mathcal{X}$.

- For random variables $X$ with a realization $x$, we use the shorthand notations $\mathbb{P}(\cdot|x) \triangleq \mathbb{P}(\cdot|X = x)$ and $\mathbb{E}[\cdot|x] \triangleq \mathbb{E}[\cdot|X = x]$.

- If a random variable appears alone without realization in a place other than the operand of $\mathbb{E}$, then it means the related equation should hold for any Borel measurable subset in its domain. For example, $\mathbb{P}(X, a|b) = \mathbb{P}(c|X, d) = \mathbb{E}[Y|X, f]$ means for any Borel measurable subset $B \in \Omega(X)$, we have

  $$\mathbb{P}(X \in B, a|b) = \mathbb{P}(c|X \in B, d) = \mathbb{E}[Y|X \in B, f].$$

# CHAPTER II

# Localization and Approximations for Distributed Optimization

## 2.1 Introduction

In this chapter, we study a localization scheme and a objective function approximations scheme for distributed optimization. These two schemes are used to alleviate two issues we find in the NEXT algorithm [80], and also in the distributed optimization literature in general. Among the first provably convergent algorithms for *non-convex* objectives using a fully distributed scheme over a network with arbitrary graphical structure, the NEXT algorithm performs local optimization by finding surrogate convex functions based on the current iterate and utilizing successive convex approximations of the non-convex objective, and then enforces consensus among the agents in the network so that a global objective can be solved in a distributed manner.

The first issue of NEXT is that the algorithm requires each node to store and update the entire vector of decision variables, irrespective of the underlying dimension or structure; this is also an issue more broadly for most distributed optimization algorithms, e.g. the DGD algorithm [91]. In certain applications, the decision variables might be the ensemble of sets of control parameters at each node, which could be of a significant dimension: e.g., multiple platoons of automated cars with a local controller for each team, or cellular base stations (BSs) each with many connected devices. Directly using NEXT would necessitate greatly increased storage at each node and also high-rate and low-latency communication between all nodes, which is impractical for a large network. In the illustrative examples above, the decision variables typically can be decomposed into blocks with a sparse interconnection between different blocks through the objective functions, which we call a *partial dependency structure*. Such a structure could be used in reducing the storage and communication requirements. This, however, has not received much attention in the distributed optimization literature. The block coordinate descent method for centralized optimization is studied in [12], wherein gradient descent is effectively carried out one block at a time. When the objective is the sum of separable functions, convergence is shown for extensions of ADMM when the number of blocks is two [12], but this no longer holds when there are three blocks [32]. A distributed optimization scheme for variables with block structure is proposed in [40], but only the convex part of the objective is decomposable.

A second issue with the approach in [80] is that the objective in many applications may not have the required smoothness. For example, the common assumption is of Lipschitz and bounded gradients as in [80], but this may not hold; as mentioned in Section 1.3.1, both the throughput function in telecommunication networks and the cross entropy function in ML, which commonly serve as objective functions in the applications, have non-Lipschitz gradients. In centralized convex optimization, apart from sub-gradient methods, proximal methods are used for non-smooth functions [100]: e.g., a substitute is the Moreau envelope that is strongly convex and maintains the minimizer.

For the first issue, we exploit the separability of the decision variables and the objectives' partial dependencies to reduce the storage and communication needs when the objective is the sum of separable functions. Although all components of the decision variable are entangled via each node's objective, we show that each component can be maintained and optimized within a local network. Our idea of localization is similar to [130], which exploits the sparsity of the constraints in convex feasibility problems (CFPs) to reduce the memory and communication needed. As we will explain in Section 2.4.3, not only is our framework more general, but it also works for non-convex problems when combined with NEXT. We demonstrate the localization scheme in combined with NEXT and DGD. For the latter case, we further assume convexity to analyze how the introduction of the localization scheme affects the convergence rate, and find it alters the "effective dimension" and "effective spectral radius." This leads to our study of the choice of the local networks, which represents a trade-off between the memory storage and communication required, and the speed of convergence. In the second direction, inspired by the proximal method, we use a series of functions with Lipschitz gradients to approximate the original objective, and significantly relax the smoothness assumptions made in [80]. We show that using the approximations scheme, convergence to stationary points is still guaranteed for *any sequence of approximation functions* that approaches the original function slowly enough when there are no unbounded gradients on the boundary. While following the same steps as NEXT when the gradients are Lipschitz, our algorithm appears to have increased numerical stability over NEXT with non-Lipschitz gradients. Finally, we apply the results and algorithms to the multi-cell resource allocation problem in many different ways, which gives numerous algorithms with provable convergence to locally optimal solutions.

The rest of the chapter is organized as follows. We first describe the setup for distributed optimization problem, as well as the localization and function approximations frameworks in Section 2.2, then we present our proposed algorithms with the convergence theorems in Section 2.3. We discuss implementation details of our algorithms in Section 2.4, including examples of localization, choosing sequence of approximation functions, and the aforementioned numerical stability issue. In Section 2.5, we illustrate the choice of local networks works as a degree-of-freedom to "optimize the optimization algorithms," and discuss the effect of graph topology on its optimal spectral radius. We describe the application to the multi-cell resource allocation problem with the simulation results in Section 2.6, and conclude in Section 2.7.

## 2.2 Models and Assumptions

In this section, we describe the standard distributed optimization setup in Section 2.2.1, and the assumptions regarding the surrogate function used for NEXT in Section 2.2.2; further specifications of the model for our localization scheme and approximation scheme are given in Section 2.2.3 and in Section 2.2.4, respectively.

### 2.2.1 Distributed Optimization Setup

In this subsection, we give the system model of distributed non-convex optimization and assumptions. The bulk of the setup mainly follows from [80]. Consider a network $\mathcal{N} = \{1, \ldots, n\}$ that consists of $n$ nodes. We aim to solve an optimization problem of the form

$$\min_{\mathbf{x} \in \mathcal{K}} \quad U(\mathbf{x}) = F(\mathbf{x}) + G(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}) + G(\mathbf{x}), \tag{2.1}$$

where all $f_i$'s are $C^1$ smooth but can be non-convex, and $G$ is convex but may be non-smooth. The goal is to let these nodes cooperatively solve the problem in a distributed fashion. Therefore, each $j \in \mathcal{N}$ maintains a copy of the entire decision variable $\mathbf{x}$, referred to as $\mathbf{x}_j$. Then (2.1) is equivalent to solving the optimization problem

$$\min_{\mathbf{x}_j \in \mathcal{K}} \quad \sum_{i=1}^{n} f_i(\mathbf{x}_j) + G(\mathbf{x}_j) \tag{2.2}$$

at each $j \in \mathcal{N}$ subject to the constraint that all nodes agree on their optimal choices, i.e., we enforce

$$\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n. \tag{2.3}$$

In the context of distributed optimization, node $i$ only has the information of $f_i$. It would require communication between the nodes to solve the problem in (2.2)-(2.3).

Below are the standard assumptions on the objective functions and the constraint set.

**Assumption A**

**(A1)** The set $\mathcal{K} \in \mathbb{R}^d$ is closed and convex;

**(A2)** $G$ is convex with bounded subgradient $L_G$ for all $\mathbf{x} \in \mathcal{K}$;

**(A3)** $U$ is coercive, that is, $\lim_{\mathbf{x} \in \mathcal{K}, |\mathbf{x}| \to \infty} U(\mathbf{x}) = \infty$; based on this we can effectively assume that $\mathcal{K}$ is compact;

**(A4)** $f_i$'s have bounded gradients, i.e. $\exists\ B$ s.t. $\|\nabla f_i(\mathbf{x})\| \leq B$ for all $\mathbf{x} \in \mathcal{K}$;

**(A5)** $f_i$'s have Lipschitz gradients, i.e. $\exists\ L_i$ s.t. $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L_i \|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$.

The set of nodes $\mathcal{N}$ along with a set of undirected edges $\mathcal{E}$ form a graph $\mathcal{G} =$

$(\mathcal{N}, \mathcal{E})$. This graph captures how communications take place – node $i$ and $j$ can only communicate if $(i,j) \in \mathcal{E}$. The communication graph $\mathcal{G}$ is assumed to be connected and simple to foster communication between the nodes; otherwise, the problem is generally unsolvable[1]. Moreover, the graph $\mathcal{G}$ is associated with a symmetric doubly stochastic matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ such that $\|\mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\|_2 < 1$, and that $W_{ij} > 0$ only if $(i,j) \in \mathcal{E}$ with the exception that $W_{ii} > 0$ is allowed even if we assume there is no self-loop.

### 2.2.2 Surrogate Functions

In the NEXT algorithm, families of convex surrogate functions of the original objective functions are used instead in each iterate for faster convergence. We assume $\tilde{f}_i(\bullet; \mathbf{x})$, the surrogate function of $f_i$ at iterate $\mathbf{x}$, satisfies the following assumptions:

**Assumption F**

**(F1)** $\tilde{f}_i(\bullet; \mathbf{x})$ is convex;

**(F2)** $\nabla \tilde{f}_i(\mathbf{x}; \mathbf{x}) = \nabla f_i(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{K}$;

**(F3)** Either $\tilde{f}_i(\bullet; \mathbf{x})$'s are coercive for all $\mathbf{x} \in \mathcal{K}$ and $i \in \mathcal{N}$ or $G(\bullet)$ is coercive.

### 2.2.3 Localization Setting and Assumptions

Consider the setup where there are $M$ local dependency sets $\mathcal{N}_1, \ldots, \mathcal{N}_M$, and the local objective function of node $i$, i.e. $f_i$, only depends on a common variable $\mathbf{x}^c$ and the local variables $\mathbf{x}^m$ whenever $i \in \mathcal{N}_m$. To be more specific, the decision variables can be split into $M+1$ parts $\mathbf{x} \triangleq (\mathbf{x}^1, \ldots, \mathbf{x}^M, \mathbf{x}^c)$ where $M$ is an arbitrary positive integer. For all $m \in [M]$ (which stands for $1, \ldots, M$), define the local dependency set $\mathcal{N}_m \triangleq \{i : f_i \text{ is a function of } \mathbf{x}^m\} \subseteq \mathcal{N}$. Denote the sizes of $\mathcal{N}_1, \ldots, \mathcal{N}_M$ as $n_1, \ldots, n_M$. We can think $\mathcal{N}$ itself as the $(M+1)$-th local dependency set $\mathcal{N}_{M+1}$ and every $f_i$ may depend on $\mathbf{x}^c \triangleq \mathbf{x}^{M+1}$. We adopt the convention that whenever $M+1$ appears in either superscript or subscript, it means $c$ or anything associated with the original network $\mathcal{G}$; this includes $n_{M+1} \triangleq |\mathcal{N}| = n$. In the other direction, define the dependent part $\mathcal{S}_i \triangleq \{m : f_i \text{ is a function of } \mathbf{x}^m, m \in [M+1]\}$. Note that for all $i$, $\mathcal{S}_i$ contains $M+1$. Also, when $\mathcal{S}_i$ appears in the superscript of a variable, for example $\mathbf{x}$,

---

[1] All results can be trivially extended to directed time-varying graphs: see [80] for NEXT and the Appendix for our algorithm. For easy of presentation we adopt the above settings.

it means the vector concatenated from all $\mathbf{x}^k$'s such that $k \in \mathcal{S}_i$, i.e. $\mathbf{x}^{\mathcal{S}_i} \triangleq (\mathbf{x}^k)_{k \in \mathcal{S}_i}$. Concrete examples of local dependency sets are provided in 2.4.1. Furthermore, we have the following assumptions:

**Assumption L**

**(L1)** Besides the fact that $f_i$ depends on $\mathbf{x}^m$ only if $i \in \mathcal{N}_m$ for $m \in [M+1]$, $G$ also only depends on $\mathbf{x}^c$;

**(L2)** The set $\mathcal{K}$ is *separable*, i.e. it is the direct product of $(M+1)$ convex sets in proper subspaces $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_{M+1} \subset \mathbb{R}^d$ such that $\mathbf{x}^m \in \mathcal{K}_m \subset \mathbb{R}^{d_m}$ for all $m \in [M+1]$ if and only if $\mathbf{x} \triangleq (\mathbf{x}^1, \ldots, \mathbf{x}^{M+1}) \in \mathcal{K}$;

**(L3)** The local network $\mathcal{G}_m = (\mathcal{N}_m, \mathcal{E}_m = \{(i,j) \in \mathcal{E} : i \in \mathcal{N}_m \text{ and } j \in \mathcal{N}_m\}$ is connected for all $m \in [M]$[2], we already assume the connectedness of $\mathcal{G}_{M+1}$ in Section 2.2.1;

**(L4)** For all $m \in [M+1]$, there is a matrix $\mathbf{W}^m \in \mathbb{R}^{n_m \times n_m}$ associated with the local network $\mathcal{N}_m$ – each entry is non-zero if and only if there is a corresponding edge in $\mathcal{E}_m$ with the exception that $W_{ii}^m > 0$ for $i \in \mathcal{E}_m$ is allowed just as before. Equivalently, for all rows $i \notin \mathcal{N}_m$ and all columns $j \notin \mathcal{N}_m$, we have $(\mathbf{W}^m)_{i,:} = 0$ and $(\mathbf{W}^m)_{:,j} = 0$. In addition, denote $\mathbf{W}'^m \in \mathbb{R}^{n_m \times n_m}$ as the matrix obtained from deleting the zero rows and columns of $\mathbf{W}^m$; then $\mathbf{W}'^m$ is doubly-stochastic and $\|\mathbf{W}'^m - \frac{1}{n_m}\mathbf{1}_{n_m}\mathbf{1}_{n_m}^\top\|_2 < 1$ where $n_m \triangleq |\mathcal{N}_m|$[3]. $\mathbf{W}^{M+1}$ does not contain zero row or column, and corresponds to the $\mathbf{W}$ defined in Section 2.2.1.

Note that $\mathcal{N}_m$'s do not have to be disjoint and form a partition of $\mathcal{N}$. Having $\mathcal{N}_m \cap \mathcal{N}_l \neq \emptyset$ for some $m \neq l$ is allowed. Without loss of generality we can assume they form a covering of $\mathcal{N}$; i.e., $\bigcup_{m=1}^{M+1} \mathcal{N}_m = \mathcal{N}$. When there exists a part $\mathbf{x}^c$ on which all $f_i$'s and $G$ depend, then $\mathcal{N}_{M+1}$ is $\mathcal{N}$ itself; otherwise, nodes outside the union do not have any cross-dependence with all the rest and can be optimized themselves.

### 2.2.4 Proximal Approximations Setting and Assumptions

In contrast to the common setting in the optimization literature, we consider a scenario where $\nabla f_i$ is not Lipschitz continuous for some $i$. Our idea to relax this

---

[2] We add nodes to get connectedness if it does not hold. More on this in 2.4.2

[3] Consider $n = 4$ with a local network $\mathcal{G}_1 = (\{3,4\}, \{(3,4)\})$. Then $\mathbf{W}'^1_{eq:A.14}$ should be understood as the weight for the edge $(3,4)$ in $\mathcal{G}_1$, even if $\mathbf{W}'^1 \in \mathbb{R}^{2\times 2}$.

Lipschitz assumption is to use a sequence of functions whose gradients *are* Lipschitz continuous to approximate $f_i$. This is commonly known as the proximal approximation method in the literature of convex optimization, except that our objective might be non-convex.

To be more specific, we want to find a series of functions $\{f^*_{i,t}\}_{t\geq 1}$ such that $\nabla f^*_{i,t}$ is globally Lipschitz continuous with constant $L_{i,t}$ and that as $t \to \infty$ we have $f^*_{i,t} \to f_i$ pointwise, or even better – uniformly. Then at iteration $t$ we can use the well-behaved $f^*_{i,t}$ instead of $f_i$. We will see that as long as the schedule of $\{L_{i,t}\}_{t\geq 1}$ satisfies certain conditions, we can still have convergence to optimality.

The following assumptions are the key features of $f^*_{i,t}$ that our algorithm needs for convergence to optimality.

**Assumption N**:

**(N1)** $\nabla f^*_{i,t}$ is globally Lipschitz continuous with constant $L_{i,t}$;

**(N2)** $\lim_{t\to\infty} f^*_{i,t} \to f_i$ uniformly, and $\lim_{t\to\infty} \nabla f^*_{i,t} \to \nabla f_i$ pointwise.

We will also need a surrogate function of $f^*_{i,t}$, denoted as $\tilde{f}^*_{i,t}$. These surrogate functions should satisfy Assumption F′ similar to Assumption F given as follows.

**Assumption F′**

**(F1′)** $\tilde{f}^*_{i,t}(\bullet; \mathbf{x})$ is uniformly strongly convex with constant $\tau_{i,t} > 0$;

**(F2′)** $\nabla \tilde{f}^*_{i,t}(\mathbf{x}; \mathbf{x}) = \nabla f^*_{i,t}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{K}$;

**(F4′)** $\nabla \tilde{f}^*_{i,t}(\mathbf{x}; \bullet)$ is uniformly Lipschitz continuous with constant $L_{i,t}$.

As [80] does not have approximation functions, they assume Assumption F′ for the surrogate function of $f_i$, i.e. $\tilde{f}_i$, with fixed $\tau_i > 0$ and $L_i < \infty$. Here our assumptions are more general than NEXT in that we do not require the strong convexity of $\tilde{f}_i(\bullet; \mathbf{x})$ and the Lipschitz continuity of $\nabla \tilde{f}_i(\mathbf{x}; \bullet)$. In particular, we can have $\lim_{t\to\infty} \tau_{i,t} = 0$ and $\lim_{t\to\infty} L_{i,t} = \infty$ as long as the schedules of $\{\tau_{i,t}\}_t$ and $\{L_{i,t}\}_t$ satisfies certain conditions. We do have an additional Assumption F3. However, that assumption is implicitly implied if $\tilde{f}_i(\bullet; \mathbf{x})$'s are strongly convex; thus, we do not lose any generality from [80]. Also for simplicity, we assume $L_{i,t}$ is a non-decreasing sequence. We denote $F^*_t = \sum_{i=1}^n f^*_{i,t}$.

## 2.3 Algorithms

We are now in a position to present our proposed algorithms that alleviate the communication overhead issue and non-Lipschitz gradient issue we described. The first algorithm, Localized Proximal Inexact NEXT given in Section 2.3.1, is based on NEXT [80] and allows non-convex objective functions. In Section 2.3.2 where we present the Localized Distributed Gradient Descent which is based on Distributed Gradient Descent (DGD) [91], however, convexity of objective functions is assumed; along with the smooth assumption, the gradients of the objective functions are effectively Lipschitz, so that the proposed Localized DGD algorithm only deals with the communication overhead issue.

### 2.3.1 Localized proximal Inexact NEXT

We start with introducing the NEXT algorithm for completeness. In the NEXT algorithm, each node performs a local convex optimization, and then some information will be exchanged in the network. At a high level, just as many primal distributed optimization algorithms, the first step is the "descent step" and the second is the "consensus step;" the two steps are iteratively applied to obtain the solution [80]. In the first step of time $n$, the node $i$ solves a convex approximation of the whole objective function by convexizing its own objective function $f_i$ *parametrized by* the current iterate $\mathbf{x}_i[t]$ to be a strongly convex surrogate function $\tilde{f}_i(\bullet; \mathbf{x}_i[t])$, while linearizing the sum of other nodes' objective functions $\sum_{j \neq i} f_j$. In particular, node $i$ solves the following local convex optimization problem at time $n$

$$\tilde{\mathbf{x}}_i(\mathbf{x}_i[t], \tilde{\pi}_i[t]) = \underset{\mathbf{x}_i \in \mathcal{K}}{\arg\min} \tilde{f}_i(\mathbf{x}_i; \mathbf{x}_i[t]) + \tilde{\pi}_i[t]^\top (\mathbf{x}_i - \mathbf{x}_i[t]) + G(\mathbf{x}_i) \tag{2.4}$$

where $\tilde{\pi}_i[t]$ is a variable maintained at node $i$ to approximate $\pi_i[t] \triangleq \sum_{j \neq i} \nabla f_j(\mathbf{x}_i[t])$ for the linearization of others' objectives. Since node $i$ does not have this information, it uses another variable $\mathbf{y}_i[t]$ to track the average of gradients $\frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\mathbf{x}_i[t])$ with the gossip of similar variables maintained in all the nodes in the network, and approximates $\pi_i[t]$ with $\tilde{\pi}_i[t] \triangleq n \cdot \mathbf{y}_i[t] - \nabla f_i[t]$ where we denote $\nabla f_i[t] \triangleq \nabla f_i(\mathbf{x}_i[t])$. The Inexact NEXT algorithm is given in Algorithm 1. All operations containing index $i$

are performed for all $i \in \mathcal{N}$, i.e., Line 1, 5, 6, 7, 9, 10, and 11.

**Algorithm 2.1** Inexact NEXT

```
1  Initialization: $\mathbf{x}_i[0] \in \mathcal{K}$, $\mathbf{y}_i[0] = \nabla f_i[0]$, $\tilde{\pi}_i[0] = n\mathbf{y}_i[0] - \nabla f_i[0]$, $t = 0$.
2  while $\mathbf{x}[t]$ does not satisfy the termination criterion do
3      $t \leftarrow t + 1$
4      Local SCA optimization: for all $i \in \mathcal{N}$
5      $\tilde{\mathbf{x}}_i[t] = \tilde{\mathbf{x}}_i(\mathbf{x}_i[t], \tilde{\pi}_i[t])$
6      Find a $\mathbf{x}_i^{inx}[t]$ s.t. $\|\tilde{\mathbf{x}}_i[t] - \mathbf{x}_i^{inx}[t]\| \le \epsilon_i[t]$
7      $\mathbf{z}_i[t] = \mathbf{x}_i[t] + \alpha[t](\mathbf{x}_i^{inx}[t] - \mathbf{x}_i[t])$
8      Consensus update: for all $i \in \mathcal{N}$
9      $\mathbf{x}_i[t+1] = \sum_{j=1}^{n} W_{ij}\mathbf{z}_j[t]$
10     $\mathbf{y}_i[t+1] = \sum_{j=1}^{n} W_{ij}\mathbf{y}_j[t] + (\nabla f_i[t+1] - \nabla f_i[t])$
11     $\tilde{\pi}_i[t+1] = n \cdot \mathbf{y}_i[t+1] - \nabla f_i[t+1]$
Output: $\mathbf{x}[t]$
```

With Assumption A and Assumption F, the result established in [80] is convergence to the stationary points (see Theorem 4 of [80]), whose definition is given as follows.

**Definition 2.1:** *A point* $\mathbf{x}^*$ *is a stationary point of Problem* (2.1) *if a subgradient* $g \in \partial G(\mathbf{x}^*)$ *exists such that* $(\nabla F(\mathbf{x}^*) + g)^\top(\mathbf{y} - \mathbf{x}^*) \ge 0$ *for all* $\mathbf{y} \in \mathcal{K}$.

Now consider the setting with Assumption A without (A5), Assumption F, Assumption F′, Assumption L, and Assumption N. Our localized proximal inexact version of NEXT, which requires less storage and communication, and allows unbounded non-Lipschitz objective gradients, is presented in Algorithm 2.2. All operations that contain index $i$, i.e. Line 1, 5, 6, 7, 9, 10, and 11, are performed for all $i \in \mathcal{N}$. Also, except Line 5, the operations have a superscript $k$, and are performed for all $k \in \mathcal{S}_i$. In Line 5,

$$\tilde{\mathbf{x}}_i^*(\mathbf{x}_i[t], \tilde{\pi}_i[t]) = \underset{\mathbf{x}_i \in \prod_{k \in \mathcal{S}_i} \mathcal{K}_k}{\arg\min} \left[ \tilde{f}_{i,t}^*(\mathbf{x}_i; \mathbf{x}_i[t]) + \sum_{k \in \mathcal{S}_i} \tilde{\pi}_i^k[t]^\top (\mathbf{x}_i^k - \mathbf{x}_i^k[t]) + G(\mathbf{x}_i^c) \right], \quad (2.5)$$

with $\mathbf{x}_i = (\mathbf{x}_i^k)_{k \in \mathcal{S}_i} = \mathbf{x}_i^{\mathcal{S}_i}$ and $\tilde{\pi}_i = (\tilde{\pi}_i^k)_{k \in \mathcal{S}_i} = \tilde{\pi}_i^{\mathcal{S}_i}$. Note that Line 6 along with Line 5 means that one solves the optimization problem in (2.5) with accuracy $\epsilon_i[t]$. Also denote $\nabla f_{i,t}^*[t] = \nabla f_{i,t}^*(\mathbf{x}_i[t])$. The output of the algorithm is the concatenation of $[\bar{\mathbf{x}}^m]_{m \in [M+1]}$, where $\bar{\mathbf{x}}^m \triangleq \frac{1}{n_m} \sum_{i \in \mathcal{N}_m} \mathbf{x}_i^m$.

**Algorithm 2.2** Localized Proximal Inexact NEXT

```
Initialization: $\mathbf{x}_i^k[0] \in \mathcal{K}_k$, $\mathbf{y}_i^k[0] = \nabla_{\mathbf{x}^k} f_{i,0}^*[0]$, $\tilde{\pi}_i^k[0] = (n_k - 1)\mathbf{y}_i^k[0]$, $t = 0$
while $\mathbf{x}[t]$ does not satisfy the termination criterion do
    $t \leftarrow t + 1$
    Local SCA optimization: for all $i \in \mathcal{N}$ and for all $k \in \mathcal{S}_i$
    $\tilde{\mathbf{x}}_i[t] = \tilde{\mathbf{x}}_i^*(\mathbf{x}_i[t], \tilde{\pi}_i[t])$
    Find a $\mathbf{x}_i^{k,inx}[t]$ s.t. $\|\tilde{\mathbf{x}}_i^k[t] - \mathbf{x}_i^{k,inx}[t]\| \leq \epsilon_i^k[t]$
    $\mathbf{z}_i^k[t] = \mathbf{x}_i^k[t] + \alpha[t](\mathbf{x}_i^{k,inx}[t] - \mathbf{x}_i^k[t])$
    Consensus update: for all $i \in \mathcal{N}$ and for all $k \in \mathcal{S}_i$
    $\mathbf{x}_i^k[t+1] = \sum_{j \in \mathcal{N}_k} W_{ij}^k \mathbf{z}_j^k[t]$
    $\mathbf{y}_i^k[t+1] = \sum_{j \in \mathcal{N}_k} W_{ij}^k \mathbf{y}_j^k[t] + (\nabla_{\mathbf{x}^k} f_{i,n+1}^*[t+1] - \nabla_{\mathbf{x}^k} f_{i,t}^*[t])$
    $\tilde{\pi}_i^k[t+1] = n_k \cdot \mathbf{y}_i^k[t+1] - \nabla_{\mathbf{x}^k} f_{i,n+1}^*[t+1]$
Output: $\mathbf{x}[t] \triangleq (\bar{\mathbf{x}}^1[t], \ldots, \bar{\mathbf{x}}^M[t], \bar{\mathbf{x}}^c[t])$
```

NEXT is a special case of Algorithm 2.2. First, NEXT would either be the case where $n_m = n$ for all $m \in [M]$, or the equivalent case where $\mathbf{x}$ consists of one part $\mathbf{x}^c$ only. In the NEXT algorithm, node $i$ keeps the whole $\mathbf{x}$, and communicates the whole $\mathbf{x}$ as well with all of its neighbors; the same also applies to the variables $\mathbf{y}$ and $\tilde{\pi}$. On the contrary, under our localization setting, node $i$ turns out only has to maintain $\mathbf{x}^{\mathcal{S}_i}$ (also $\mathbf{y}^{\mathcal{S}_i}$ and $\tilde{\pi}^{\mathcal{S}_i}$) in Algorithm 2.2; moreover, it only communicates $\mathbf{x}^{\mathcal{S}_i \cap \mathcal{S}_j}$ (also $\mathbf{y}^{\mathcal{S}_i \cap \mathcal{S}_j}$) with its neighbor $j$. This could mean substantial savings for memory and communication, especially when the dependency structure is sparse. Example 2.8 provides an example that explicitly specifies the storage and communication needed before and after localization. Secondly, NEXT would be when $f_{i,t}^* = f_i$, $L_{i,t} = L_i < \infty$, and $\tau_{i,t} = \tau_i > 0$ for all $i$ and $n$. Algorithm 2.2 also accommodates a bigger class of objective functions and supports a more flexible choice of the schedules of $\{L_{i,t}\}$ and $\{\tau_{i,t}\}$.

**Remark 2.2:** *Note that Algorithm 2.2 is not the result of the direct application of NEXT in the localization setting given in Section 2.2.3. Suppose* $m'$ *is such that* $i \notin \mathcal{N}_{m'}$ *and under the connectedness assumption. Different from NEXT, node* $i$ *no longer keeps the gradient trace* $\tilde{\pi}_i^{m'}$ *in Algorithm 2.2, nor does the variable* $\mathbf{x}_i^{m'}$ *appear in the objective of the local optimization. The fact that convergence can still be obtained with Algorithm 2.2 is not obvious from [80].*

The following theorem generalizes the convergence to stationary points result of

NEXT when gradients are bounded, specifying restrictions on the schedules of $\{L_{i,t}\}$, $\{\tau_{i,t}\}$. When gradients are unbounded, stricter constraints are needed.

**Theorem 2.3:** *For all* $m \in [M+1]$, *let* $\{\mathbf{x}^m[t]\}_t \triangleq \{(\mathbf{x}_i^m[t])_{i\in\mathcal{N}_m}\}_t$ *be the sequences generated by Algorithm 2.2,* $\{\bar{\mathbf{x}}^m[t]\}_t \triangleq \left\{\frac{1}{n_m}\sum_{i\in\mathcal{N}_m}\mathbf{x}_i^m[t]\right\}_t$ *be their averages, and* $\{\bar{\mathbf{x}}[t]\}_t = \{(\bar{\mathbf{x}}^1[t],\ldots,\bar{\mathbf{x}}^M[t],\bar{\mathbf{x}}^c[t])\}_t$ *be the ensemble of averages. Let* $L_t^{\max} = \max_i L_{i,t}$, $\tau_t^{\min} = \min_i \tau_{i,t}$, $\epsilon[t] = \min_{i,k}\epsilon_i^k[t]$, $\zeta_t = \max_{\mathbf{x}\in\mathcal{K}}\|F_t^*(\mathbf{x}) - F_{n-1}^*(\mathbf{x})|$, $\eta_{i,t} = \max_{\mathbf{x}\in\mathcal{K}}\|\nabla f_{i,t}^*(\mathbf{x}) - \nabla f_{i,n-1}^*(\mathbf{x})\|$, *and* $\eta_t^{\max} = \max_i \eta_{i,t}$.

*(a) Suppose Assumptions A excluding (A5), F, L, N, and F' hold*[4]*. Also,* $\alpha[t] \in (0,1]$ *is such that* $\sum_{t=0}^{\infty}(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2 < \infty$, $\sum_{t=0}^{\infty}\tau_t^{\min}\alpha[t] = \infty$, $\sum_{t=0}^{\infty}\alpha[t]L_t^{\max}\epsilon[t] < \infty$. *Then (1) all sequences* $\{\mathbf{x}_i^m[t]\}_t \ \forall\ m \in [M+1]$ *asymptotically agree, i.e.,* $\lim_{t\to\infty}\|\mathbf{x}_i^m[t] - \bar{\mathbf{x}}^m[t]\| = 0 \quad \forall\ i \in \mathcal{N}_m$*; (2)* $\{\bar{\mathbf{x}}[t]\}_t$ *is bounded, and its limit points are stationary points of the original problem.*

*(b) If we do not assume Assumption A4, but we have* $\lim_{t\to\infty}\alpha[t]\frac{(L_t^{\max})^5}{(\tau_t^{\min})^3} = 0$, *as well as* $\lim_{t\to\infty}\frac{\eta_t^{\max}}{\tau_t^{\min}} = 0$, $\sum_{t=0}^{\infty}\frac{\alpha[t]L_t^{\max}\eta_t^{\max}}{\tau_t^{\min}} < \infty$, $\sum_{t=0}^{\infty}\zeta_t < \infty$, *and* $\lim_{t\to\infty}\nabla F_t^* \to \nabla F$. *Then we still have results (1) and first part of (2). When the limit points lie in the interior of* $\mathcal{K}$, *or* $\nabla F$ *is bounded on those limit points, they will be stationary points.*

*(c) Continuing (b), if a limit point* $\bar{\mathbf{x}}^\infty$ *lies on the boundary of* $\mathcal{K}$ *and* $\|\nabla F(\bar{\mathbf{x}}^\infty)\| = \infty$, *then the definition of stationary point does not apply, and* $\bar{\mathbf{x}}^\infty$ *could be a local minimum, or a point that is not a local minimum.*

**Proof:** See Appendix A.2. ■

**Remark 2.4:** *Note that the conditions in (a) imply* $\lim_{t\to\infty}\alpha[t]\frac{(L_t^{\max})^3}{(\tau_t^{\min})^3} = 0$. *In (b) we apply the stricter* $\lim_{t\to\infty}\alpha[t]\frac{(L_t^{\max})^5}{(\tau_t^{\min})^3} = 0$ *as well as other constraints.*

We can see that although all parts of the variable $\mathbf{x}$ are coupled through all the objectives $f_i$'s, the decision on $\mathbf{x}^m$ is actually dictated by the nodes in $\mathcal{N}_m$. We give a class of approximation functions that satisfies Assumption N, namely the Lasry-Lions envelope or double envelope [69], in Section 2.4.4. Note that the convergence

---

[4] Note that Assumption N1 requires $\lim_{t\to\infty}L_t^{\min} = \infty$ where $L_t^{\min} = \min_i L_{i,t}$ with $f_{i,t}^*$ given as (5.22). For any other choices of $f_{i,t}^*$ other than the double Moreau envelope function [69], Assumption N1 requires $\lim_{t\to\infty}L_{i,t} = \infty$ for any $i$ such that $\nabla f_i$ is non-Lipschitz continuous.

of Theorem 2.3 only requires the Lipschitz constants of $\{\nabla f_{i,t}^*\}_{n\geq 1}$ follow certain schedules, but $\{f_{i,t}^*\}_{n\geq 1}$ can be any sequence of functions that approaches $f_i$ in the limit, and does not need to be double envelope. We will give many examples of such sequences in Section 2.4.6 and Section 2.6.4. Also, an example of the schedules of the the parameters $\{L_t^{\max}\}_t$, $\{\tau_t^{\min}\}_t$, $\{\alpha[t]\}_t$, and $\{\epsilon[t]\}_t$ that satisfies the related conditions in Theorem 2.3 (a) and (b) using $p$-series is given in Section 2.4.5. We will discuss the unbounded gradient issue in full detail in Section 2.4.6, study different examples there, and show the numerical stability of our algorithm for these examples.

### 2.3.2 Localized DGD

We will again start with describing the DGD algorithm first proposed in [91]; however, we will consider the convex objective function setup with simpler convergence analysis given in [55]. Specifically, we consider the setup given in Section 2.2.1 with a few more constraints: we let $\mathcal{K} = \mathbb{R}^d$ to save the hassle of projections and $G = 0$; also, we assume $f_i$'s are convex and denote $L \triangleq L^{\max} = \max_i L_i$ for simplicity. Just as many primal distributed optimization algorithms, in DGD node $i$ maintains a local copy of the decision variable, and updates the copy by taking the convex combination of the copies in its neighbor set and subtracting its current gradient. The DGD algorithm is presented in Algorithm 2.3.

**Algorithm 2.3** DGD

```
1 Initialization: $\mathbf{x}_i[0] \in \mathbb{R}^d$, $t = 0$.
2 while $\mathbf{x}[t]$ does not satisfy the termination criterion do
3     $t \leftarrow t + 1$
4     $\mathbf{x}_i[t+1] = \sum_j W_{ij}\mathbf{x}_j[t] - \alpha[t]\nabla f_i(\mathbf{x}_i[t])$    (2.6)
  Output: $\mathbf{x}[t]$
```

Line 4 is performed for all $i \in \mathcal{N}$, and $\alpha[t]$ is the learning rate at time $t$. The initial copy value $\mathbf{x}_i[0]$ can be chosen arbitrarily from $\mathbb{R}^d$. Note that the matrix $\mathbf{W}$ already characterizes the communication constraint, as node $i$ will not directly obtain any copy from the node other than its neighbors in (2.6).

Let $\mathbf{x}^*$ be one optimal solution of (2.1) and $F^* = F(\mathbf{x}^*)$ be the optimal value. Define the weighted running average to be $\mathbf{x}_{i,ra}[t] = (\sum_{k=0}^{t-1} \alpha_k \mathbf{x}_i[k])/(\sum_{k=0}^{t-1} \alpha_k)$. Then [55] establishes that with the choice of $\alpha[t] = \frac{1}{2L(t+1)^{1/3}}$, one can achieve the convergence rate of $F(\mathbf{x}_{i,ra}[t]) - F^* = O(1/t^{2/3})$ for all $i$.

**Theorem 2.5:** *Let* $\vartheta(\mu) \triangleq \frac{2}{1-\mu} + \sup_{t\geq 1} t\mu^{t/2}$ *where* $\mu \triangleq \|\mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\|_2 \in [0,1)$, *which is the spectral radius of* $\mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ *since we consider symmetric* $\mathbf{W}$. *Assume Assumption A with* $\mathcal{K} = \mathbb{R}^d$, $G = 0$, $L = \max_i L_i$, $\alpha[t] = \frac{1}{2L(t+1)^{1/3}}$ *and* $f_i$*'s being convex. Then the iterates of Algorithm 2.3 converge with the rate*

$$F(\mathbf{x}_{i,ra}[t]) - F^* \leq C(\mu, n, B, L, d)\frac{1}{t^{2/3}} \tag{2.7}$$

*where*

$$\begin{aligned} &C(\mu, n, B, L, d) \\ &\triangleq \left[n^2 d B^2 \vartheta^2(\mu)\|\mathbf{x}^*\|^2 + \frac{n^2 d^2 B^4 \vartheta^4(\mu)}{2L}\right]^{1/2} + \frac{n^2 d B^2 \vartheta^2(\mu)}{8L} + nL\|\mathbf{x}^*\|^2 + ndB^2\vartheta^2(\mu). \end{aligned} \tag{2.8}$$

**Proof:** The result follows by working through the details of [55], which only considers the simpler case when $d = 1$. ■

Note that as $C(\mu, n, B, L, d)$ is an increasing function of $\vartheta(\mu)$ which is in turn an increasing function of $\mu$, $C(\mu, n, B, L, d)$ is an increasing function of $\mu$; hence we want $\mu$ as small as possible.

Now consider the case where a partial dependency structure exists in the objective function, that is, assume Assumption L with $M$ parts (instead of $M+1$ parts where the last part has universal dependency). Then the DGD algorithm can be modified to the Localized DGD algorithm given in Algorithm 2.4 to reduce the memory storage and communication involved.

**Algorithm 2.4** Localized DGD

```
1 Initialization: $\mathbf{x}_i[0] \in \mathbb{R}^d$, $t = 0$.
2 while $\mathbf{x}[t]$ does not satisfy the termination criterion do
3     $t \leftarrow t + 1$
4     $\mathbf{x}_i^m[t+1] = \sum_{j\in\mathcal{N}_m} W_{ij}^m \mathbf{x}_j^m[t] - \alpha[t]\nabla_{\mathbf{x}^m} f_i(\mathbf{x}_i^{\mathcal{S}_i}[t])$        (2.9)
  Output: $\mathbf{x}[t]$
```

In Line 4, the operation is performed for all $i \in \mathcal{N}$ and $m \in \mathcal{S}_i$. To see why after localization the algorithm still converges intuitively, we define the average vector

$\bar{\mathbf{x}}^{m,l}[t] = \frac{1}{n_m}\sum_{i\in\mathcal{N}_m}\mathbf{x}_i^m[t]$, and then apply the operator $\frac{1}{n_m}\sum_{i\in\mathcal{N}_m}(\bullet)$ on (2.9) to get

$$\bar{\mathbf{x}}^{m,l}[t+1] = \bar{\mathbf{x}}^{m,l}[t] - \frac{\alpha[t]}{n_m}\sum_{i\in\mathcal{N}_m}\nabla_{\mathbf{x}^m} f_i(\mathbf{x}_i^{\mathcal{S}_i}[t]). \tag{2.10}$$

We can see that each variable part $\mathbf{x}^m$ still descends in the correct direction. In contrast, if without localization the algorithm will iterate as

$$\bar{\mathbf{x}}^{m}[t+1] = \bar{\mathbf{x}}^{m}[t] - \frac{\alpha[t]}{n}\sum_{i}\nabla_{\mathbf{x}^m} f_i(\mathbf{x}_i^{\mathcal{S}_i}[t]), \tag{2.11}$$

where $\bar{\mathbf{x}}^m[t] = \frac{1}{n}\sum_{i\in\mathcal{N}_m}\mathbf{x}_i^m[t]$, which only differs from (2.10) by a constant factor. Also, the part $\mathbf{x}_i^m[t]$ approaches $\bar{\mathbf{x}}^{m,l}[t]$ via $\mathbf{W}^m$ with localization, while it approaches $\bar{\mathbf{x}}^m[t]$ via $\mathbf{W}$ without localization.

In sum, the localization scheme essentially differs in the way of taking averages from its counterpart. It changes the "effective $\mathbf{W}$" used for averaging and in turn $\vartheta(\mu)$ in the constant $C(\mu, n, B, L, d)$. Thus, it is intuitive that adopting localization affects the convergence rate by a constant factor, which is given in the next theorem.

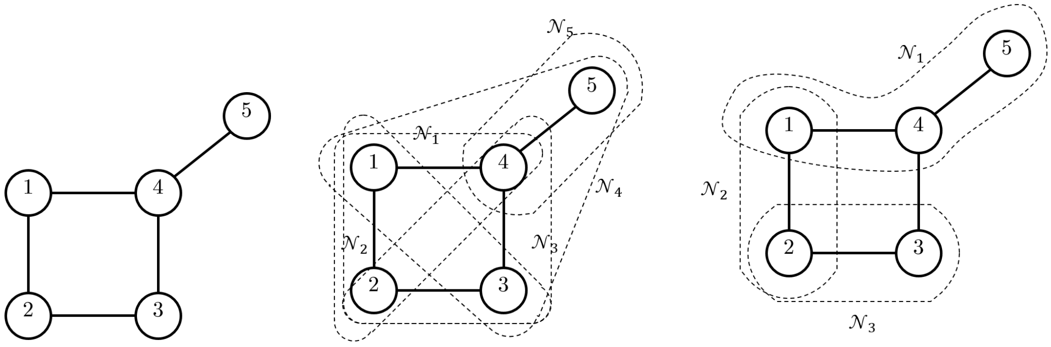
**Theorem 2.6:** *Let* $\mu_m \triangleq \|\mathbf{W}'^m - \frac{1}{n_m}\mathbf{1}_{n_m}\mathbf{1}_{n_m}^\top\|_2 \in [0,1)$ *be the spectral radius of* $\mathbf{W}'^m - \frac{1}{n_m}\mathbf{1}_{n_m}\mathbf{1}_{n_m}^\top$ *for all* $m \in [M]$, $\mu' \triangleq \max\{\mu_1, \ldots, \mu_M\}$, *and* $d' = \frac{\sum_{m=1}^M n_m d_m}{n}$. *In addition to the assumptions made in Theorem 2.5, we further assume Assumption L (with M parts). Then the iterates of Algorithm 2.4 converge with the rate*

$$F(\mathbf{x}_{i,ra}[t]) - F^* \le C(\mu', n, B, L, d')\frac{1}{t^{2/3}}. \tag{2.12}$$

**Proof:** See Appendix A.4. ■

## 2.4 Discussions

In this section, we discuss some details of our proposed algorithms. Two localization examples that compare the effect of localization are given in Section 2.4.1. We comment on how to exploit localization when Assumption L3 is violated in Section 2.4.2. Comparison to the localization algorithm proposed by Hu *et al.* in [130] is given in Section 2.4.3. Section 2.4.4 and Section 2.4.5 summarize the property of



(a) The communication graph.
(b) Local dependency sets for Example 2.7.
(c) Local dependency sets for Example 2.8

Figure 2.1: Graphical illustrations of the local dependency set examples.

double envelope that we will use and provide the $p$-series example for the parameters $(L, \tau, \alpha, \epsilon)$, respectively. We explain how our approximation scheme can deal with objective functions with unbounded gradients in Section 2.4.6 through examples, where we provide concrete series of approximation functions $\{f^*_{i,t}\}_{t\geq 1}$ that satisfy the conditions in Theorem 2.3 and demonstrate better numerical stability in the simulations for the examples.

### 2.4.1 Examples of Localization

In this subsection we give a few examples to illustrate the concept and effectiveness of localization.

**Example 2.7:** *Consider the communication graph given in Figure 2.1 (a). In this example we consider the most common situation, where the dependency is directly given by the communication graph. Specifically, every node* $i$ *has a corresponding variable part* $\mathbf{x}^i$ *in the whole variable tuple* $\mathbf{x}$*. Moreover, the objective at node* $i$ *only depends on its own variable* $\mathbf{x}^i$ *and its neighbors' variables* $\{\mathbf{x}^j : j \in N(i)\}$*, where* $N(i)$ *is the set of neighboring nodes of node* $i$*. In other words, the local dependency set* $\mathcal{N}_i$ *corresponding to* $\mathbf{x}^i$ *is equal to* $Nb(i) := N(i) \cup \{i\}$*.* $G$ *is assumed to be* $0$*. This situation also arises in the resource allocation application we discuss in Section 2.6.*

*For this communication graph, this situation will be the following. The variable* $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5)$ *can be split into five parts, and we have* $f_1 = f_1(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^4)$ *(*$f_1$ *only depends on* $\mathbf{x}^1$*,* $\mathbf{x}^2$*, and* $\mathbf{x}^4$*),* $f_2 = f_2(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$*,* $f_3 = f_3(\mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4)$*,* $f_4 =$

*$f_4(\mathbf{x}^1, \mathbf{x}^3, \mathbf{x}^4, \mathbf{x}^5)$, and $f_5 = f_5(\mathbf{x}^4, \mathbf{x}^5)$. The local dependency sets for this example are given in Figure 2.1 (b). Before localization, node 1 needs to store all of $\mathbf{x}^1$ to $\mathbf{x}^5$ and communicates this information to its neighbors $\{2, 4\}$. In contrast, after localization node 1 only keeps $\mathbf{x}^1$, $\mathbf{x}^2$, and $\mathbf{x}^4$, and exchanges some of this information with $\{2, 4\}$. In addition, node 1 does not maintain the information of $\mathbf{y}^3$, $\mathbf{y}^5$, or $\tilde{\pi}^3$, $\tilde{\pi}^5$, either. Before localization, NEXT performs*

$$\arg\min_{\mathbf{x}_5 \in \mathcal{K}} \tilde{f}_5(\mathbf{x}_5; \mathbf{x}_5[t]) + \tilde{\pi}_5[t]^\top (\mathbf{x}_5 - \mathbf{x}_5[t])$$

*in the local optimization step at node 5, while after localization Algorithm 2.2 performs the following*

$$\arg\min_{(\mathbf{x}_5^4, \mathbf{x}_5^5) \in \mathcal{K}_4 \times \mathcal{K}_5} \tilde{f}_5(\mathbf{x}_5^4, \mathbf{x}_5^5; \mathbf{x}_5^4[t], \mathbf{x}_5^5[t]) + \tilde{\pi}_5^4[t]^\top (\mathbf{x}_5^4 - \mathbf{x}_5^4[t]) + \tilde{\pi}_5^5[t]^\top (\mathbf{x}_5^5 - \mathbf{x}_5^5[t]).$$

*The first terms of the two operations are actually the same. Indeed, since $f_5$ only depends on $\mathbf{x}^4$ and $\mathbf{x}^5$, we can definitely choose a surrogate function that only depends on the local storage of $\mathbf{x}^4$ and $\mathbf{x}^5$. The difference is node 5 does not have to store, say $\tilde{\pi}_5^1$, and optimize $\mathbf{x}_5^1$ based on $\tilde{\pi}_5^1$. The variable $\tilde{\pi}_5^1$ is asymptotically tracking $\sum_{j \neq 5} \nabla_{\mathbf{x}^1} f_j$. Our localization result says that since node 5 does not have any preference on $\mathbf{x}^1$, it only follows others' decision of $\mathbf{x}^1$ through $\tilde{\pi}_5^1$. As for DGD, the local optimization and consensus step are performed in a single equation. The local optimization step for DGD can be seen as the gradient descent operation $-\alpha[t]\nabla_{\mathbf{x}^4, \mathbf{x}^5} f_5(\mathbf{x}_5^4[t], \mathbf{x}_5^5[t])$ is exactly the same with and without localization. Before localization, node 5 still maintains $\mathbf{x}^1$, $\mathbf{x}^2$, and $\mathbf{x}^4$; however, it only updates the variable parts through gossip since $f_5$ does not depend on the parts. Hence, it is unnecessary for node 5 to maintain $\mathbf{x}_5^1$ and $\tilde{\pi}_5^1$ (in the case of NEXT) – it can just take other nodes' decision at the end, even though $\mathbf{x}^1$ and $\mathbf{x}^5$ are coupled through, say $f_4$. This saves nodes from unnecessary memory storage and communication in the presence of sparse dependency structure, which is crucial when the network is large.*

**Example 2.8:** *We consider the same communication network but a different dependency structure: $f_1 = f_1(\mathbf{x}^1, \mathbf{x}^2)$, $f_2 = f_2(\mathbf{x}^2, \mathbf{x}^3)$, $f_3 = f_3(\mathbf{x}^3)$, $f_4 = f_4(\mathbf{x}^1)$, and $f_5 = f_5(\mathbf{x}^1)$. There are three local dependency sets in this example as shown in Fig-*

*ure 2.1 (c). The variable parts that are stored at each node and the communication required to other node for each part are given in Table 2.1*[5]*. Naturally, before localization every node keeps all parts and communicates them with all of its neighbors. On the contrary, the storage and communication required are greatly reduced after localization. Notice that even though node 3 is directly linked with node 4, they do not communicate as there is no common part they both depend on.*

Table 2.1: Storage and communication required in Example 2.8.

| Node | Before localization | After localization |
|---|---|---|
| 1 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$, $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$, $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 2, 4 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$ to 4, $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$ to 2 |
| 2 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$, $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$, $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 1, 3 | $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$ to 1, $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 3 |
| 3 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$, $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$, $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 2, 4 | $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 2 |
| 4 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$, $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$, $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 1, 3, 5 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$ to 1, 5 |
| 5 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$, $\mathbf{x}^2$, $\mathbf{y}^2$, $\tilde{\pi}^2$, $\mathbf{x}^3$, $\mathbf{y}^3$, $\tilde{\pi}^3$ to 4 | $\mathbf{x}^1$, $\mathbf{y}^1$, $\tilde{\pi}^1$ to 4 |

### 2.4.2 Relaxing Assumption L3

Our requirement of connectedness of $\mathcal{G}_m$ is not a strong assumption. If any $\mathcal{G}_m$ is not connected, we can always "add" nodes as "relays" into the local dependency set. For example, consider Example 2.8 with $f_3$ revised to be a constant and $f_4$ revised as $f_4 = f_4(\mathbf{x}^1, \mathbf{x}^3)$. Then $\mathcal{G}_3 = (\{2, 4\}, \phi)$ is not connected. We can add node 3 into $\mathcal{G}_3$ by making $f_3 = f_3(\mathbf{x}^3)$ where the dependency is actually trivial. Node 3 can thus relay the information of $\mathbf{x}^3$ for node 2 and 4. Alternatively, node 1 can also do the relay job. Hence, we assume without loss of generality that the nodes have been added by some algorithm that might depends on network structure and communication requirements so that every $\mathcal{G}_m$ is connected. As we will see in Section 2.5, the flexibility of choosing local networks allows us for further optimization of the algorithms in the sense of reducing memory storage and communication required or accelerating the convergence.

---

[5] The table is for NEXT algorithms; for DGD algorithms, the result is similar but excluding any $\mathbf{y}$ and $\tilde{\pi}$ variables.

### 2.4.3 Comparison to Hu et al. [130]

In [130], a similar idea of localization for convex feasibility problems (CFPs) is proposed, where they also exploit the sparsity of the constraints to reduce the storage and communication required. Our framework is different from theirs in two aspects. First, in their framework each node $i$ owns a part of the variable tuple $\mathbf{x}^i$, whose corresponding "dependency network graph" $(\mathcal{N}_i, \{(i,j) : j \in \mathcal{N}_i\})$ must be a subgraph of $i$'s local graph $(N(i), \{(j,k) \in \mathcal{E} : j \in N(i) \text{ and } k \in N(i)\})$ where $N(i)$ is $i$'s neighbors in the communication graph $\mathcal{G}$. On the other hand, in our framework each part of the variable tuple $\mathbf{x}^m$ does not have specific relation with the nodes and the local dependency set $\mathcal{N}_m$ is only required to be a connected component in $\mathcal{G}$, which is more general in the sense that their framework is a sub-case of ours. Second, their dependency is built in the constraint sets, while ours is based on objective function's dependency. This difference arises from the nature of CFPs and optimization problems. Namely, we focus on solving the optimal solution for optimization problems while they aim to find the solution lying in the intersection of a batch of sets. Although one would be able to solve convex optimization with CFP algorithms [22], it is still unclear whether this could be generalized to the case of non-convex optimization.

### 2.4.4 An Example of Approximation Functions Satisfying Assumption N

The Lasry-Lions envelope or double envelope [69][107] is a class of approximation functions that serves this purpose. We use this to illustrate the feasibility of our approach but also point out that any such sequence of functions satisfying the assumptions and conditions can be used instead.

**Definition 2.9:** *The double envelope, or Lasry-Lions envelope [69][107], of a function $f$ is defined by*

$$f_{t,s}(\mathbf{x}) = \sup_{\mathbf{z}} \inf_{\mathbf{y}} \left\{ f(\mathbf{y}) + \frac{1}{2t}\|\mathbf{z}-\mathbf{y}\|^2 - \frac{1}{2s}\|\mathbf{x}-\mathbf{z}\|^2 \right\}, \tag{2.13}$$

*where $0 < s < t < \infty$.*

**Fact 2.10:** *If $f$ is lower bounded, then $\nabla f_{t,s}(\cdot)$ is Lipschitz continuous with constant* $\max\left\{\frac{1}{s}, \frac{1}{t-s}\right\}$.

**Fact 2.11:** *$f_{t,s} \to f$ pointwise as $s,t \to 0$. If further we have $f$ being uniformly continuous, then $f_{t,s} \to f$ uniformly as $s,t \to 0$. Furthermore, $\nabla f_{t,s} \to \nabla f$ pointwise as $s,t \to 0$.*

**Proof:** See [4]. ■

Now it is clear that if we define

$$f^*_{i,t}(\mathbf{x}) = \sup_{\mathbf{z}} \inf_{\mathbf{y}} \left\{ f_i(\mathbf{y}) + \frac{L_{i,t}}{4}\|\mathbf{z}-\mathbf{y}\|^2 - \frac{L_{i,t}}{2}\|\mathbf{x}-\mathbf{z}\|^2 \right\}, \tag{2.14}$$

then we have $\nabla f^*_{i,t}$ being globally Lipschitz continuous with constant $L_{i,t}$. Since $U$ is coercive (Assumption A3), we can restrict our attention to some compact set in $\mathcal{K}$, where $f_i$ is uniformly continuous. Then over this set, we will have $\lim_{t\to\infty} f^*_{i,t} \to f_i$ uniformly as well.

### 2.4.5 An Example of Sequences Satisfying the Conditions Using $p$-series

Examples of the tuples $(\alpha[t], \epsilon[t], L_t^{\max}, L_t^{\min}, \tau_t^{\min})$ satisfying the conditions of Theorem 2.3 exist with all schedules in $p$-series form. Assume $\alpha[t] = \alpha_0 t^{-\beta}$, $\epsilon[t] = \epsilon_0 t^{-\gamma}$, $L_t^{\max} = L_t^{\min} = L_0 t^{\lambda}$, and $\tau_t^{\min} = \tau_0 t^{-\delta}$ for some positive constants $\alpha_0$, $\epsilon_0$, $L_0$, and $\tau_0$. Then the constraints on the parameters are

$$\begin{cases} \lim_{t\to\infty} \alpha[t]\frac{(L_t^{\max})^5}{(\tau_t^{\min})^3} = 0 & \Rightarrow \quad \beta - 5\lambda - 3\delta > 0, \\ \sum_{t=0}^{\infty} (L_t^{\max})^3 \left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2 < \infty & \Rightarrow \quad 2\beta - 3\lambda - 2\delta > 1, \\ \sum_{t=0}^{\infty} \tau_t^{\min}\alpha[t] = \infty & \Rightarrow \quad \beta + \delta \le 1, \\ \lim_{t\to\infty} L_t^{\min} = \infty & \Rightarrow \quad \lambda > 0, \\ \sum_{t=0}^{\infty} \alpha[t] L_t^{\max}\epsilon[t] < \infty & \Rightarrow \quad \beta + \gamma - \lambda > 1. \end{cases} \tag{2.15}$$

A possible tuple $(\beta, \gamma, \lambda, \delta)$ satisfying the above equations is $(0.9, 0.2, 0.05, 0.1)$. If the $\nabla f_i$'s are Lipschitz continuous and then constant $\tau_i > 0$ for all $i$, then $\lambda = \delta = 0$ and the above requirements degenerate to $0.5 < \beta \le 1$ and $\gamma > 1 - \beta$ as in [80].

### 2.4.6 Examples of Objectives with Unbounded Gradients

When it comes to non-Lipschitz gradients, a first example might be functions with unbounded gradients, but this need not be the only case; for example, $x\sqrt{x}$ on $[0,1]$. Its derivative is $\frac{3\sqrt{x}}{2}$, which is obviously bounded on $[0,1]$ but actually not Lipschitz continuous on $[0,1]$. Though convergence is not established for this case in [80], NEXT still works well numerically in these kind of simple examples. We first turn to more challenging examples with unbounded gradients, and give the numerical results of the examples.

**Example 2.12** (Interior local optimum)**:** *Consider* $\mathcal{N} = \{1,2,3\}$ *in a triangle network, with* $f_1(x) = x^2 - (\ln 2)x$, $f_2(x) = x\ln\left(\frac{8}{x^2}\right)$, *and* $f_3(x) = -x\ln\left(\frac{4}{x^2}\right)$, *where* $x \in \mathcal{K} = [-1,2]$. *There is no* $G$. *The unique stationary point is* $x^* = 0$ *as* $F(x) = x^2$. *The derivatives for the node objectives are* $f_1'(x) = 2x - (\ln 2)$, $f_2'(x) = \ln\left(\frac{8}{x^2}\right) - 2$, *and* $f_3'(x) = -\ln\left(\frac{4}{x^2}\right) + 2$. *Obviously the derivatives are unbounded at* $x = 0$ *for* $f_2'(x)$ *and* $f_3'(x)$, *and this example is thus not covered in the theory developed in [80].*

*On the other hand, we can simply choose the following approximation functions:* $f_{2,t}^*(x) = x\ln\left(\frac{8}{x^2+1/p(t)}\right)$ *and* $f_{3,t}^*(x) = -x\ln\left(\frac{4}{x^2+1/p(t)}\right)$ *with derivatives being* ${f_{2,t}^*}'(x) = \ln\left(\frac{8}{x^2+1/p(t)}\right) - \frac{2x^2}{x^2+1/p(t)}$ *and* ${f_{3,t}^*}'(x) = -\ln\left(\frac{4}{x^2+1/p(t)}\right) + \frac{2x^2}{x^2+1/p(t)}$. *We can choose* $f_{1,n}^*(x) = f_1(x)$ *throughout. One may check that* $L_{1,t} = O(1)$ *and* $L_{2,t} = L_{3,t} = O(p(t)^{1/2})$. *The conditions in Theorem 2.3 (b) are checked in the following.*

- $(\alpha[t], L_t^{\max}, \tau_t^{\min})$ *series conditions: suppose we choose* $\alpha[t] = \alpha_0 t^{-0.7}$ *and* $\tau_t^{\min} = \tau > 0$, *since* $L_t^{\max} = O(p(t)^{1/2})$, *if we further choose* $p(t) = t^{0.2}$ *then it is evident that all conditions are satisfied.*

- $\sum_{t=0}^{\infty} \zeta_t < \infty$: *we actually have* $F_t^*(x) = F(x)$ *for all* $x$ *and hence* $\zeta_t = 0\ \forall\ n$.

- $\lim_{t\to\infty} \frac{\eta_t^{\max}}{\tau_t^{\min}} = 0$: *the maximum differences of derivatives for node 2 and 3 always occur at* $x = 0$, *and* $\nabla f_{2,t}^*(0) = \nabla f_{3,t}^*(0) = O(\log p(t)) = O(\log t)$ *for the choice* $p(t) = t^{0.2}$. *The condition holds because* $\log t - \log(t-1) = O(1/t) \to 0$.

- $\sum_{t=0}^{\infty} \frac{\alpha[t] L_t^{\max} \eta_t^{\max}}{\tau_t^{\min}} < \infty$: *notice that for our choices of* $\alpha[t] = \alpha_0 t^{-0.7}$ *and* $p(t) = t^{0.2}$ *the summed term equals* $O(t^{-0.7+0.2/2-1})$ *and thus is summable.*

- $\lim_{t\to\infty} \nabla F_t^* \to \nabla F$: *notice* $\nabla F_t^*(x) = \nabla F(x)$ *for all* $x$.

*As we have* $\nabla F$ *finite in* $\mathcal{K}$*, by Theorem 2.3 it is guaranteed that our algorithm converges to the unique stationary point* $x = 0$*, which is also a global minimum in this example. Note that this minimum lies in the interior of* $\mathcal{K}$*.*

**Example 2.13** (Boundary local optimum)**:** *Consider a one node network, and the objective function is* $f(x, y) = \sqrt{1-(x^2+y^2)} + \frac{x}{2}$ *on the region inside the unit circle* $\mathcal{K} = \{(x, y) : x^2 + y^2 \leq 1\}$*, so that the graph of* $f(x, y)$*, namely* $(x, y, f(x, y))$*, is the upper half of the unit sphere lifted in the direction of positive* $x$*-axis. For the upper half of the unit sphere, the set of global minima is the unit circle; now that we tilt the sphere, the unique global minimum is* $(-1, 0)$*. There is no stationary point inside the unit circle. Since the gradient*

$$\nabla f(x, y) = \begin{bmatrix} \frac{1}{2} - \frac{x}{\sqrt{1-(x^2+y^2)}} \\ -\frac{y}{\sqrt{1-(x^2+y^2)}} \end{bmatrix}$$

*is unbounded on the unit circle, the definition of stationary point fails there. This example again clearly lies outside the theory of [80].*

*We consider the approximation function* $F_t^*(x, y) = \sqrt{1-(x^2+y^2)+1/p(t)} + \frac{x}{2}$*. Its gradient can also be obtained by changing the 1's in the denominators of the original gradient to* $1 + 1/p(t)$*. We have* $L_t = O(t^{3/2})$*.*

- $(\alpha[t], L_t^{\max}, \tau_t^{\min})$ *series conditions: choose* $\alpha[t] = \alpha_0 t^{-0.8}$ *and* $\tau_t^{\min} = \tau > 0$*, and* $p(t) = t^{0.1}$*.*

- $\sum_{t=0}^{\infty} \zeta_t < \infty$*: notice that* $F_t^* \downarrow F$ *monotonically. Since the largest decreasing of* $F_t^* - F_{t-1}^*$ *always happen on the unit circle,* $\sum_{t=0}^{\infty} \zeta_t < \infty$ *is simply* $F_1^* - F$ *evaluated on the unit circle, which is* $\sqrt{2}$*.*

- $\lim_{t\to\infty} \frac{\eta_t^{\max}}{\tau_t^{\min}} = 0$*: roughly* $\|\nabla F_t^*\| = O(\sqrt{p(t)}) = O(t^{0.05})$ *for the choice of* $p(t) = t^{0.1}$*. The condition is true because* $t^{0.05} - (t-1)^{0.05} = O(t^{-0.95}) \to 0$*.*

- $\sum_{t=0}^{\infty} \frac{\alpha[t] L_t^{\max} \eta_t^{\max}}{\tau_t^{\min}} < \infty$*: the term is decaying in the rate of* $O(t^{-0.8+0.15-0.95})$*, which is summable.*

- $\lim_{t\to\infty} \nabla F_t^* \to \nabla F$: *this one is obvious as we only have one node.*

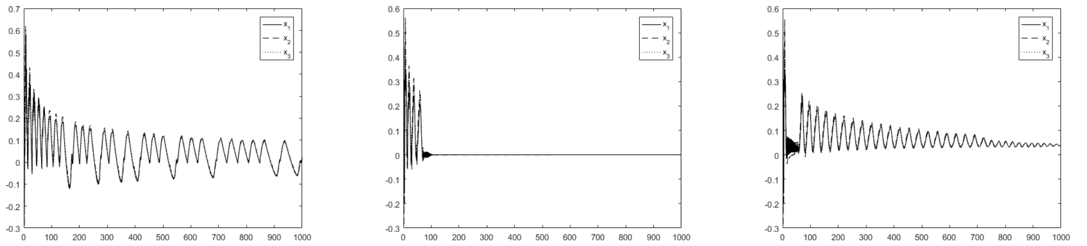
*Our method will not converge to any point in int($\mathcal{K}$); otherwise, by part (b) of Theorem 2.3 we know it must be a stationary point, but there is no stationary point inside the unit circle. Thus, our method will converge to some point in bd($\mathcal{K}$); since $\nabla F$ is infinite on the unit circle, this falls into the case of Theorem 2.3 (c), and the point is not guaranteed to be a local minimum. Even so, we find in the simulation results shown below that for a wide range of initialization of $x$, our algorithm can converge to $(-1, 0)$ while NEXT does not. Unfortunately, since the objective is symmetric with respect to $x$-axis and the unique global maximum is $(\frac{1}{\sqrt{5}}, 0)$, if we start with any point to the right of the maximum on the $x$-axis, the process will inevitably approach $(1, 0)$ in the limit.*

Now we compare the performance of our proposed Algorithm 2.2 with NEXT for the above two examples. We will see that NEXT fails numerically in the above examples while our method works much better.

Figure 2.2 depicts how the local decision variables in Example 2.12 change for NEXT and Algorithm 2.2 within 500 iterations. We start with initial value of $(x_1[0], x_2[0], x_3[0]) = (-1, -0.5, -0.25)$. We choose the following parameters: $\tau = 0.1$ (independent of $i$ and $t$), $\alpha[t] = 0.98t^{-0.7}$, $p(t) = 10t^{0.2}$,

$$\mathbf{W} = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{bmatrix},$$

and the surrogate functions are chosen to be direct linearization plus the quadratic regularization term (see (2.27) for an example of such kind of choices). In Figure 2.2 (a) we can see that NEXT oscillates and is not numerically stable for this example. Whenever the iterates go near the global minimum at $x = 0$, they jump to values far away. This is because the gradients $\nabla f_2$ and $\nabla f_3$ are infinite at the point, even though $\nabla F$ is zero there. The trackings of $(x_1, x_2, x_3)$ to $\bar{x}$ and $(y_1, y_2, y_3)$ to $\bar{y}$ are in the fast time-scale (see ??) and actually happen quite fast, as can be seen in the figure. However, a slight mismatch between $(x_1, x_2, x_3)$ and $(y_1, y_2, y_3)$ is sufficient to cause very large $\{\tilde{\pi}_i[t] + \nabla f_i[t]\}_i$, which is supposed to be very small when $\bar{x} \approx 0$,



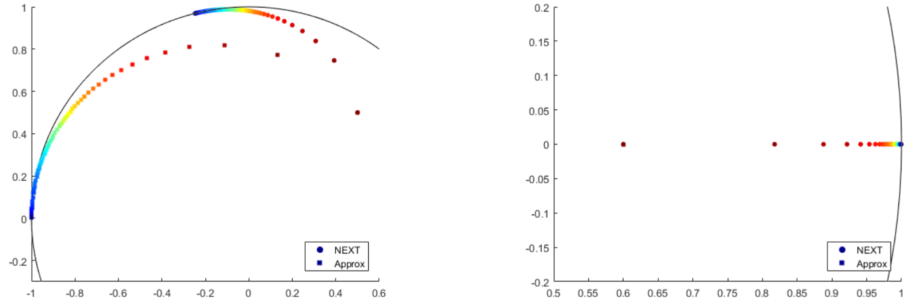
(a) NEXT (b) Algorithm 2.2 (c) Algorithm 2.2 with wrong $f^*_{2,t}$

Figure 2.2: Comparison between NEXT and Algorithm 2.2 for the local variables versus iterations in Example 2.12.

driving $x_i$'s to the boundary. If two of them jump to 2 and one of them to $-1$, then in the next few iterations they jump up; if two jump to $-1$ and one to 2, they jump down.

We cannot ensure that NEXT is not converging when $t \to \infty$ theoretically, but we observe it is still oscillating when $t$ is as large as $10^4$. In contrast, Algorithm 2.2 essentially converges to the global minimum $x = 0$ within 150 iterations in Figure 2.2 (b). In Figure 2.2 (c), we show the case when $f^*_{2,t}(x) = x \ln\left(\frac{8}{x^2+1.1/p(t)}\right)$, where we have everything satisfied except $\lim_{t\to\infty} \nabla F^*_t \to \nabla F$ – one can check there will be an additional $\ln(1/1.1)$ term. From the figure we see that not only it exhibits oscillating behavior, but it seems to converge to a wrong point other than $x = 0$.

The converging behaviors of NEXT and Algorithm 2.2 for Example 2.13 are shown in Figure 2.3 for two different initializations. The parameters are $\tau = 0.05$, $\alpha[t] = 0.98t^{-0.85}$, and $p(t) = 10t^{0.1}$. The surrogate function is again direct linearization plus quadratic regularizer. The 2-D iterates for both algorithms are plotted from red to blue, with NEXT being circle dots and Algorithm 2.2 being square dots. In Figure 2.3 (a), we start with $(0.5, 0.5)$, and both algorithms are executed for 5000 iterations (the dots are down sampled though). We see that while our algorithm is converging to the global minimum at $\mathbf{x} = (-1, 0)$, NEXT is "converging" to some point $(-0.2499, 0.9683)$ near the boundary. Due to the unbounded gradient near the boundary, the $\frac{1}{2}$ term in $\partial_x f(x, y)$ is completely dominated by the rest $(-x, -y)/\sqrt{1-(x^2+y^2)}$, which directs the iterate only to descend in the radial direction. Again we cannot ensure NEXT does not converge to $(-1, 0)$ if we run it forever; however, it does not visit the region $x < -0.25$ after $10^5$ iterations. By



(a) $\mathbf{x}[0] = (0.5, 0.5)$ (b) $\mathbf{x}[0] = (0.6, 0)$

Figure 2.3: Comparison between NEXT and Algorithm 2.2 for the evolution of the local variable in Example 2.13.

slowly changing the objective, we are able to escape this dominance and obtain the correct solution.

In (b) we start from $\mathbf{x} = (0.6, 0)$, which is to the right of the global maximum $(\frac{1}{\sqrt{5}}, 0)$, for 100 iterations, and both algorithms are converging to $(1, 0)$. Since we start on the $x$-axis and the gradients always direct to $(1, 0)$, without any perturbation there is no way to escape $x$-axis for all gradient-descent-like methods. Example 2.13 falls into the case (c) in Theorem 2.3, where the algorithm is converging to some point in the boundary and $\nabla F$ is not bounded. The definition of stationary point does not apply, and NEXT fails to converge to global minimum for both $\mathbf{x}[0] = (0.5, 0.5)$ and $\mathbf{x}[0] = (0.6, 0)$, while Algorithm 2.2 succeeds for $\mathbf{x}[0] = (0.5, 0.5)$ but also fails for $\mathbf{x}[0] = (0.6, 0)$.

## 2.5 Choice of Local Networks

As we mentioned in Section 2.4.2, choosing local networks is flexible to some extent as independent nodes can be freely added to a network to relay information due to trivial dependence. The choice of local networks affects not only memory storage and communication required, but also the convergence rate through the consensus mixing time mainly decided by the spectral radius of the chosen network. In Section 2.5.1 we introduce the notion of optimizing such choice with the specification of the constraints the local networks need to satisfy. We discuss how the spectral radius of a network might depend on its graph topology in Section 2.5.2. Then we perform numerical simulation to demonstrate the benefit of localization scheme in a convex objectives setting with DGD algorithm in Section 2.5.3.

### 2.5.1 Optimizing Local Networks

In our localized algorithms (Algorithm 2.2 and Algorithm 2.4), one of the conditions required for convergence is (L3), stating the local networks $\mathcal{G}_m = (\mathcal{N}_m, \mathcal{E}_m = \{(i,j) \in \mathcal{E} : i \in \mathcal{N}_m \text{ and } j \in \mathcal{N}_m\}$ for all $m \in [M]$ must be connected, where $\mathcal{N}_m$'s are directly defined by the objective dependency. Obviously, without localization, one uses $\mathcal{G}$ as the local networks for all the parts, which also leads to convergent algorithms. Suppose we associate the local networks $\mathcal{G}'_m = (\mathcal{N}'_m, \mathcal{E}'_m)$, $m \in [M]$, each corresponds to the variable part $\mathbf{x}^m$, with the doubly stochastic matrices $\mathbf{W}^{m\prime}$, $m \in [M]$, and then perform the localized algorithms with $\mathbf{W}^{m\prime}$. It is straightforward that only the following constraints need to be satisfied to make the algorithms adhere to the communication constraints and converge to the correct optimal value:

(a) $\mathcal{N}_m \subset \mathcal{N}'_m$ so that the whole local dependency set is covered in the new local network;

(b) $\mathcal{E}'_m \subset \{(i,j) \in \mathcal{E} : i \in \mathcal{N}'_m \text{ and } j \in \mathcal{N}'_m\} := \mathcal{E}(\mathcal{N}'_m)$ so that the algorithm only utilizes nodes inside the local network and existing edges;

(c) the local network $\mathcal{G}'_m = (\mathcal{N}'_m, \mathcal{E}'_m)$ is connected.

The above constraints should hold for all $m \in [M]$.

Throughout this subsection we will adopt the notation that $\mathcal{G}'_m = (\mathcal{N}'_m, \mathcal{E}'_m)$'s are the local networks used in the algorithm while $\mathcal{N}_m$'s are the local dependency sets defined from objectives' dependencies. The choice of local networks affects the convergence rate, the memory storage, and the communication involved in the algorithms. For the memory cost, let us assume one node storing one dimension of the variable costs one unit. Then, since only the nodes in $\mathcal{N}'_m$, a total number of $n'_m$ nodes, need to store the part $\mathbf{x}^m$, this leads to a memory cost of $\sum_{m=1}^{M} n'_m d_m$. For the communication cost, we assume it takes one unit cost for one node to transmit one dimension of the variable to one of its neighbors. Thus, the amount of communication required in each iteration is $\sum_{m=1}^{M} 2e'_m d_m$, where $e'_m \triangleq |\mathcal{E}'_m|$ and the 2 factor comes from the bidirectional communication in an edge. Finally, one could consider the negative convergence rate constant $-\sqrt{\sum_{m=1}^{M} n'_m d_m}\vartheta(\max\{{\mu^*_{[M]}}'\})$ or the number of iterations required to reach $\epsilon$, i.e.,

$$t_{\min}(n'_{[M]}, {\mu^*_{[M]}}', \epsilon) \triangleq \min_t \left\{ t : \frac{C(\max\{{\mu^*_{[M]}}'\}, n, B, L, \sum_{m=1}^{M} n'_m d_m/n)}{t^{2/3}} < \epsilon \right\} \tag{2.16}$$

where $n'_{[M]} = \{n'_1, \ldots, n'_M\}$, ${\mu^*_{[M]}}' = \{\mu_1^{*\prime}, \ldots, {\mu^*_M}'\}$, and the $C$ function is defined in (2.8), as the cost for convergence. For the local network $\mathcal{G}'_m$, we mentioned that the matrix $\mathbf{W}^{m\prime}$ is associated to it for the averaging, and $\mu'_m$ is defined by $\mathbf{W}^{m\prime}$. Actually, given $\mathcal{G}'_m$, there is an optimal way to choose $\mathbf{W}^{m\prime}$ entailing the smallest $\mu'_m$, which is referred to as ${\mu^*_m}'$ here. The details will be given in Section 2.5.2. Multiplying the memory or communication consumed per round by $t_{\min}$, one could alternatively consider the total memory or communication required through the process as the cost. Assume the cost coefficients for convergence rate, communication, and memory are $c_r$, $c_c$, and $c_m$, respectively. Then an example of local network optimization problem can be formulated as

$$\begin{aligned} &\min_{\mathcal{N}'_{[M]}, \mathcal{E}'_{[M]}} c_r t_{\min} + c_c t_{\min} \sum_{m=1}^{M} 2e'_m d_m + c_m \sum_{m=1}^{M} 2n'_m d_m \\ &\text{s.t. } \mathcal{N}_m \subset \mathcal{N}'_m, \mathcal{E}'_m \subset \mathcal{E}(\mathcal{N}'_m), (\mathcal{N}'_m, \mathcal{E}'_m) \text{ connected } \forall m \in [M]. \end{aligned} \tag{2.17}$$

It is understood that $\mathcal{N}'_{[M]} = \{\mathcal{N}'_1, \ldots, \mathcal{N}'_M\}$, $\mathcal{E}'_{[M]} = \{\mathcal{E}'_1, \ldots, \mathcal{E}'_M\}$, and $t_{\min} = t_{\min}(n'_{[M]}, \mu'_{[M]}, \epsilon)$ in (2.17). Problem (2.17) is hard, and we do not have any method other than simple enumeration that can guarantee an optimal solution. Moreover, in [55] $\vartheta(\mu)$ is given as an upper bound, and we find that directly using it would *overemphasize* the importance of $\mu$. It is of interest to study a more precise expression of the cost from $\mu$ and efficient ways to solve (2.17) in future work.

### 2.5.2 Optimal Spectral Radius and Graph Topology

In this subsection, we study the relation between the optimal spectral radius of $\mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ and the graph topology $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathbf{W}$ *conforms to* $\mathcal{G}$ in the sense that is described in Section 2.2.1. The optimal spectral radius of a graph is solved by the fastest mixing Markov chain (FMMC) problem, whose detail is given below;

most of them come from [77, 23].

#### 2.5.2.1 Fatest Mixing Markov Chain Problem Setup

Consider a Markov chain on an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with transition probability matrix $\mathbf{W}$. Let $\lambda_i(\mathbf{W})$ and $\lambda_{-j}(\mathbf{W})$ denote the $i$-th largest and $j$-th smallest eigenvalues of matrix $\mathbf{W}$, respectively.

**Definition 2.14:** *The FMMC problem for the Markov chain conforming to the undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is defined as follows:*

$$\begin{aligned} &\min_{\mathbf{W}\in\mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}} \mu_{\mathcal{G}}(\mathbf{W}) = \max\{\lambda_2(\mathbf{W}), -\lambda_{-1}(\mathbf{W})\} \\ &s.t.\ \mathbf{W} \geq 0, \mathbf{W1} = \mathbf{1}, \mathbf{W} = \mathbf{W}^\top, W_{ij} = 0 \textit{ if } (i,j) \notin \mathcal{E}. \end{aligned} \tag{2.18}$$

*Denote the optimal value of this problem to be $\rho(\mathcal{G})$, which is only a function of $\mathcal{G}$.*

**Fact 2.15:** *The FMMC problem for $\mathcal{G}$ can be alternatively expressed as follows:*

$$\begin{aligned} &\min_{\mathbf{W}\in\mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}} \mu_{\mathcal{G}}(\mathbf{W}) = \left\| \mathbf{W} - \frac{1}{|\mathcal{N}|}\mathbf{1}\mathbf{1}^\top \right\|_2 \\ &s.t.\ \mathbf{W} \geq 0, \mathbf{W1} = \mathbf{1}, \mathbf{W} = \mathbf{W}^\top, W_{ij} = 0 \textit{ if } (i,j) \notin \mathcal{E}. \end{aligned} \tag{2.19}$$

Indeed, the fact follows from $\lambda_1\left(\mathbf{W} - \frac{1}{|\mathcal{N}|}\mathbf{1}\mathbf{1}^\top\right) = \max\{\lambda_2(\mathbf{W}), -\lambda_{-1}(\mathbf{W})\}$. Note that $\lambda_1(\mathbf{W}) = 1$, and by subtracting $\frac{1}{|\mathcal{N}|}\mathbf{1}\mathbf{1}^\top$ from $\mathbf{W}$ this eigenvalue is effectively shifted to 0.

**Fact 2.16:** *The FMMC problem has yet another formulation:*

$$\begin{aligned} &\min_{c\in\mathbb{R}^{|\mathcal{E}|}} {\mu'}_{\mathcal{G}}(\mathbf{c}) = \left\| \mathbf{I} - \mathbf{A}\,\mathrm{diag}(\mathbf{c})\mathbf{A}^\top - \frac{1}{|\mathcal{N}|}\mathbf{1}\mathbf{1}^\top \right\|_2 \\ &s.t.\ 1 - \sum_{u\in N(v)} c_{uv} \geq 0\ \forall\ v \in \mathcal{N}, c \geq 0, \end{aligned} \tag{2.20}$$

*where the matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{E}|}$ is the incidence matrix with $\mathbf{L} = \mathbf{A}\mathbf{A}^\top$ being the Laplacian matrix.*

This latter formulation just comes from the fact that $\mathbf{W}$ can be expressed as $\mathbf{I} - \mathbf{A}\,\mathrm{diag}(\mathbf{c})\mathbf{A}^\top$, where $c_e$ is the "weight" of the edge $e \in \mathcal{E}$, $\mathbf{A}$ is actually $\mathbf{A}_{\mathcal{E}}$, and $\mathbf{I}$

is the identity matrix of proper dimension ($|\mathcal{N}| \times |\mathcal{N}|$). With some abuse of notations write $\mathbf{W}_{\mathcal{G}}(\mathbf{c}) = \mathbf{I} - \mathbf{A}_{\mathcal{E}} \operatorname{diag}(\mathbf{c}) \mathbf{A}_{\mathcal{E}}^{\top}$. Then the objective functions of the last two formulations are the same and are related by $\mu'_{\mathcal{G}}(\mathbf{c}) = \mu_{\mathcal{G}}(\mathbf{W}_{\mathcal{G}}(\mathbf{c}))$. This formulation has the advantage that it already incorporated the double stochasticity constraint, the symmetry constraint, and the edge constraint. The constraint $1 - \sum_{u \in N(v)} c_{uv} \geq 0$ where $N(v)$ means the neighboring nodes of $v$, states that the weight for the self-loop edge of $v$ should be non-negative.

#### 2.5.2.2 Structures in Graph Relation Formed by Optimal Spectral Radius

We are now ready to discuss the dependence of $\rho(\mathcal{G})$ on $\mathcal{G}$. When two graphs possess the same number of nodes, Proposition 2.17 tells us the one with more edges has a smaller optimal spectral radius. We then compare the optimal spectral radii of two graphs whose numbers of nodes differ by one. In particular, let us consider the case where the graph with fewer nodes is referred to as the old graph $\mathcal{G}$, and the new graph $\mathcal{G}'$ is obtained from the old graph by adding one node and a few edges between this new node and the old graph. We compared the spectral radii of many real graph examples via simulation, and found the following. If the new node is linked to the old graph by only one edge, then the spectral radius necessarily (weakly) goes up, which is stated and proved in Proposition 2.20. If the new node is linked to all the nodes in the old graph, then the spectral radius necessarily (weakly) goes down, which is left as a conjecture in Conjecture 2.21. Other than these two extreme cases, the spectral radius could either go up or down.

**Proposition 2.17:** *Consider two graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{N}, \mathcal{E}')$ of the same size. Then $\mathcal{E} \supset \mathcal{E}'$ implies that $\rho(\mathcal{G}) \leq \rho(\mathcal{G}')$.*

**Proof:** Let $\mathcal{E}'' = \mathcal{E} \setminus \mathcal{E}'$, and let $\mathbf{c} = (\mathbf{c}', \mathbf{c}'') \in \mathbb{R}^{|\mathcal{E}|}$, where $\mathbf{c}' \in \mathbb{R}^{|\mathcal{E}'|}$ and $\mathbf{c}'' \in \mathbb{R}^{|\mathcal{E}''|}$. Let

$$\begin{aligned}
\mathcal{C}_{\mathcal{G}} &= \{\mathbf{c} \in \mathbb{R}^{|\mathcal{E}|} : 1 - \sum_{u \in N(v)} c_{uv} \geq 0 \ \forall \ v \in \mathcal{N}, \mathbf{c} \geq 0\}, \\
\mathcal{C}'_{\mathcal{G}} &= \{\mathbf{c} \in \mathbb{R}^{|\mathcal{E}|} : 1 - \sum_{u \in N(v)} c_{uv} \geq 0 \ \forall \ v \in \mathcal{N}, \mathbf{c}' \geq 0, \mathbf{c}'' = 0\}, \qquad (2.21) \\
\mathcal{C}_{\mathcal{G}'} &= \{\mathbf{c}' \in \mathbb{R}^{|\mathcal{E}'|} : 1 - \sum_{u \in N(v)} c'_{uv} \geq 0 \ \forall \ v \in \mathcal{N}, \mathbf{c}' \geq 0\}.
\end{aligned}$$

It is clear that $\mathcal{C}_{\mathcal{G}} \supset \mathcal{C}'_{\mathcal{G}}$. Note that the set of edges of the graph $\mathcal{E}$ (or $\mathcal{E}'$) implicitly appears in the definition in the form of neighboring function $N(\bullet)$. Then

$$\rho(\mathcal{G}) = \min_{\mathbf{c} \in \mathcal{C}_{\mathcal{G}}} \mu'(\mathbf{c}) \leq \min_{\mathbf{c} \in \mathcal{C}'_{\mathcal{G}}} \mu'(\mathbf{c}) = \min_{\mathbf{c}' \in \mathcal{C}_{\mathcal{G}'}} \mu'(\mathbf{c}') = \rho(\mathcal{G}').$$

■

**Remark 2.18:** *We allow arbitrary permutations of the node label before checking the condition $\mathcal{E} \supset \mathcal{E}'$. For example, consider $\mathcal{G} = (\{v_1, v_2, v_3\}, \{v_1v_2, v_2v_3\})$ and $\mathcal{G}' = (\{v_1, v_2, v_3\}, \{v_1v_3\})$. Then $\mathcal{E} \supset \mathcal{E}'$ still holds if we permute $v_1$ and $v_2$ in the label of $\mathcal{G}'$.*

**Lemma 2.19** (Weyl's inequality)**:** *Let $\mathbf{M} = \mathbf{H} + \mathbf{R}$, where $\mathbf{H}$ and $\mathbf{R}$ are symmetric $n \times n$ real matrices. Suppose the eigenvalues of $\mathbf{M}$, $\mathbf{H}$, and $\mathbf{R}$, are $\mu_1 \geq \cdots \geq \mu_n$, $\nu_1 \geq \cdots \geq \nu_n$, and $\rho_1 \geq \cdots \geq \rho_n$, respectively. If $j + k - n \geq i \geq r + s - 1$, then $\nu_j + \rho_k \leq \mu_i \leq \nu_r + \rho_s$, where $(i, j, k, r, s)$ are all in $[n] = \{1, \ldots, n\}$.*

**Proposition 2.20:** *Consider an $(n-1)$-node graph $\mathcal{G} = (\mathcal{N} = \{v_1, \ldots, v_{n-1}\}, \mathcal{E})$ and an $n$-node graph $\mathcal{G}' = (\{v_1, \ldots, v_{n-1}, v_n\}, \mathcal{E} \cup \{v_{n-1}v_n\})$. The later graph $\mathcal{G}'$ is built on $\mathcal{G}$ plus one additional node $v_n$ which has only one connection to the original set of nodes $\mathcal{N}$, through $v_{n-1}$ in the description. Then $\rho(\mathcal{G}) \leq \rho(\mathcal{G}')$.*

**Proof:** See Appendix A.5. ■

**Conjecture 2.21:** *Consider an $(n-1)$-node graph $\mathcal{G} = (\mathcal{N} = \{v_1, \ldots, v_{n-1}\}, \mathcal{E})$ and an $n$-node graph $\mathcal{G}' = (\{v_1, \ldots, v_{n-1}, v_n\}, \mathcal{E} \cup \{v_1v_n, \ldots, v_{n-1}v_n\})$. The later graph $\mathcal{G}'$ is built on $\mathcal{G}$ plus one additional node $v_n$ which is linked to all the nodes in the original set of nodes $\mathcal{N}$. Then $\rho(\mathcal{G}) \geq \rho(\mathcal{G}')$.*

These results establish structures in graph relation. Specifically, consider an $(n-1)$-node graph $\mathcal{G} = (\mathcal{N} = \{v_1, \ldots, v_{n-1}\}, \mathcal{E})$ and the set of $n$-node graphs that include $\mathcal{G}$ as their induced subgraph, i.e. $\Xi(\mathcal{G}) \triangleq \{\mathcal{G}' : \mathcal{G}' = (\mathcal{N}' = \{v_1, \ldots, v_{n-1}, v_n\}, \mathcal{E}') \text{ s.t. } \mathcal{E} \subset \mathcal{E}'\}$. Obviously, $(\Xi(\mathcal{G}), \subset)$ is a poset, where $\mathcal{G}'_1 = (\mathcal{N}', \mathcal{E}'_1) \subset \mathcal{G}'_2 = (\mathcal{N}', \mathcal{E}'_2)$ means there exists a permutation of the node label such that $\mathcal{E}'_1 \subset \mathcal{E}'_2$. Proposition 2.17 implies $\rho(\mathcal{G}'_1) \leq \rho(\mathcal{G}'_2)$ if $\mathcal{G}'_1 \subset \mathcal{G}'_2$. Moreover, the original graph $\mathcal{G}$ will be able to form a cut in the Hasse diagram of $(\Xi(\mathcal{G}), \subset)$; on one side of the cut are the graphs

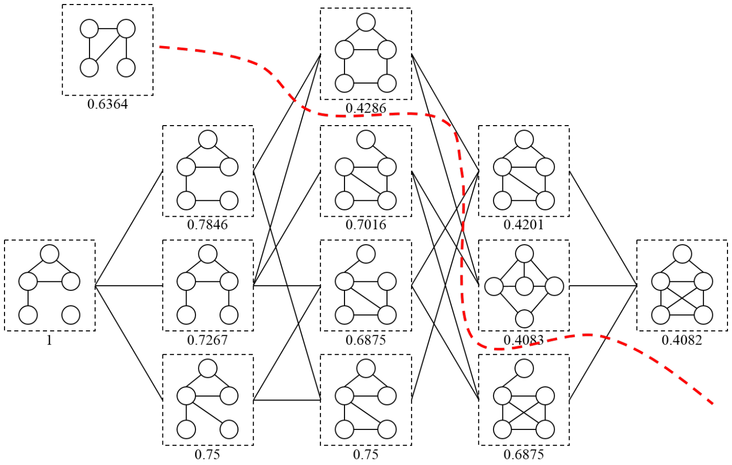


Figure 2.4: An example of graph relation established by the optimal spectral radius. $\mathcal{G}'$'s whose $\rho(\mathcal{G}')$'s are larger than $\rho(\mathcal{G})$, and on the other side are the ones less than $\rho(\mathcal{G})$ (assuming there is no equal case). Conjecture 2.21 says that the greatest element of the Hasse diagram is necessarily on the smaller side, while Proposition 2.20 means the direct parents of the least element (which is an unconnected graph $(\mathcal{N}', \mathcal{E})$ with $\rho = 1$) are necessarily on the larger side. Figure 2.4 gives an example of such structures. On the top-left corner lies the original graph $\mathcal{G}$, which has four nodes. The remaining graphs are five nodes graphs that contains $\mathcal{G}$ as an induced subgraph, and form a Hasse diagram where the rightest graph is the greatest element and the leftest graph is the least element. The numbers written below the graphs are their corresponding optimal spectral radii. As mentioned, the value of $\rho(\mathcal{G})$ forms a cut in the Hasse diagram, which is shown as the dashed line in Figure 2.4.

### 2.5.3 Simulation Result

We consider a two-wheel communication network structure where the two wheels have ten nodes each and are joined by sharing one of them, as shown in Fig. 2.5. This example is chosen to manifest the benefit of optimizing the choice of local network in contrast to directly following the objective dependency. We consider the decision variable can be split into three parts $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)$, with the left ring being the first local dependency set $\mathcal{N}_1 = \{i : i = 2, \ldots, 11\}$, the right ring being the third local dependency set $\mathcal{N}_3 = \{i : i = 11, \ldots, 20\}$, and $\mathcal{N}_2 = \{1, 11, 21\}$. We assume each $\mathbf{x}^m \in \mathbb{R}^2$, $m = 1, 2, 3$. Intuitively, since each part depends on less than half of the nodes, the full localization scheme would represent at least a $\sqrt{2}$ saving from the

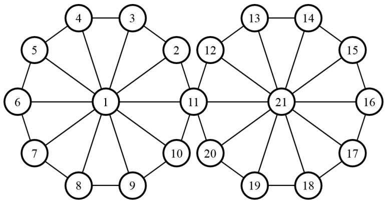


Figure 2.5: The two-wheel graph.

$\sqrt{n_m}$ term for the convergence rate. If we include node 1 into $\mathcal{N}_1'$ and node 21 into $\mathcal{N}_3'$, though this slows down the convergence for adding additional nodes, the two additional nodes may work as communication facilitators and reduce $\mu_1^{*\prime}$ and $\mu_3^{*\prime}$ so that overall the iterations converge faster. From our experiment, this is indeed the case.

From the description, it is clear that the nodes in $\{2, \ldots, 10\}$ depend only on $\mathbf{x}^1$, the nodes in $\{12, \ldots, 20\}$ depend only on $\mathbf{x}^3$, node 1 and 21 depend only on $\mathbf{x}^2$, and node 11 depends on all three parts. These define the dependent part $\mathcal{S}_i$ for each node $i$. We consider the simplest quadratic case. Specifically, for all $i$, $f_i$ is given as

$$f_i(\mathbf{x}^{\mathcal{S}_i}) = \frac{1}{2}\mathbf{x}^{\mathcal{S}_i \top}\mathbf{M}_i^\top \mathbf{M}_i \mathbf{x}^{\mathcal{S}_i} + \mathbf{b}_i^\top \mathbf{x}^{\mathcal{S}_i}, \tag{2.22}$$

where $\mathbf{M}_i$ is a matrix of proper dimension with *i.i.d.* elements from Unif$[-1, 1]$, and $\mathbf{b}_i$ is a column vector of proper dimension with *i.i.d.* elements from Unif$[-150, -50]$. The negative choices of $\mathbf{b}_i$ is to make the optimal solution for each coordinate mostly falls in $\mathbb{R}^+$, and the mean is around 135. The initial choice of $\mathbf{x}_i^{\mathcal{S}_i}[0]$ for every node is drawn *i.i.d.* for all coordinates from $\chi^2(50)$. Our choice of the learning parameter schedule is $\alpha[t] = \frac{0.8}{(t+1)^{0.8}}$, instead of the optimal power of $\frac{1}{3}$ described in [55], since we find the later unstable in the first few iterations.

Fig. 2.6 shows how the relative objective error $(F(\bar{\mathbf{x}}[t]) - F(\mathbf{x}^*))/F(\mathbf{x}^*)$ decreases with iterations for three different schemes, where $\bar{\mathbf{x}}[t]$ is the ensemble of $\{\bar{\mathbf{x}}^{1,l}, \bar{\mathbf{x}}^{2,l}, \bar{\mathbf{x}}^{3,l}\}$ and $\bar{\mathbf{x}}^{m,l}[t] = \frac{1}{n_m}\sum_{i\in\mathcal{N}_m} \mathbf{x}_i^m[t]$ is as defined before. Denote $\mathcal{G}_1'$, $\mathcal{G}_2'$, and $\mathcal{G}_3'$ the local network actually used in the schemes for the parts $\mathbf{x}^1$, $\mathbf{x}^2$, and $\mathbf{x}^3$, respectively. We assume that any edge is exploited whenever its both nodes are in the network, so that $\mathcal{E}_m' = \{(i, j) \in \mathcal{E} : i \in \mathcal{N}_m' \text{ and } j \in \mathcal{N}_m'\}$ and we only have to specify $\mathcal{N}_m'$'s. The first two schemes are straightforward: the no localization scheme requires all nodes to maintain all three parts $\mathcal{N}_1' = \mathcal{N}_2' = \mathcal{N}_3' = \mathcal{N}$, while the full localization scheme

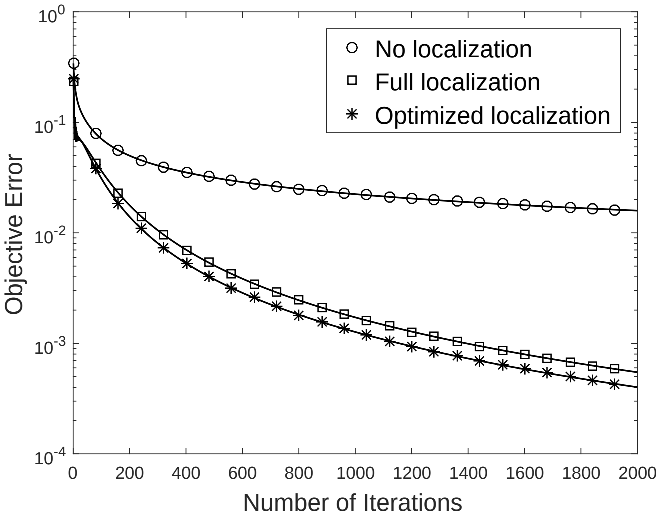


Figure 2.6: Objective errors for three different schemes.

directly follows the objective dependency $\mathcal{N}_1' = \mathcal{N}_1$, $\mathcal{N}_2' = \mathcal{N}_2$, and $\mathcal{N}_3' = \mathcal{N}_3$. For the optimized localization scheme, we consider adding the facilitator node 1 to $\mathcal{N}_1'$ and facilitator node 21 to $\mathcal{N}_3'$ at the cost of larger communication overhead and memory storage and observe if this fasten the convergence; for the second part $\mathcal{N}_2' = \mathcal{N}_2$ remains the same, since there is no further way to decrease $\mu_2$. The scheme is optimal in the sense of convergence speed, that is, we set $c_c$ and $c_m$ to 0 and only consider $t_{\min}$ in (2.17). We do not have to enumerate all possible choices though, since only adding 1 to $\mathcal{N}_1'$ and 21 to $\mathcal{N}_3'$ are helpful in reducing $\mu^{*\prime}$ and have the potential to beat the full localization scheme.

From the figure, we observe that the no localization scheme converges much slower than its localization counterparts, as the both rings consisting of the nodes sets $\mathcal{N}_1$ and $\mathcal{N}_3$ keep sending their decision of $\mathbf{x}^1$ and $\mathbf{x}^3$ to each other through a single node 11, while only the objectives in $\mathcal{N}_1$ depend on $\mathbf{x}^1$ so that those nodes can completely dictate the decision as the nodes not in $\mathcal{N}_1$ are merely updating $\mathbf{x}^1$ from the information sent from $\mathcal{N}_1$, and same for $\mathcal{N}_3$ and $\mathbf{x}^3$. The middle line consisting of the nodes set $\mathcal{N}_2$ also tries to spam the whole network with their new decisions of $\mathbf{x}^2$. The optimized localization scheme performs slightly better than the full localization scheme. Adding the two facilitator nodes hurts the convergence rate by increasing $\sqrt{n_1'}$ and $\sqrt{n_3'}$, but benefits from reduced $\mu_1^{*\prime}$ and $\mu_3^{*\prime}$. It turns out that in this case the reduced $\mu$ outweighs, since $n_1 = n_3 = 10$ is already large enough and adding one more node does not hurt too much, while the decrease in $\mu_1^{*\prime}$ and $\mu_3^{*\prime}$ is substantial.

Table 2.2: Communication cost and memory storage required for three different schemes.

| Scheme | Communication cost | Memory storage |
|---|---|---|
| No localization | 603840 | 126 |
| Full localization | 16016 | 46 |
| Optimized localization | 25200 | 50 |

Suppose we require the algorithms to achieve a relative objective error smaller than 0.02 before termination. Then the three schemes require 1258, 182, and 150 iterations, respectively (see $t_{\min}$ defined in (2.16)). Table 2.2 summarizes the communication cost and memory storage required for the three schemes. Not surprisingly, the no localization scheme needs the largest amount of communication and memory. The optimized localization scheme necessarily requires more memory than full localization. The communication cost is the cost per iteration times the total number of iterations needed to meet the termination criteria. We can see that in this case, the communication cost per round approximately doubles when the ring structure turns to wheel structure, but the number of iterations required is only reduced by around $\frac{5}{6}$, so that overall the cost for optimized localization is higher.

In summary, generally speaking no localization has the worst convergence as well as communication and memory performance. Optimized localization converges the fastest (since it is optimal in terms of having the smallest $t_{\min}$), while full localization requires the least memory storage. Depending on different objectives, graph structures, and termination criteria, the scheme that requires least amount of total communication varies. Recall that total communication is the product of $t_{\min}$ and communication per round. On one side, optimized localization has the smallest $t_{\min}$; on the other side, full localization requires the least amount of communication per round. This exhibits a tradeoff between $t_{\min}$ and communication per round, and where the minimum product lies depends.

## 2.6 Application to Resource Allocation

In this section, we present how to apply the Localized Proximal Inexact NEXT (Algorithm 2.2) to wireless resource allocation, and along way also describe how the two issues that motivated our generalizations arise.

### 2.6.1 Problem Formulation

We consider an OFDMA wireless cellular network, where a set of base stations (BSs) $B$ transmit downlink data to users through the set of channels or resource blocks (RBs) $K$. For a BS $b \in B$, $I_b$ denotes the set of users associated with it, which is an input that's fixed. The transmitted power of BS $b$ in channel $k$ is denoted by $p_{bk}$, and the maximum total sum power transmitted by BS $b$ is limited to $P_b$. The allocation variable of BS $b$ to user $i$ in channel $k$ is denoted by $x_{bik}$, with gain $g_{bik}$: $x_{bik} = 1$ means $b$ transmits to $i$ in the RB $k$, and $x_{bik} = 0$ otherwise. For all users, we also introduce a scheduling weight, $w_i$ for user $i$. Finally, $\sigma^2$ is the variance of the independent zero mean additive white Gaussian noise (AWGN) for all BSs. We assume that BS $b$ only possesses the information of $\{g_{b'ik} : b' \in B, i \in I_b, k \in K\}$. In other words, BS $b$ can only compute the weighted-sum rate of the users associated with itself (knowing the powers of the other base-sites). This is a reasonable assumption, as each user equipment (UE) reports its measured channel gains to the BS it is associated with, whereas all the channel gains of UEs served by other BSs is unknown.

When the BS $b$ transmits a non-zero power in channel $k$, it interferes with all the other transmissions in channel $k$. However, owing to propagation-based loss, the powers of nearby BSs will dominate the whole interference term. Hence, with the definition that $N(b)$ is the neighboring BSs of BS $b$, we can neglect all the interference from $b' \notin N(b)$ to $b$. This is a modeling assumption that is reasonably accurate in practice, and will be in force henceforth. For ease of exposition we assume that neighbor relation is mutual, i.e. $b' \in N(b)$ if and only if $b \in N(b')$. In terms of the interference graph, where nodes are BSs and edges only exist between BSs that interfere with each other, we reduce a complete graph to an undirected and connected one. Our work can be trivially extended to the directed case assuming that strong connectivity holds.

We consider a one-shot weighted sum-rate maximization problem, subject to the allocation limit constraint, the power limit constraint, non-negative power constraint, and the fact that $x_{bik}$ is either 0 or 1; we will justify the weighted sum-rate maximization problem in Section 2.6.6. To make the overall constraint set convex, we relax the integer constraints on $x_{bik}$ as in [51, 52]. In future work we will study appropriate integer rounding schemes.

The joint power control and scheduling problem (P1) is then formalized as:

$$\begin{aligned}
\text{(P1)} \qquad & \max_{p_{BK}, x_{BI(B)K}} \sum_{b \in B} \sum_{i \in I_b} w_i \sum_{k \in K} x_{bik} \log\left(1 + \frac{\Gamma_{bik}}{\sigma^2 + \bar{\Gamma}_{bik}}\right) \\
& \text{subject to } \sum_{i \in I_b} x_{bik} \le 1 \quad \forall\ b \in B, k \in K \\
& \qquad \sum_{k \in K} p_{bk} \le P_b \quad \forall\ b \in B \\
& \qquad 0 \le x_{bik} \le 1 \quad \forall\ b \in B, k \in K, i \in I_b \\
& \qquad 0 \le p_{bk} \quad \forall\ b \in B, k \in K,
\end{aligned} \tag{2.23}$$

where $\Gamma_{bik} = p_{bk} g_{bik}$ and $\bar{\Gamma}_{bik} = \sum_{b' \in N(b)} p_{b'k} g_{b'ik}$ are the signal and interference for user $i$ in channel $k$. We use $p_{BK}$ to refer to the collection of the variables $p_{bk}\ \forall\ b \in B, k \in K$; also, $x_{BI(B)K}$ can be viewed in a similar way, where $I(B) \triangleq \bigcup_{b \in B} I_b$. This is a shorthand for easy referencing.

As we will solve the problem in a distributed manner, we let each BS maintain the decision variables $p_{BK}$. Denote the copy of $p_{b'k}$ at BS $b$ by $p^b_{b'k}$ for all $b' \in B, k \in K$. The idea is to perform the optimization at each BS, and then enforce consensuses of the decision variables among all BSs, transforming (P1) into (P2) given in the following:

$$\text{(P2)} \qquad \max_{p^B_{BK}, x_{BI(B)K}} \sum_{b \in B} \sum_{i \in I_b} w_i \sum_{k \in K} x_{bik} \log\big(1 + \frac{\Gamma^b_{bik}}{\sigma^2 + \bar{\Gamma}^b_{bik}}\big) \tag{2.24}$$

where $\Gamma^b_{bik} = p^b_{bk} g_{bik}$ and $\bar{\Gamma}^b_{bik} = \sum_{b' \in N(b)} p^b_{b'k} g_{b'ik}$. The set of constraints includes all the constraints in (2.23) now with $p^b_{b'k}$ and the second and fourth constraints hold for all copies at $b \in B$, and an additional constraint that the consensus is reached $p^{b_1}_{b'k} = p^{b_2}_{b'k}\quad \forall\ b_1, b_2, b' \in B, k \in K$.

We can split the $\log(\cdot)$ term in the objective to two parts $x_{bik} \log(\sigma^2 + \Gamma^b_{bik} + \bar{\Gamma}^b_{bik})$ and $-x_{bik} \log(\sigma^2 + \bar{\Gamma}^b_{bik})$, and then modify the former to be $x_{bik} \log(\sigma^2 + \frac{\Gamma^b_{bik} + \bar{\Gamma}^b_{bik}}{x_{bik}})$ as in [51, 52], which is jointly strictly concave in $x_{bik}$ and $p^b_{Bk}$. We define the modified function to be 0 when $x_{bik} = 0$ so that it is continuous. Then (P2) becomes the following:

$$\text{(P3)} \quad \max_{p^B_{BK}, x_{BI(B)K}} \sum_{b\in B} \sum_{i\in I_b} w_i \sum_{k\in K} \left[ x_{bik} \log\left(\sigma^2 + \frac{\Gamma^b_{bik} + \bar{\Gamma}^b_{bik}}{x_{bik}}\right) - x_{bik} \log\left(\sigma^2 + \bar{\Gamma}^b_{bik}\right)\right], \tag{2.25}$$

subject to the same set of constraints as in (2.24). Note that (P2) and (P3) are the same if $x_{bik}$'s are restricted to be integers, that is, $x_{bik} \in \{0, 1\}$.

We will now reiterate the two issues we identified earlier with existing distributed optimization approaches but in the specific context of (2.25). If we take the approach in [80], then every BS would need to keep a copy of all the decision variables, both $p_{BK}$ and $x_{BI(B)K}$, and perform consensus on them and also any relevant gradient terms. This is simply impractical and forces the localization idea. We also note that (2.25) contains functions of the form $x \log\bigl(a + (p + p')/x\bigr) - x \log(a + p')$ that are smooth but where the gradients are not Lipschitz; in particular $x \log(a + (p+p')/x)$ has some terms of its gradient becoming infinite when $x \downarrow 0$. These functions clearly fall outside the framework of [80], and force approaches like our proximal approximations scheme. In the following, we apply the developed distributed optimization framework to the problem in (2.23)-(2.25). In the problem, the set of BSs in the cellular network $B$ corresponds to $\mathcal{N}$ in the framework, a BS $b$ corresponds to a node $i$, and the set of edges $\mathcal{E}$ in the framework is the one-tier interference graph here. There are many different ways to apply our framework to the resource allocation, which we discuss in detail next.

### 2.6.2 Direct Method

In this method, we directly let $f_b$ be the weighted sum-rate of BS $b$: $f_b = -\sum_{i,k} w_i x_{bik} \log\left(1 + \frac{\Gamma_{bik}}{\sigma^2 + \bar{\Gamma}_{bik}}\right)$, and $G = 0$. In the first version of this method, every BS $b$ keeps the powers of all BSs as decision variables, that is, we have $p^b_{b'k}$ for all $b, b' \in B$. But instead of keeping $x^b_{b'ik}$'s as decision variables at BS $b$ for all $b' \in B$ if we exactly follow NEXT, we allow every BS $b$ only keeps its own $x^b_{bik}$, which we denote as $x_{bik}$ for short. As we see from Section 2.3.1 and Section 2.4.1, this suffices because BS $b$ dictates the decision of $x_{bik}$.

If we reconsider the problem from the perspective of our localization framework Section 2.2.3, there are $2|B|$ local dependency sets. There are $|B|$ local dependency sets $\{b\}$ for all $b \in B$ which correspond to the variables $x_{bI_bK}$, and $|B|$ local dependency sets $Nb(b) \triangleq N(b) \cup \{b\}$ for all $b \in B$ which correspond to the variables $p_{bK}$. In the first version of the direct method, referred to as the Localized X Globalized P-diRect Method (LXGP-RM) algorithm, only $x_{BI(B)K}$ follows the localization framework, $p_{BK}$ is still globalized in the sense that every BS keeps a copy of the whole variable.

We only describe the algorithm in words here, for the pseudo code see Appendix B. The algorithm basically proceeds as Algorithm 2.2 with the common variable $p_{BK}$ and $|B|$ local variables $x_{bI_bK}$ $\forall$ $b$ (and without approximation functions since objectives have Lipschitz gradients). At BS $b$ we use $r^b_{b'k}$ to track $\frac{1}{|B|}\sum_{b''\in B}\frac{\partial f_{b''}}{\partial p_{b'k}}$ and $\tilde{\pi}^b_{b'k}$ to track $\sum_{b''\neq b\in B}\frac{\partial f_{b''}}{\partial p_{b'k}}$, which correspond to $\mathbf{y}$ and $\pi$ in Algorithm 2.2. In each iteration, we let $\alpha[t] = \frac{\alpha_0}{(t+1)^\beta}$, and BS $b$ performs the minimization of

$$\tilde{f}_b(p^b_{BK}, x_{bI(b)K}; \bar{p}^b_{BK}, \bar{x}_{bI(b)K}) + \tilde{\pi}^b_{BK} \cdot (p^b_{BK} - \bar{p}^b_{BK}) \tag{2.26}$$

with respect to the variables $p^b_{BK}$ and $x_{bI(b)K}$, and $\bar{p}^b_{BK}$, $\bar{x}_{bI(b)K}$, and $\tilde{\pi}^b_{BK}$ being the current iterate of the variables. The surrogate function $\tilde{f}_b(\mathbf{p}^b, \mathbf{x}_b; \bar{\mathbf{p}}^b, \bar{\mathbf{x}}_b)$ is chosen as

$$\begin{aligned} &f_b + \frac{\tau_b}{2}\Big[\sum_{i,k}(x_{bik} - \bar{x}_{bik})^2 + \sum_{b'\in B,k}(p^b_{b'k} - \bar{p}^b_{b'k})^2\Big] \\ &+ \sum_{i,k}\frac{\partial f_b}{\partial x_{bik}}\cdot(x_{bik} - \bar{x}_{bik}) + \sum_{b'\in Nb(b),k}\frac{\partial f_b}{\partial p^b_{b'k}}\cdot(p^b_{b'k} - \bar{p}^b_{b'k}), \end{aligned} \tag{2.27}$$

where $f_b$ and $\frac{\partial f_b}{\partial p^b_{b'k}}$ are functions of $(\bar{\mathbf{p}}^b, \bar{\mathbf{x}}_b)$, but $\frac{\partial f_b}{\partial x_{bik}}$ is just a function of $\bar{\mathbf{p}}^b$. The quadratic term in (2.27) is to maintain the strict convexity of the surrogate. Finally, we have a universal doubly stochastic matrix $W$ to average $p_{BK}$ and $r_{BK}$ for them to reach consensus. We remark that with the objective in (2.26) only having up to quadratic terms and our constraints being linear, the minimization can be solved efficiently using quadratic programming (QP) with coefficient matrices of the quadratic term being positive-semidefinite [65].

The second version of the direct method, termed as the Localized X Localized P-diRect Method (LXLP-RM) algorithm, makes better use of our localization idea as opposed LXGP-RM so that for all $b \in B$ we now only maintain the variables $p^b_{b'k}$

for $b' \in Nb(b)$ in BS $b$, as the BSs in $Nb(b')$ dictate the decision of $p_{b'k}$; the variables $x_{BI(B)K}$ still follow the localization framework just as before in LXGP-RM.

The main change from LXGP-RM is that the index set of the variable tuple $B$ is now replaced by $Nb(b)$, as BS $b$ does not keep the variable $p^b_{b'k}$ for $b' \notin Nb(b)$ any more. As a result, the steps regarding the weighted sum for reaching consensus need to be modified. We introduced the matrix $W(b)$ for each BS $b$. The matrix $W(b)$ concerns with the weighting of the local dependency set $Nb(b)$ regarding the variable $p_{bK}$, that is, $\mathcal{G}(b) = (Nb(b), \mathcal{E}(b))$ where $\mathcal{E}(b) = \{(i,j) \in \mathcal{E} : i, j \in Nb(b)\}$ if the whole network is $\mathcal{G} = (B, \mathcal{E})$. Its $i$-th row $W_{i:}(b)$ and $j$-th column $W_{:j}(b)$ should be zero if and only if $i \notin Nb(b)$ or $j \notin Nb(b)$. After deleting all these zero rows and columns, it would become a doubly-stochastic matrix, as described in Assumption L4.

The second version of the direct method still follows the framework of Algorithm 2.2, with $2|B|$ local variables $p_{bK}, x_{bI_bK}$ $\forall$ $b$ and no common variable. Now at BS $b$ we use $r^b_{b'k}$ to track $\frac{1}{|Nb(b')|}\sum_{b'' \in Nb(b')} \frac{\partial f_{b''}}{\partial p_{b'k}}$, and $\tilde{\pi}^b_{b'k}$ to track $\sum_{b'' \neq b \in Nb(b')} \frac{\partial f_{b''}}{\partial p_{b'k}}$, where $b' \in Nb(b)$. Also, the surrogate $\tilde{f}_b$ is minorly changed so that now the quadratic term of $p^b_{b'k}$ only sums over $b' \in Nb(b)$ instead of $b' \in B$.

We finally remark that one can also consider a GXGP-RM algorithm (basically NEXT from [80]) where copies of both the power and allocation variables are maintained at each node, or even a GXLP-RM algorithm where localization is performed only for the power variables. Note that only the fully localized scheme, i.e. LXLP-RM, will be scalable in practice. However, we will evaluate its performance relative to the other schemes.

### 2.6.3 Decomposed Method

In (P3), there is a part of the objective that is concave (or convex after taking minus sign), and the optimization of this part should be easy. The algorithm might run faster if we properly exploit this fact. To achieve this goal, let us assume that the channel gains $g_{bik}$'s are known to all BSs. Then we could apply the framework in Section III by letting $f_b = \sum_{i,k} w_i x_{bik} \log(\sigma^2 + \bar{\Gamma}_{bik})$ and $G = -\sum_{b,i,k} w_i x_{bik} \log\left(\sigma^2 + \frac{\Gamma_{bik} + \bar{\Gamma}_{bik}}{x_{bik}}\right)$.

As $G$ is in general a function of not only $p_{bk}$ but also $x_{bik}$ for all $b \in B$ and we need to optimize $G$ at every BS, this method does not allow any localization. In other words, the tuple consisting of all decision variables is the common variable $\mathbf{x}^c$ in Algorithm 2.2 itself, and there is no local dependency set besides $\mathcal{N}$ itself. At BS $b$ we need to maintain $p^b_{b'k}$ as well as $x^b_{b'ik}$ $\forall$ $b' \in B$. Note that with the derivatives of $f_b$ being Lipschitz continuous and no localization, this method is a direct application of [80].

The algorithm, which we call the Globalized X Globalized P-deComposed Method (GXGP-CM), is also largely the same as LXGP-RM, except that now we have to optimize $G$ as well, and we need to maintain and update $x^b_{b'ik}$. The algorithm and the surrogate function also need minor revisions (see Appendix A.6).

### 2.6.4 Partially Linearized Method

While the decomposed method enjoys the benefit of using the intrinsic convex part in the objective, it is impractical since it requires every BS knows all channel gains. We could instead put the convex part in $f_b$ as well, and take advantage of it by not linearizing it when forming the surrogate function. Specifically, we let $f_b = f_{b\cup} + f_{b\cap}$ where $f_{b\cup} = -\sum_{i,k} w_i x_{bik} \log\left(\sigma^2 + \frac{\Gamma_{bik}+\bar{\Gamma}_{bik}}{x_{bik}}\right)$ and $f_{b\cap} = \sum_{i,k} w_i x_{bik} \log\left(\sigma^2 + \bar{\Gamma}_{bik}\right)$. Then we can choose the surrogate function $\tilde{f}_b(\mathbf{p}^b, \mathbf{x}_b; \bar{\mathbf{p}}^b, \bar{\mathbf{x}}_b)$ as $f_{b\cup}(\mathbf{p}^b, \mathbf{x}_b) + \tilde{f}_{b\cap}(\mathbf{p}^b, \mathbf{x}_b; \bar{\mathbf{p}}^b, \bar{\mathbf{x}}_b)$, where $\tilde{f}_{b\cap}$ has exactly the same form as in (2.27) (for LXGP case), i.e., linearized with the current iterate $(\bar{\mathbf{p}}^b, \bar{\mathbf{x}}_b)$ plus the quadratic terms.

With $\nabla f_b$ not being Lipschitz continuous, we apply the approximation functions detailed in Section 2.2.4 for better convergence and numerical stability. We may choose $f^*_{b,t} = f^*_{b\cup,t} + f_{b\cap}$ where

$$f^*_{b\cup,t} = -\sum_{i,k} w_i(x_{bik} + e[t]) \log\left(\sigma^2 + \frac{\Gamma_{bik} + \bar{\Gamma}_{bik}}{x_{bik} + e[n]}\right). \tag{2.28}$$

One can easily show that $\nabla f^*_{b,t}$ is Lipschitz continuous with constant $L_{b,t}$ the reciprocal of $e[t]$. We can then choose a schedule of $e[t] \to 0$ according to Theorem 2.3 and Section 2.4.5. We refer to this method as the Partially Linearized method (PL) algorithm, which could be LXLP or any of the other combinations. Note that we do not have the guarantee of convergence to stationary point in this case because the objective function has unbounded gradient on the boundary.

### 2.6.5 Consensus Scheme

Let $\mathcal{G} = (B, \mathcal{E})$ be the BS network we are considering, and let $d_i$ be the degree of BS $i$. The choice of $W$ must meet the following two criteria to conform to Assumption L4: (1) it must be doubly-stochastic; (2) $W_{ij} \geq 0$ is non-zero if and only if $(i, j) \in \mathcal{E}$. We denote the set of $\mathbf{W}$'s that satisfy these criteria as $\Omega(\mathbf{W})$, which is a subset in $\mathbb{R}_+^{|B|\times|B|}$. We choose $\mathbf{W}$ as follows

$$W_{ij} = \begin{cases} 0 & \text{if } j \notin N(i) \\ \frac{1}{\bar{d}} & \text{if } j \in N(i) \text{ and } i \neq j \\ \frac{\bar{d}-d_i}{\bar{d}} & \text{if } j \in N(i) \text{ and } i = j \end{cases}, \tag{2.29}$$

where $\bar{d} = \max_i d_i + 1$. It is easy to verify that this choice of $\mathbf{W}$ is row-stochastic. Since $\mathbf{W}$ is symmetric, it is then also column-stochastic. By definition of $\bar{d}$, we will have $W_{ii} > 0$. By construction, $W_{ij} > 0$ if $(i, j) \in \mathcal{E},\ \ i \neq j$.

In LXLP-RM algorithm we need a $\mathbf{W}(b)$ for every BS $b$. Let $\mathcal{E}(b) = \{(i, j) \in \mathcal{E} : i, j \in Nb(b)\}$. Then we choose $\mathbf{W}(b)$ as described above but treat $\mathcal{G}$ as $\mathcal{G}(b) = (Nb(b), \mathcal{E}(b))$.

As we discussed in Section 2.5.2, with symmetric weights $\mathbf{W} = \mathbf{W}^\top$, [77] suggests that the best convergence speed is obtained with the solution of (see also Fact 2.15)

$$\min_{\mathbf{W}\in\mathbb{R}_+^{|B|\times|B|}} \left\| \mathbf{W} - \frac{1}{|B|}\mathbf{1}_{|B|}\mathbf{1}_{|B|}^\top \right\|_2 \text{ s.t. } \mathbf{W} \in \Omega(\mathbf{W}). \tag{2.30}$$

For symmetric graphs, the optimized result is $W_{ij} = \frac{1}{d_i}$ when $(i, j) \in \mathcal{E}$ and $W_{ij} = 0$ otherwise, which is exactly our choice.

### 2.6.6 Simulation Results

We adopt the framework of the network utility maximization problem as in [51, 52] where we maximize

$$U(\mathbf{R}_T) = \sum_{i\in I(B)} U_i(R_{i,T}), \tag{2.31}$$

where $U_i(\cdot)$ is given by

$$U_i(\mathbf{R}_{i,t}) = \begin{cases} \frac{c_i}{\xi}(R_{i,t})^{\xi}, & \xi \leq 1, \xi \neq 0, \\ c_i \log(R_{i,t}), & \xi = 0, \end{cases} \tag{2.32}$$

$R_{i,t}$ is the average throughput of user $i$ up to time $t$, $\xi \leq 1$ is the fairness parameter, and $c_i$ is a QoS weight. The gradient-based scheduling approach [92] leads to solving the optimization problem given below at each time instance

$$\max_{\mathbf{r}_t \in \mathcal{R}(e_t)} \sum_i c_i (R_{i,t})^{\xi-1} r_{i,t}. \tag{2.33}$$

This is exactly the one-shot optimization problem we consider in (2.23), where $w_i = c_i(R_{i,t})^{\xi-1}$ is the weight of user $i$, $r_{i,t}$ is the rate of user $i$ given by the Shannon capacity, and $\mathcal{R}(e_t)$ is the capacity region dependent on current channel state $e_t$ and constrains the choice of $r_{i,t}$ as the constraints set in (2.23).

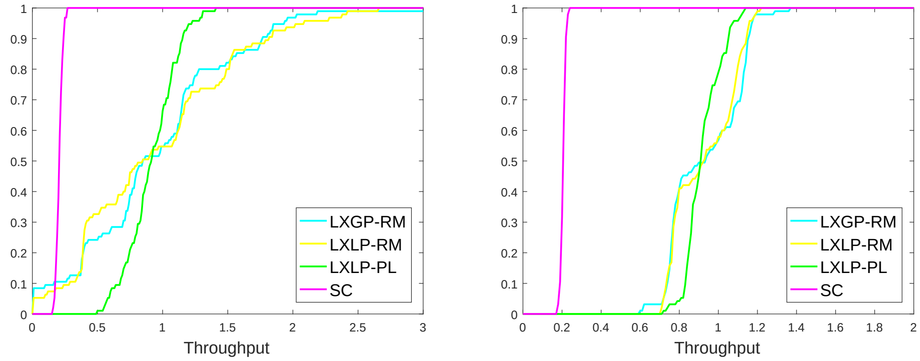
Now consider problem (2.23). A naive solution would be disregarding the interference and solving the resource allocation and scheduling for each cell separately. The optimization for a single-cell is well solved in literature, e.g. in [51]. Specifically, neglecting the interference, for a BS $b$ we can solve

$$\max_{p_{bK}, x_{bI_bK}} \sum_{i \in I_b} w_i \sum_{k \in K} x_{bik} \log\left(1 + \frac{p_{bk} g_{bik}}{\sigma^2 x_{bik}}\right), \tag{2.34}$$

subject to the constraints. This is a convex problem, and can be solved with existing methods in convex optimization. We call this method the Single-Cell No-Iteration (SC-NI) algorithm.

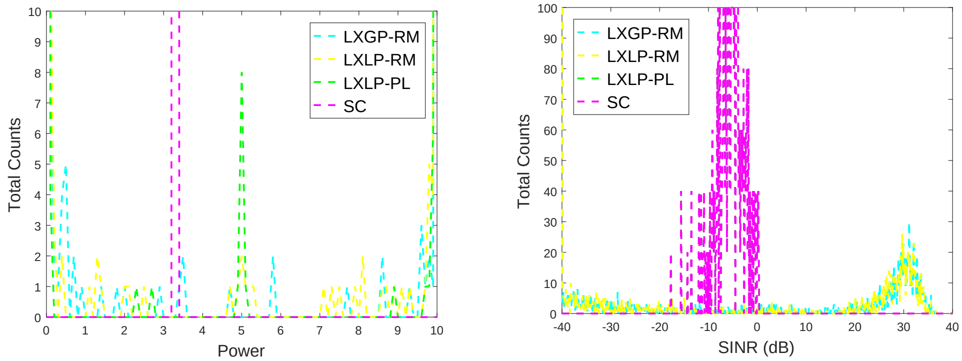
A refinement of the SC-NI algorithm is to update the interference terms after first optimization for each cell. We then optimize for each cell again while treating the powers of neighboring BSs as constants, and then repeat until convergence. We call this the Single-Cell (SC) algorithm.

We adopt the 19 cell wrap-around model from [53] as the network scenario used in our simulations. Furthermore, each UE associates with exactly one BS and each BS has five UEs associated with it. Suppose a UE is served by a BS. Then there is a signal



(a) $\xi = 1$ (b) $\xi = 0.5$

Figure 2.7: Empirical CDFs of users' throughputs for $\xi = 1$ and $\xi = 0.5$.



(a) Distribution of transmission power. (b) Distribution of SINR

Figure 2.8: Distributions of transmission power and SINR for $\xi = 1$.

link between the UE and the BS, while all the neighbors of the BS cause interferences to the UE. The time horizon $T$ is chosen to be 20. The channel gains are directly generated by Rayleigh distribution, with parameter 1 for associated BS-UE pair, and 0.5 for interference, instead of choosing random locations for the UEs and calculating the path loss. The channel gains are assumed to be independent among all links in a scheduling instance and also across all scheduling instances. We use identical QoS weights ($c_i = 1$). Other parameters include: $|K| = 3$, $\sigma^2 = 0.01$, $\alpha_0 = 0.99$, $\beta = 0.53$, and $P_b = 10 \ \forall \ b$. For simplicity we treat all scheduling terms $x_{bik}$ as real numbers and use the local optimal results to compute utilities. In future work we will include integer rounding procedures in the simulations. The entire process is simulated only one time, as multiple time slots already brought in the averaging effect.

Figure 2.7 depicts the CDFs of user throughput of algorithms LXGP-RM, LXLP-RM, LXLP-PL, and SC for $\xi = 1$ (maximum total throughput) and $\xi = 0.5$, respec-

Table 2.3: Fraction of utilized channels and scheduled users per BS.

| Channels/Users | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| LXLP-RM Channels | 0.3447 | 0.6316 | 0.0237 | 0 | - | - |
| LXLP-RM Users | 0.3447 | 0.5395 | 0.1158 | 0 | 0 | 0 |
| LXLP-PL Channels | 0.3763 | 0.6105 | 0.0132 | 0 | - | - |
| LXLP-PL Users | 0.3763 | 0 | 0 | 0 | 0.0079 | 0.6158 |
| SC Channels | 0 | 0 | 0 | 1 | - | - |
| SC Users | 0 | 0 | 0 | 0 | 0 | 1 |

tively. When $\xi = 0.5$, we can observe that the RM and PL methods stochastically dominate the SC algorithm, and this is also nearly the case when $\xi = 1$. In fact, the RM and PL methods yield a roughly 4-fold average throughput gain over SC method. This is not surprising, as we simulate a rich interference environment, and the new methods can coordinate the scheduled UEs and transmission powers of nearby BSs, while the SC method does not. The RM methods show similar performance as we use the same termination criteria. They are a little bit different from the PL method possibly due to optimizing different objective functions (only equivalent before integer relaxation).

Figure 2.8 illustrates the power and SINR distributions of the same four algorithms for $\xi = 1$. The proposed three new methods choose one of three values for the power: the maximum, half of the maximum or zero. This corresponds to assigning either one, two or no blocks; the two blocks can be assigned to one UE or two. On the contrary, the SC methods schedule all three blocks and UEs with the power to each block around $\frac{10}{3}$. At each time instance the three new methods give up serving some subcarriers and users in exchange of boosting the SINR of the scheduled UEs on the chosen blocks. On the other hand, the SC method tries to serve everyone, and ends up with lower power and increased interference. The details of the scheduling decisions are in Table 2.3.

Table 2.4 compares the performance of the four algorithms with $\xi = 0.5$ and $\xi = 1$. The three coordination-based methods outperform the SC method significantly in this problem instance. The LXGP method always requires more iterations to converge than the fully-localized methods like the LXLP family. When $\xi = 0.5$, LXLP-PL converges much faster than RM methods.

Table 2.4: Total utilities and average numbers of iterations required of different algorithms and choices of $\xi$.

| Algorithm | LXGP-RM | LXLP-RM | LXLP-PL | SC |
|---|---|---|---|---|
| Utilities | | | | |
| $\xi = 1$ | 89.61 | 89.73 | 86.98 | 19.62 |
| $\xi = 0.5$ | 182.7 | 182.5 | 182.1 | 86.10 |
| Iterations required | | | | |
| $\xi = 1$ | 280.3 | 176.9 | 176.2 | 2 |
| $\xi = 0.5$ | 273.6 | 216.9 | 142.9 | 2 |

## 2.7 Conclusion and Future Directions

In this chapter, we generalized existing distributed optimization methods in two directions. First, we reduced the algorithm storage and communication complexity by exploiting a decomposable structure of the problem, and obtained a localization scheme that works for primal-consensus type algorithms, which we demonstrated through NEXT and DGD. We analyzed the effect of introducing localization to the convergence rate, and found the later's dependence on the local network size and the spectral radius of the local averaging matrix corresponding to the local network, which raises the problem of optimizing local networks and our study of the dependency of the optimal spectral radius to the network topology. We also performed numerical simulations that shows the benefits of adopting our localization scheme, including lower memory and communication consumption as well as faster convergence. Second, we relaxed the requirement of Lipschitz continuous gradients with a series of slowly-changing approximations (for NEXT). We found in simulations that our approximation scheme endows much better numerical stability. Finally, we applied the developed algorithmic framework in different ways to generate distributed algorithms for the multi-cell resource allocation problem. We compared these algorithms with the single-cell algorithm via simulation and showed the potential gains of using the distributed optimization methods.

For the localization part, an interesting direction would be looking for heuristic methods that can quickly decide the dependency of optimal spectral radius to the network topology (e.g. adding one node and a few edges will increase or decrease the optimal spectral radius), and find good sub-optimal solutions of the local network optimization problem, especially from the theory of conductance in graph theory. For the approximations part, extending our framework to dual distributed optimization methods is worth investigating.

## CHAPTER III

# Nonlinear Consensus for Distributed Optimization

## 3.1 Introduction

The network consensus problem, which concerns the convergence behavior of a multi-agent network reaching consensus, is widely studied [11, 77, 96, 28]. In the problem, each node in the underlying directed or undirected network is given an arbitrary initial value, and the nodes exchange their values through the network links to reach a consensus. The bulk of the literature on the network consensus problem has focussed on the following: choosing the consensus weight matrix to maximize the convergence speed [77, 28], analysis of the dependence of the convergence rate on the number of nodes and spectral radius of the weight matrix [96], and the conditions for convergence of different concensus schemes using stability analysis [11].

Whereas consensus schemes have broad applications, they are critical to distributed optimization. Nearly all distributed optimization algorithms are composed of two core steps: the "descent step" and the "consensus step." The bulk of the literature on distributed optimization studies various schemes of the descent process, but just takes simple linear consensus, so that the role of the consensus scheme in the context of distributed optimization has received less attention. Nonlinear gossip or consensus schemes are studied in [85] from the perspective of two time-scale stochastic approximation, with distributed optimization discussed as a special case. In the perspective, the fast process is repeated (possibly nonlinear) subprojections onto the consensus plane satisfying certain regulatory conditions, while the slow process tracks the trajectory of some ordinary differential equation (ODE). Commonly, the goal of the ODE is either to find the minima of the objective function, or to find the fixed points of some function, or to find the zeros of some function [21]. There has been follow-up work that uses the two time-scale stochastic approximation perspective but with the focus on problems different from convergence analysis. For example, the goal of [109] is to investigate projections onto the intersection of the constraint sets of the nodes, whereas in [29] concentration bounds for stochastic approximation algorithms with contractive maps are derived with application to convergence behavior of the value functions in reinforcement learning. While most distributed optimization algorithms simply alternate between the descent step and the consensus step each by once, another line of work studies the optimal frequency between them and analyzes the convergence rate specifically for the distributed optimization setting [13, 14]. In particular, unlike in the literature where a local optimization step (e.g. a local gradient descent step) is performed immediately followed by taking average once intermittently, [13, 14] suggest multiple applications of the "descent operator" followed by multiple applications of the "consensus operator" with certain schedules for convergence speed up.

In this chapter, we aim to answer the following questions: if one wants to adopt consensus schemes other than linear ones in distributed optimization algorithms, what a general class of schemes would be appropriate? Also, can we establish convergence to the optimum for the schemes, and contrast their convergence behavior with linear consensus schemes? Using the NEXT algorithm introduced in Chapter II as an example, we start by introducing the roles of the descent steps and the consensus steps and the relation between them from the perspective of stochastic approximation theory; this we accomplish by reproving the convergence of the NEXT algorithm. We study a class of nonlinear consensus schemes that exploit *nonlinear transformations*, where linear averages are taken after the transformations; convergence guarantees are established for modified versions of two distributed optimization algorithms – NEXT and DGD. Here, we generalize the result of [85] in the sense that we allow *time-varying* subprojections in the distributed optimization setting, which are characterized by the transformations in our schemes that are not necessarily fixed and can be varied at every step on the fly (using past information). For the NEXT algorithm, we also relax the column stochasticity requirement commonly assumed with the necessary row stochasticity in the literature, which leads to more flexible provably convergent algorithms; a similar counterpart algorithm for DGD with gradient tracking variables is also provided, with its convergence only conjectured. Finally, we combine the two ideas to obtain an even more general algorithm class and provide convergence guarantees for all of them. This more general algorithm class along with the family of $p$-means transformations allows us to choose any point in the “shrunk cube hull” (defined in Section 3.5.1) that ranges from element-wise minimum to element-wise maximum in the consensus step. Our analysis of the convergence of these algorithms along with some numerical examples demonstrates that the convergence rate critically depends on the relation between the consensus scheme and the gradient direction. We then use this as motivation to propose algorithms that *align consensus steps with negative gradient directions*.

The rest of the chapter is organized as follows. In Section 3.1.1 we briefly describe the setting as well as some terminologies for later usage. We provide an alternative proof of NEXT from the perspective of stochastic approximation in Section 3.2. In Section 3.3.1, we study the convergence behavior of nonlinear consensus schemes that use nonlinear transformations in a pure consensus problem setting (without coupling with optimization); then we show the convergence of NEXT and DGD when these nonlinear transformations are used in the consensus step. We then show the convergence of NEXT without column stochasticity in Section 3.4.1, while the convergence of the DGD algorithm counterpart with gradient tracked by another variable is left as a conjecture in Section 3.4.2. Combining the nonlinear transformation with column stochasticity relaxation, we further enlarge the possible choices for the next iterate in nonlinear consensus schemes from a “shrunk convex hull” described in Section 3.4 to a “shrunk cube hull” in Section 3.5.1, and propose algorithms that align the consensus step direction to the negative total gradient in Section 3.5.2. Simulations for the numerical examples and conclusion are given in Section 3.6 and Section 3.7, respectively.

### 3.1.1 Setup and Key Definitions

In this chapter, we consider a distributed optimization setup similar to that described in Section 2.2.1, except that we set the possibly non-smooth regularizer

$G = 0$ here, so that the overall objective is $F(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$. There are $n$ nodes (agents), and each node $i \in [n]$ maintain a copy of the decision variable $\mathbf{x}_i \triangleq [x_{i,1} \ \cdots \ x_{i,d}]^\top \in \mathbb{R}^d$. We denote the ensemble vector consisting of the copies from all the nodes as $\mathbf{X} \triangleq [\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_n^\top]^\top \in \mathbb{R}^{nd}$, and the dimension $l$ part of the ensemble as $\mathbf{X}_l = [x_{1,l} \ \cdots \ x_{n,l}]^\top \in \mathbb{R}^n$.

**Definition 3.1:** *The consensus plane $\mathcal{C}$ contains all the ensemble vectors such that the variables in all the nodes agree, i.e.*

$$\mathcal{C} \triangleq \{\mathbf{X} = [\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_n^\top]^\top \in \mathbb{R}^{dn} : \mathbf{x}_1 = \cdots = \mathbf{x}_n\}.$$

**Definition 3.2** (from [85])**:** *A projection $\mathcal{P}$ onto a set $\mathcal{C}$ is a function such that $\mathcal{P}(\mathcal{P}(\mathbf{x})) = \mathcal{P}(\mathbf{x})$ and $\mathcal{P}(\mathbf{x}) \in \{\mathbf{x} : \mathcal{P}(\mathbf{x}) = \mathbf{x}\}$ for all $\mathbf{x}$. A function $\mathcal{F}$ is called a projection subroutine of $\mathcal{P}$ if the following holds: (1) $\mathcal{F}$ is continuous; (2) $\lim_{n\to\infty} \mathcal{F}^n = \mathcal{P}$ uniformly; (3) $\mathcal{P}(\mathcal{F}(x)) = \mathcal{F}(\mathcal{P}(x)) = \mathcal{P}(x)$. We will also call this $\mathcal{F}$ a subprojection onto $\mathcal{C}$.*

## 3.2 A Stochastic Approximation Viewpoint

In this section, we review NEXT from the viewpoint of stochastic approximation. The main objective is to exemplify the way the "descent part" and the "consensus part" of consensus-based primal distributed optimization algorithms are woven together. We provide an alternative proof of the convergence of the Ineact NEXT algorithm given in Algorithm 2.1 using results from stochastic approximation. The notation for this section is exactly the same as in Chapter II. As we will demonstrate, NEXT can be seen as a two time-scale process, where $\mathbf{y}$ tracks the total gradient in the fast time-scale, and $\mathbf{x}$ tracks the fixed point iteration of $\hat{\mathbf{x}}(\bullet)$ (where $\hat{\mathbf{x}}_i(\mathbf{x}) = \tilde{\mathbf{x}}_i(\mathbf{x}, \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\mathbf{x}))$, also see (A.25)) with a subprojection onto the *consensus plane* in the slow time-scale.

Substituting the definitions of $\mathbf{z}$ and $\tilde{\pi}$ into Algorithm 2.1, we can rewrite each iteration of the algorithm in the following two steps:

$$\mathbf{y}_i[t] = \sum_{j=1}^{n} W_{ij}\mathbf{y}_j[t-1] + \Big[\nabla f_i(\mathbf{x}_i[t] - \nabla f_i(\mathbf{x}_i[t-1]))\Big], \tag{3.1}$$

$$\mathbf{x}_i[n+1] = \sum_{j=1}^{n} W_{ij}\Big[\mathbf{x}_j[t] + \alpha[t]\left(\tilde{\mathbf{x}}_j(\mathbf{x}_j[t], \mathbf{y}_j[t]) - \mathbf{x}_j[t] + \mathbf{e}_j[t]\right)\Big], \tag{3.2}$$

where $\|\mathbf{e}_i[t]\| \leq \epsilon_i[t]\ \forall\ i$, and $\tilde{\mathbf{x}}_i(\mathbf{x}_i[t], \mathbf{y}_i[t])$ is given by (2.4) with $\tilde{\pi}_i$ substituted by $\mathbf{y}_i$ using Line 11 of Algorithm 2.1. By letting $\mathbf{u}_i[t] = \mathbf{y}_i[t] - \nabla f_i(\mathbf{x}_i[t])$, (3.1) can be rewritten as

$$\begin{aligned}
\mathbf{u}_i[t] &= \sum_{j=1}^{n} W_{ij}\Big[\mathbf{u}_j[t-1] + \nabla f_j(\mathbf{x}_j[t-1])\Big] - \nabla f_i(\mathbf{x}_i[t-1]) \\
&= \mathbf{u}_i[t-1] + \beta[t]\Bigg\{\sum_{j=1}^{n} W_{ij}\Big[\mathbf{u}_j[t-1] + \nabla f_j(\mathbf{x}_j[t-1])\Big] \\
&\quad - \Big[\mathbf{u}_i[t-1] + \nabla f_i(\mathbf{x}_i[t-1])\Big]\Bigg\},
\end{aligned} \tag{3.3}$$

where $\beta[t] = 1$. It is evident that $\alpha[t] = o(\beta[t])$. As a result, (3.2) and (3.3) together form a two time-scale stochastic approximation algorithm [20], where $\mathbf{u}_i$ or $\mathbf{y}_i$ is on a faster, natural time-scale with constant step sizes, and $\mathbf{x}_i$ is updated on a slower, logarithmic time-scale with shrinking step sizes.

To analyze this process, we first begin with the fact that the fast variable $\mathbf{u}$ or $\mathbf{y}$ views the slow variable $\mathbf{x}$ as quasi-static, i.e. we can see $\mathbf{x}$ as constant in (3.3). Denote $\mathbf{u}$ as the ensemble of $\mathbf{u}_i$'s, i.e. $\mathbf{u} = \begin{bmatrix}\mathbf{u}_1^\top & \cdots & \mathbf{u}_I^\top\end{bmatrix}^\top$, $\nabla f$ as the ensemble of $\nabla f_i$'s, and similarly for $\mathbf{x}$, $\mathbf{y}$, etc. Then the iterate of $\mathbf{u}$ will asymptotically track the following ODE

$$\dot{\mathbf{u}}(t) = [\mathbf{W} \otimes \mathbf{I}_d - \mathbf{I}_{dn}][\mathbf{u}(t) + \nabla f(\mathbf{x})],\ \ \mathbf{u}(0) = 0, \tag{3.4}$$

where $\mathbf{I}$ is the identity matrix, $d$ denotes the dimension of $\mathbf{x}_i$'s, and $\otimes$ means the Kronecker product. Using the $\mathbf{y}$ variable, (3.4) can be written as

$$\dot{\mathbf{y}}(t) = (\mathbf{W} \otimes \mathbf{I}_d - \mathbf{I}_{dn})\mathbf{y}(t),\ \ \mathbf{y}(0) = \nabla f(\mathbf{x}). \tag{3.5}$$

**Lemma 3.3:** *We have* $\lim_{t\to\infty}\mathbf{y}(t) = \overline{\nabla f}(\mathbf{x}) \otimes \mathbf{1}_n$*, where* $\mathbf{1}$ *is the all one vector and* $\overline{\nabla f} = \frac{1}{n}\sum_{i=1}^n \nabla f_i$ *is the averaged gradient function.*

**Proof:** We have

$$\begin{aligned}
\lim_{t\to\infty}\mathbf{y}(t) &= \lim_{t\to\infty} e^{t(\mathbf{W}\otimes\mathbf{I}_d - \mathbf{I}_{dn})}\mathbf{y}(0) \\
&= \lim_{t\to\infty} e^{t\mathbf{W}\otimes\mathbf{I}_d} e^{-t\mathbf{I}_{dn}} \nabla f(\mathbf{x}) \\
&= \lim_{t\to\infty} (e^{t\mathbf{W}} \otimes \mathbf{I}_d) \cdot \frac{1}{e^t}\mathbf{I}_{dn}\nabla f(\mathbf{x}) \\
&= \lim_{t\to\infty} \left(\frac{e^t}{n}\mathbf{1}_n\mathbf{1}_n^\top\right) \otimes \mathbf{I}_d \cdot \frac{1}{e^t}\nabla f(\mathbf{x}) \\
&= \left[\frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top) \otimes \mathbf{I}_d\right] \cdot \nabla f(\mathbf{x}) = \overline{\nabla f}(\mathbf{x}) \otimes \mathbf{1}_n,
\end{aligned} \tag{3.6}$$

where the second equality follows from the two matrices being multiplication commutative, third from $e^{\mathbf{AI}+\mathbf{IB}} = e^{\mathbf{A}} \otimes e^{\mathbf{B}}$, and fourth from the fact that $\lim_{t\to\infty}\mathbf{W}^t = \frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top)$. ■

We see that $\mathbf{y}(t)$ indeed converges to the unique global asymptotically stable equilibrium, where all component of $\mathbf{y}$, i.e. $\mathbf{y}_i$'s, equal to the average of the gradients as desired.

Next, from the perspective of the slow variable $\mathbf{x}$, the fast variable $\mathbf{y}$ already reaches its equilibrium $\bar{\mathbf{y}}(\mathbf{x})$. That is to say, in (3.2) $\tilde{\mathbf{x}}_j(\mathbf{x}_j[t], \mathbf{y}_j[t])$ can be seen as $\tilde{\mathbf{x}}_j(\mathbf{x}_j[t], \overline{\nabla f}(\mathbf{x}_j[t]))$, which is exactly $\hat{\mathbf{x}}_j(\mathbf{x}_j[t])$, so that (3.2) become

$$\mathbf{x}_i[t+1] = \sum_{j=1}^n W_{ij}\Big[\mathbf{x}_j[t] + \alpha[t]\big(\hat{\mathbf{x}}_j(\mathbf{x}_j[t]) - \mathbf{x}_j[t] + \mathbf{e}_j[t]\big)\Big]. \tag{3.7}$$

As stated in [85], this recursive relation is again a two time-scale stochastic approximation in disguise, with a fast averaging process and a slow learning process. In fact, the averaging process is also on the natural time-scale as is $\mathbf{y}$. From [85] we know that the iterates of $\mathbf{x}$ will reach consensus $\mathbf{x}[t] = \begin{bmatrix}\mathbf{x}_c[t]^\top & \cdots & \mathbf{x}_c[t]^\top\end{bmatrix}^\top$, where each component $\mathbf{x}_c[t] \in \mathbb{R}^d$ tracks the ODE

$$\begin{aligned}\dot{\mathbf{x}}_c(t) &= \frac{1}{n}(\mathbf{1}_n \otimes \mathbf{I}_d)^\top \begin{bmatrix} \cdots & \hat{\mathbf{x}}_i(\mathbf{x}_c(t))^\top - \mathbf{x}_c(t)^\top & \cdots \end{bmatrix}^\top \\ &= \frac{1}{n}\sum_{i=1}^n \hat{\mathbf{x}}_i(\mathbf{x}_c(t)) - \mathbf{x}_c(t),\end{aligned} \tag{3.8}$$

as $\mathbf{1}_n/n$ is the unique stationary distribution resulted from the doubly stochastic transition matrix $\mathbf{W}$.

Note that with $\hat{\mathbf{x}}_i$'s being Lipschitz continuous ([80], Proposition 5(a)), this ODE is well-posed. We assume the differentiability of $G$ for now to avoid dealing with trickier non-differentiable Lyapunov functions here. We consider the whole objective itself as the Lyapunov function $V(\mathbf{x}_c) = U(\mathbf{x}_c) = F(\mathbf{x}_c) + G(\mathbf{x}_c)$. Then

$$\begin{aligned}\dot{V}(\mathbf{x}_c(t)) &= [\nabla U(\mathbf{x}_c(t))]^\top \cdot \left[\frac{1}{n}\sum_{i=1}^n \hat{\mathbf{x}}_i(\mathbf{x}_c(t)) - \mathbf{x}_c(t)\right] \\ &\le -c_\tau \frac{1}{n}\sum_{i=1}^n \|\hat{\mathbf{x}}_i(\mathbf{x}_c(t)) - \mathbf{x}_c(t)\|^2 \le 0,\end{aligned} \tag{3.9}$$

for some positive constant $c_\tau$. The first inequality is established similarly as [80], Proposition 5(b). By Lasalle's invariance principle, the iterates converge to the set of equilibria $\{\mathbf{x}_c : \frac{1}{n}\sum_{i=1}^n \hat{\mathbf{x}}_i(\mathbf{x}_c) = \mathbf{x}_c\}$ ([21], p. 57 and p. 118), which is the set of stationary points of the original optimization problem ([80], Proposition 2).

Remark that the conditions of applying [20] and [85] are either established in [80] or implied by the assumptions in the convergence of NEXT. Specifically, the boundedness of $\mathbf{x}$ follows from the recursive relation (3.2) and Proposition 9(a) in [80], $\sum_t \alpha[t] = \sum_t \beta[t] = \infty$, $\sum_t \alpha[t]^2 < \infty$, and $\sup \sum_t \alpha[t]\mathbf{e}_i[t] < \infty$ are just as assumed in NEXT convergence. Note that we do not need $\sum_t \beta[t]^2 < \infty$ as there is no noise in the recursion of $\mathbf{y}$. Also, we have deterministic convergence rather than almost sure convergence, since instead of being martingale differences, our noise term $\mathbf{e}_i[t]$ is actually deterministically bounded.

## 3.3 Nonlinear Consensus Schemes

As we seen from the previous section, from the perspective of stochastic approximation, the linear consensus update is just a special type of subprojection onto the consensus plane whereas the "consensus variable" $\mathbf{x}_c$ descends to the optimum. Enlightened from the result of [85], which broadens the types of subprojection that can be used, in this section we study nonlinear subprojection methods which transform variables into another domain where linear averages are taken, in particular the family of $(p, w)$-means. We first study the convergence behavior of nonlinear subprojections to the consensus plane in Section 3.3.1, then in Section 3.3.2 we describe the combinations of the nonlinear subprojections with NEXT and DGD.

### 3.3.1 Nonlinear Subprojection by Transformation

The Line 9 in Algorithm 2.1 can be seen as a type of subprojection to the consensus plane $\mathcal{C} := \{\mathbf{X} = [\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_n^\top]^\top \in \mathbb{R}^{dn} : \mathbf{x}_1 = \cdots = \mathbf{x}_n\}$. From the viewpoint of [85], a subprojection $\mathcal{F}(\cdot)$ is a function such that under infinitely many applications, the iterate will end up in $\mathcal{C}$ for any starting point $\mathbf{X}[0]$ (see Definition 3.2). In the distributed optimization setting, in each iteration, a node $i$ takes account of the information from its neighborhood $\{\mathbf{x}_j : j \in Nb(i)\}$ where $Nb(i) := N(i) \cup \{i\}$ and $N(i) := \{j : (i, j) \in \mathcal{E}\}$ and projects that to a "sub-consensus plane" $\mathcal{C}_i := \{\mathbf{X} = [\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_n^\top]^\top \in \mathbb{R}^{dn} : \mathbf{x}_j = \mathbf{x}_k \ \forall \ j, k \in Nb(i)\}$. The connectedness assumption of the communication graph $\mathcal{G}$ guarantees that the intersection of these sub-consensus planes $\bigcap_i \mathcal{C}_i$ is $\mathcal{C}$. In Algorithm 2.1, for node $i$ this is done by taking a convex combination of all its neighbors' information $\mathbf{X}_i = [\mathbf{x}_j]_{j \in Nb(i)}$, which can be thought of as the weighted least square solution, i.e. finding the point on $\mathcal{C}_i$ that minimizes the weighted sum of squares of $\ell_2$ distances from the point to each of the element in $\mathbf{X}_i$. Formally, the next iterate of node $i$ after the projection subroutine is

$$\mathbf{x}_i[t] = \sum_{j \in Nb(i)} W_{ij} \mathbf{x}_j[t] = \operatorname*{argmin}_{\mathbf{X} \in \mathcal{C}_i} \sum_{j \in Nb(i)} W_{ij} \|\mathbf{x}_j - \mathbf{x}_j[t]\|^2 = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^d} \sum_{j \in Nb(i)} W_{ij} \|\mathbf{x} - \mathbf{x}_j[t]\|^2.$$

We refer to this as the linear consensus scheme.

Here, we study nonlinear consensus schemes and their convergence behavior. For simplicity, in this subsection let us consider the one-dimension case ($d = 1$) and focus on repeated projection subroutines onto the consensus plane (instead of the distributed optimization context where the descent steps and consensus steps are interlaced together) first. In [121], it is proved that under certain "axioms of mean," the mean of $x_1, \cdots, x_n$ should take the form of $M(x_1, \cdots, x_n) = \varphi^{-1}\left(\frac{\varphi(x_1)+\cdots+\varphi(x_n)}{n}\right)$ where $\varphi$ is a continuous increasing function. Since we do not require the symmetry property (i.e., $M(x_1, \cdots, x_n)$ does not have to be a symmetric function), and want to include the weights $W_{ij}$ in the average expression, we consider

$$x_i[t+1] = \varphi^{-1}\left(\sum_j W_{ij}\varphi(x_j[t])\right), \tag{3.10}$$

with $\varphi$ being a strictly monotonic bijective function[1] on $\mathbb{R}$. Note that node $i$ can only use the information from its neighbors due to the communication constraint imposed by the network. If we start with $\mathbf{x}[0] = [x_1[0] \ \cdots \ x_n[0]]^\top$ and repeatedly apply (3.10), then this consensus scheme will converge to

$$x^* = \chi(\mathbf{x}[0]) = \varphi^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\varphi(x_i[0])\right). \tag{3.11}$$

We can rewrite each iteration in (3.10) as $z_i[t+1] \triangleq \varphi(x_i[t+1]) = \sum_j W_{ij}\varphi(x_j[t])$, which simply means taking convex combinations of $z_j \triangleq \varphi(x_j)$'s. That is to say, this is a linear consensus scheme in the *transformed domain* of $\varphi$. Since the infinite product of $\mathbf{W}$ converges to $\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, i.e. $\lim_{t\to\infty}\mathbf{W}^t = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, the iterations converge to $z^* = \frac{1}{n}\sum_{i=1}^{n} z_i[0]$ where $z_i[0] = \varphi(x_i[0])$ and from the invertibility of $\varphi$ we can find an $x^*$ such that $z^* = \varphi(x^*)$.

If we limit the domain of consideration to $\mathbb{R}^+$, a common choice of $\varphi$ is $\varphi(x) = x^p$:

$$x_i[t+1] = \left(\sum_j W_{ij}x_j^p[t]\right)^{1/p}, \tag{3.12}$$

which is called the $(p, w)$-mean in [71], or mean of order $p$ when $\mathbf{W}$ is uniform in [11]. The common arithmetic, geometric, and harmonic means fall into this category, corre-

---

[1] We do not require $\varphi$ to be increasing; otherwise, sometimes a minus sign is needed.

sponding to $p = 1$, $p \to 0$, and $p = -1$ when $\mathbf{W}$ is uniform, respectively. When $p \to \infty$ and $p \to -\infty$, (3.12) also corresponds to the max operation $x_i[t+1] = \max_{j \in Nb(i)} x_j[t]$ and the min operation $x_i[t+1] = \min_{j \in Nb(i)} x_j[t]$ in the support, respectively [71]. It is natural to ask whether this consensus scheme leads to faster projection onto $\mathcal{C}$, and whether it could increase the convergence speed of Algorithm 2.1. As we shall see, the answers to both are affirmative under certain situations.

The result of [90] implies that the "sample variance" in the transformed domain $\sum_{i=1}^n (z_i[t] - z^*)^2 = \sum_{i=1}^n [\varphi(x_i[t]) - \varphi(z^*)]^2$ works as a Lyapunov function that decreases over time. Due to the monotonicity of $\varphi$, we know that $V[t] = \sum_{i=1}^n (x_i[t] - x^*)^2$ also monotonically decreases over time. A reasonable evaluation of a consensus scheme is how many iterations it takes for this $V[t]$ to decrease from its initial value to a fraction of $\epsilon$, i.e. $T_\epsilon$ taken from the infimum of the set satisfying the condition

$$T_\epsilon \triangleq \inf \left\{ t : \frac{V[t]}{V[0]} = \frac{\sum_i (x_i[t] - x^*)^2}{\sum_i (x_i[0] - x^*)^2} \le \epsilon \right\}. \tag{3.13}$$

Notice that the max operation only need $d(\mathcal{G})$ iterations to project any point onto $\mathcal{C}$, where $d(\mathcal{G})$ is the diameter of $\mathcal{G}$; indeed, the max consensus scheme is equivalent to the node with the maximum value broadcasting its information, and within $d(\mathcal{G})$ number of iterations it will reach any node in the network, and decrease $V$ to 0. On the contrary, for linear consensus scheme $\frac{V[t]}{V[0]}$ is proportional to $\|\mathbf{W}^t - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\|_F^2$ which decreases geometrically, and it takes infinitely many iterations for $V$ to go down to 0. Thus we have the following.

**Fact 3.4:** *Denote $T_{\epsilon,p}$ as the minimal number of iterations required for the sample variance to reduce to a fraction of $\epsilon$ using the $p$-mean consensus scheme in* (3.12). *Then given any* $\mathbf{x}[0]$, *there exists small enough* $\epsilon > 0$ *and large enough* $p > 1$ *such that* $T_{\epsilon,1} > T_{\epsilon,p}$.

Given any consensus scheme, achieving a smaller *tolerance* $\epsilon$ requires a larger $T_\epsilon$. For a fixed tolerance $\epsilon$, a faster consensus scheme takes smaller $T$; and given $T$, a faster consensus scheme achieves smaller $\epsilon$. The above fact states that for smaller tolerance requirement or larger number of iterations running, there exists a $p$-mean scheme converging faster than linear consensus. This is, however, not necessarily true when $T$ is not large enough, as given in the following counterexample.

**Example 3.5:** *Consider a ring of five nodes* $\mathcal{V} = \{1, 2, 3, 4, 5\}$ *and* $\mathcal{E} = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)\}$ *with initial values* $\mathbf{x}[0] = \begin{bmatrix} 7 & 2 & 12 & 2 & 7 \end{bmatrix}^\top$ *and a doubly stochastic matrix*

$$\mathbf{W} = \frac{1}{5} \begin{bmatrix} 3 & 1 & 0 & 0 & 1 \\ 1 & 3 & 1 & 0 & 0 \\ 0 & 1 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 1 & 0 & 0 & 1 & 3 \end{bmatrix}.$$

*Then after one iteration, linear consensus applies* $\mathbf{W}$ *on* $\mathbf{x}[0]$ *and gives* $\begin{bmatrix} 6 & 5 & 8 & 5 & 6 \end{bmatrix}^\top$*, while the max operation yields* $\begin{bmatrix} 7 & 12 & 12 & 12 & 7 \end{bmatrix}^\top$*. The ratio of sample variances of linear scheme* $\frac{6}{70}$ *is smaller than that of max operation, which is* $\frac{50}{250}$*.*

However, we find from simulations that the ratio of sample variances given any $T$ monotonically decreases with $p$ ($p \geq 1$) when $\mathbf{x}[0]$ is uniformly distributed. This along with Fact 3.4 suggests that although there are counterexamples, $p$-mean with a larger $p$ usually leads to faster projection onto $\mathcal{C}$.

### 3.3.2 Distributed Optimization with Nonlinear Transformation

As we saw earlier, many primal distributed optimization algorithms, such as NEXT and DGD, interleave the “consensus steps” at a faster time-scale with the ”descent steps” at a slower time-scale, where the consensus steps are subprojections onto the consensus plane $\mathcal{C}$. From Section 3.3.1, we know that the nonlinear subprojection by transformation is also a subprojection onto $\mathcal{C}$. Thus, it is of interest to study the convergence behavior of algorithms that combine the nonlinear subprojection with the original “descent part” of a primal distributed optimization algorithm.

#### 3.3.2.1 NEXT with Nonlinear Transformation

We consider taking element-wise transformations, where in each element the transformation function is strictly monotonic and bi-Lipschitz (and hence bi-continuous).

The strict monotone property is to ensure the inverse transformation always exists.

**Definition 3.6:** *Denote* $\varphi_{i,t} : \mathcal{K} \to \mathcal{K}$ *such that* $\varphi_{i,t}(\mathbf{x}) = \begin{bmatrix} \varphi_{i,t,1}(x_1) & \cdots & \varphi_{i,t,d}(x_d) \end{bmatrix}^\top$ *where* $\mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_d \end{bmatrix}^\top$ *and* $\{\varphi_{i,t,l}\}_{l=1}^d$ *are all one-dimensional functions. Note the three indices in the subscript of* $\varphi_{i,t,l}$ *are the agent index, the time index, and the dimension index, respectively. Also, for any transformation* $\varphi : \mathcal{K} \to \mathcal{K}$, *define* $\varphi(S) := \{\varphi(\mathbf{x}) : \mathbf{x} \in S\}$.

In the following revised version of NEXT algorithm, in each time step and for each agent we take an element-wise transformation of the intermediate result from local optimization, apply the doubly-stochastic matrix, and take the inverse element-wise transformation. It is easy to see that the combination is a subprojection onto the consensus plane. Note that for different time steps and for different agents we could take different transformations, so that they could be chosen in an online manner depending on all the past information, giving us more flexibility.

```
Algorithm 3.1 Inexact NEXT with Nonlinear Transformation
In Line 9 of Algorithm 2.1: $\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^n W_{ij}\varphi_{i,t}(\mathbf{z}_j[t])\right)$
```

**Theorem 3.7:** *Let* $\{\mathbf{x}[t]\}_t \triangleq \{(\mathbf{x}_i[t])_{i=1}^n\}_t$ *be the sequence generated by Algorithm 3.1. Assume* $\varphi_{i,t,l}$ *and* $\varphi_{i,t,l}^{-1}$ *are strictly monotonic Lipschitz continuous functions with constants* $L_+$ *and* $L_-$, *respectively, for all* $i \in [n]$, $t \in \{0\} \cup \mathbb{N}$ *and* $l \in [d]$. *Under the same assumptions as the convergence of Algorithm 2.1 (Assumption A and Assumption F in Chapter II), all sequences* $\{\mathbf{x}_i[t]\}_t$ *asymptotically agree, and their limit points are stationary points of the original problem.*

**Proof:** See Appendix B. ■

Note that in Algorithm 3.1, node $j$ still sends $\mathbf{z}_j[t]$ to node $i$; the transformation from $\mathbf{z}_j[t]$ to $\varphi_{i,t}(\mathbf{z}_j[t])$ is taken at node $i$.

#### 3.3.2.2 DGD with Nonlinear Transformation

The nonlinear transformation idea can work with many primal algorithms, and here we combine it with DGD (Algorithm 2.3) as another example. Here we also provide a convergence rate analysis.

---

**Algorithm 3.2** DGD with Nonlinear Transformation

---

```
In Line 4 of Algorithm 2.3: $\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_{i,t}(\mathbf{x}_j[t] - \alpha[t]\nabla f_j(\mathbf{x}_j[t])\right)$
```

---

**Theorem 3.8:** *Assume $\varphi_{i,t,l}$ and $\varphi_{i,t,l}^{-1}$ are strictly monotonic Lipschitz continuous functions and have uniformly bounded first derivatives and second derivatives*[2]*, all with constants $L_+$ and $L_-$, respectively, for all $i \in [n]$, $t \in \{0\} \cup \mathbb{N}$ and $l \in [d]$. Also for all $i \in [n]$, assume the objective functions $f_i$ is strongly convex with constant $\nu$, has gradient uniformly bounded by $B$ (Assumption (A4) from Section 2.2.1), and has Lipschitz continuous gradient with constant $L_f$ (Assumption (A5) from Section 2.2.1 with $L_f = \max_i L_i$), which implies the existence of a unique optimal solution $\mathbf{x}^*$. Suppose we choose a constant learning rate satisfying $\alpha[t] = \alpha < \frac{1}{L_f}$. Then the sequence generated by Algorithm 3.2 $\{\mathbf{x}[t]\}_t \triangleq \{(\mathbf{x}_i[t])_{i=1}^n\}_t$ satisfies*

$$\frac{n\nu}{2}\|\bar{\mathbf{x}}[t] - \mathbf{x}^*\|^2 \le F(\bar{\mathbf{x}}[t]) - F(\mathbf{x}^*) \le \bar{\rho}^{t-1}\big[F(\bar{\mathbf{x}}[0]) - F(\mathbf{x}^*)\big] + O(\alpha^2), \tag{3.14}$$

*where the factor $\bar{\rho} \triangleq 1 - \alpha\nu$ is in $[0, 1)$. In other words, $\bar{\mathbf{x}}[t]$ converges to an $O(\alpha)$ neighborhood of $\mathbf{x}^*$ exponentially fast, while the optimality gap in the mean $F(\bar{\mathbf{x}}[t]) - F(\mathbf{x}^*)$ also decreases exponentially until it settles to a value that is $O(\alpha^2)$. For the agreement from the nodes, for all $i \in [n]$, the deviation from the mean $\|\mathbf{x}_i[t] - \bar{\mathbf{x}}[t]\|$ yet again decreases exponentially until reaching $O(\alpha)$ (see Lemma B.3 for the detailed description).*

**Proof:** See Appendix B. ■

In Theorem 3.8 and its proof, we see that the convergence rate is in line with the results in the literature [132, 13]; taking nonlinear transformations do not affect the factor $\bar{\rho}$ in the geometric convergence, but only cause larger constants in the expressions of the $O(\alpha)$ and $O(\alpha^2)$ neighborhoods. So to speak, the nonlinear consensus scheme does not improve the theoretical guarantee of the convergence rate. However, we do see improvements from using various nonlinear consensus schemes in Section 3.6. Some of them arise from the detailed relative relations for the variables exemplified in Section 3.5.2 that are hard to be captured by analysis.

---

[2] We actually only need the existence of the second derivatives of $\varphi_{i,t,l}$ and $\varphi_{i,t,l}^{-1}$; the boundedness of $\mathbf{x}_i[t]$ will then ensure the boundedness of the derivatives in the region of interest. We would however like to use the same constants for simplicity of notation.

## 3.4 Relaxation of Column Stochasticity for Consensus

Many primal distributed optimization algorithms require double stochastic matrices for averaging. In contrast, in this subsection we study relaxing this need of column stochasticity. Recall from Section 3.2 that the NEXT algorithm (as well as many primal algorithms that use the averaging consensus scheme) is essentially performing a two time-scale stochastic approximation [85], where the algorithm interlaces the descent steps and the consensus steps. In the fast process consisting of the consensus steps, the decision variables maintained in the nodes asymptotically agree as the ensemble vector $\mathbf{X} = [\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_n^\top]^\top$ converges to the consensus plane $\mathcal{C}$, which is referred to as "consensus convergence." On the other hand, in the slow process consisting of the descent steps, the consensus vector $\mathbf{x}_c$ (which is suitably taken as the average vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$) converges to one of the local minima, which we call "aggregate convergence." Row stochasticity is required for both these convergences, as it ensures the new iterate stays inside the convex hull spanned by the old iterates during averaging; otherwise, the new iterate could fall outside the feasible region, let alone guaranteeing convergence. On the other hand, column stochasticity is to ensure that the objectives of the nodes are weighed in equally, since the overall objective is the sum of them (and hence an equivalent overall objective is the uniformly weighted average of them); hence, it is necessary for aggregate convergence but not consensus convergence. Here, we split out the two convergences, and relax the column stochasticity requirement for the consensus convergence which leads to more general and flexible consensus schemes.

### 3.4.1 NEXT with Column Stochasticity of Consensus Relaxed

We start with the definition of a "shrunk convex hull."

**Definition 3.9:** *Denote co$(T)$ to be the convex hull of a finite set $T$. Further denote $m_S$ to be the centroid of a convex set $S$, and $\delta \circ S := \{(1-\delta)m_S + \delta x : x \in S\}$, i.e. the set obtained by shrinking $S$ to its centroid by a factor of $\delta \in [0, 1]$. The shrunk convex hull of $T$ with the factor $\delta$ is then defined as $\delta \circ$ co$(T)$.*

The inexact NEXT algorithm can be revised by choosing any point inside the shrunk convex hull spanned by the neighboring local optimization results with a factor $\delta$ smaller than 1, as given in the following algorithm.

```
Algorithm 3.3 Inexact NEXT with Consensus in Convex Hull
In Line 9 of Algorithm 2.1: pick any $\mathbf{x}_i[n+1] \in \delta \circ co(\{\mathbf{z}_j[t] : j \in Nb(i)\})$
```

**Theorem 3.10:** *Let* $\{\mathbf{x}[t]\}_t \triangleq \{(\mathbf{x}_i[t])_{i=1}^n\}_t$ *be the sequence generated by Algorithm 3.3. Assume* $\delta \in (0,1)$. *Under Assumption A and Assumption F, all sequences* $\{\mathbf{x}_i[t]\}_t$ *asymptotically agree, and their limit points are stationary points of the original problem.*

**Proof:** See Appendix B. ■

In short, the above theorem says that the NEXT algorithm can work well with $\mathbf{x}$ and $\mathbf{y}$ using different consensus matrices, and the consensus matrices for $\mathbf{x}$ do not have to be column stochastic. The column stochasticity is, however, crucial for the consensus matrices for $\mathbf{y}$, since it ensures $\mathbf{y}$ tracking the uniformly weighted average of $\nabla f_i$'s, and hence $\mathbf{x}$ converging to the correct optima of $\sum_i f_i$ instead of a non-uniformly weighted version.

### 3.4.2 DGD with Gradient Tracking and Column Stochasticity of Consensus Relaxed

In the DGD algorithm, on the other hand, the gradient information is gossiped through the averaging of $\mathbf{x}$. Therefore, the relaxation of column stochasticity of the averaging matrices could lead to convergence to the optima of a non-uniformly weighted version of the objective functions $f_i$'s. One approach that allows DGD to enjoy the flexibility of column stochasticity relaxation is through tracking the gradient information with another variable $\mathbf{y}$ [94] given in Algorithm 3.4, which is still averaged by doubly stochastic matrices, while $\mathbf{x}$ is averaged by possibly non-column stochastic matrices.

To ensure convergence to correct optima of $\sum_i f_i$, the matrix $\mathbf{W}^y$ has to be doubly stochastic while the matrix $\mathbf{W}^x$ only has to be row stochastic. The detailed convergence analysis of this algorithm is left as future work.

**Algorithm 3.4** DGD with Gradient Tracking

```
Initialization: $\mathbf{x}_i[0] \in \mathbb{R}^d$, $\mathbf{y}_i[0] = \nabla f_i(\mathbf{x}_i[0])$, $t = 0$.
while $\mathbf{x}[t]$ does not satisfy the termination criterion do
    $t \leftarrow t + 1$
    $\mathbf{x}_i[t+1] = \sum_j W^x_{ij}\mathbf{x}_j[t] - \alpha[t]\mathbf{y}_i[t]$
    $\mathbf{y}_i[t+1] = \sum_j W^y_{ij}\mathbf{y}_j[t] + \nabla f_i(\mathbf{x}_i[t+1]) - \nabla f_i(\mathbf{x}_i[t])$
Output: $\mathbf{x}[t]$
```

## 3.5 Combining Nonlinear Transformation and Column Stochaticity Relaxation

Recall that in Section 3.3.1 we argued that the max and min operations are generally faster than linear consensus schemes – it only takes the number of steps equal to the diameter of the communication graph for the operations to reach exact consensus. A natural question then is how will primal distributed optimization algorithms perform when the consensus is reached through the same operations, and can we establish convergence guarantees for such schemes. For the formal question, in some numerical examples depending on the initial value of $\mathbf{x}[0]$ given in Section 3.6, we do find that algorithms with max and min operations used for consensus converge faster than those with linear consensus schemes. We are hence motivated to study the latter question by investigating the nonlinear transformations of the shrunk convex hull, a combination of the ideas from Section 3.3.2 and Section 3.4, with NEXT algorithm as the example.

**Algorithm 3.5** Inexact NEXT with Nonlinear Transformation and Column Stochaticity Relaxation

```
In Line 9 of Algorithm 2.1: pick any $\mathbf{x}_i[t+1] \in \varphi^{-1}_{i,t}(\delta \circ co(\{\varphi_{i,t}(\mathbf{z}_j[t]) : j \in Nb(i)\}))$
```

**Theorem 3.11:** *Let* $\{\mathbf{x}[t]\}_t \triangleq \{(\mathbf{x}_i[t])_{i=1}^n\}_t$ *be the sequence generated by Algorithm 3.5. Assume* $\varphi_{i,t,l}$ *and* $\varphi^{-1}_{i,t,l}$ *are strictly monotonic Lipschitz continuous functions with constants* $L_+$ *and* $L_-$*, respectively, for all* $i \in [n]$*,* $t \in \{0\} \cup \mathbb{N}$ *and* $l \in [d]$*. Also assume* $\delta \in (0, 1)$*, Assumption A, and Assumption F. Then all sequences* $\{\mathbf{x}_i[t]\}_t$ *asymptotically agree, and their limit points are stationary points of the original problem.*

**Proof:** The result follows by combining the proofs of Theorem 3.7 and Theorem 3.11. In fact, the proof of Theorem 3.7 essentially applies here too since we do not require the $\mathbf{W}$ matrices to be column stochastic nor time-invariant in the proof. The only requirements regarding $\mathbf{W}$ are the assumptions in Section 2.2.1 ($W_{ij} > \vartheta$ for $(i, j) \in \mathcal{E}$ for some $\vartheta > 0$ or $i = j$, and $W_{ij} = 0$ otherwise), which is ensured by the positive shrinking factor $\delta$. ■

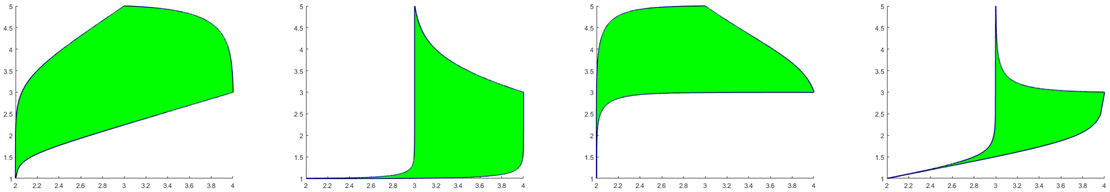
### 3.5.1 Enlarging to Cube Hull with $p$-means Transformations

One reason that the max operation sometimes outperforms linear consensus is it has larger "step size" in the consensus step than the linear consensus schemes. Indeed, it is common that the outcome of the max operation falls outside the convex hull in Section 3.4 spanned by the neighboring iterates. In general, the convex hull is the best one could hope for, since performing the max operation can lead to local variables falling outside $\mathcal{K}$; however, with additional constraint on $\mathcal{K}$, e.g. $\mathcal{K} = \mathbb{R}^d_+$, we can further expand the "hull of consensus choices" to the *cube hull* defined below using the family of $p$-means transformations to include the max and min operations. The main idea is to take the union of all the convex hulls transformed by any transformation inside the family. The theory developed here serves as an application and hence a special case of Theorem 3.11.

**Definition 3.12:** *The cube hull of a finite set $T \subset \mathbb{R}^d$, denoted by $cb(T)$, is defined as the smallest $d$-dimensional cube that contains $T$ with all edges parallel with the Cartesian axes. That is, if we write $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, then $cb(T) := \{\mathbf{y} : \min_i x_{i,l} \leq y_l \leq \max_i x_{i,l} \;\forall\; i = 1, \cdots, n$ and $l = 1, \cdots, d\}$.*

Note that the $\varphi_{i,t}$, the transformation at time $t$ for node $i$, can be an arbitrary transformation that satisfies the assumptions of our theorems, so it can actually be chosen at time $t$ dependent on all the past information. Suppose we have a family of transformations $\Phi$. Algorithm 3.5 then implies that we can choose $\mathbf{x}_i[t+1]$ from an even bigger set $\bigcup_{\varphi \in \Phi} \varphi^{-1}(\delta \circ co(\{\varphi(\mathbf{z}_j[t]) : j \in Nb(i)\}))$. Let us take $\Phi$ to be the family of power of $p$ functions $\{\hat{\varphi}_p : p \in \mathbb{R} \setminus \{0\}\}$ where $\hat{\varphi}_p(x) = x^p$ with set $\mathcal{K} = \mathbb{R}^d_+$.

**Fact 3.13:** *Given any finite set $T \in \mathbb{R}^d_+$ and $\delta' \in (0, 1)$, there exist a $\bar{p} < \infty$ and $\delta \in (0, 1)$ such that $\delta' \circ cb(T) \subseteq \bigcup_{\varphi \in \Phi_{\bar{p}}} \varphi^{-1}(\delta \circ co(\{\varphi(T)\}))$, where $\Phi_{\bar{p}}$ is defined as the family of power functions $\{\hat{\varphi}_p : p \in [-\bar{p}, \bar{p}] \setminus \{0\}\}$.*



(a) $p_1 = 10, p_2 = 10$ (b) $p_1 = 10, p_2 = -10$ (c) $p_1 = -10, p_2 = 10$ (d) $p_1 = -10, p_2 = -10$

Figure 3.1: An example of transformed convex hull with two-dimensional element-wise $p$-power functions $\varphi_{p_1,p_2}(\mathbf{x}) = [\hat{\varphi}_{p_1}(x_1)\ \ \hat{\varphi}_{p_2}(x_2)]^\top$.

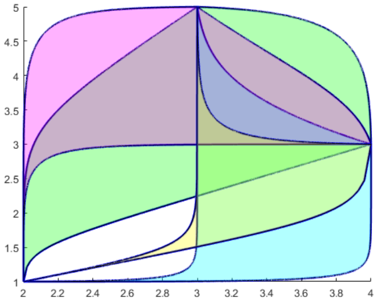


Figure 3.2: The four regions in Figure 3.1 plotted in different colors in one graph.

Figure 3.1 gives an example of "transformed convex hull" $\varphi^{-1}(co(\{\varphi(T)\}))$ using the family of power of $p$ functions in a two-dimensional setting. In the example, we let $T = \{[2\ \ 1]^\top, [3\ \ 5]^\top, [4\ \ 3]^\top\}$, and $\varphi_{p_1,p_2}(\mathbf{x}) = [\hat{\varphi}_{p_1}(x_1)\ \ \hat{\varphi}_{p_2}(x_2)]^\top$. The four sub-figures depict the region of the transformed hull with four different values of $(p_1, p_2)$. To get a better sense of their union, we plot them together in Figure 3.2. From the figure, we can indeed see that $0.8 \circ cb(T)$ is contained in the union of the transformed hulls with $\varphi$ coming from $\{\varphi_{p_1,p_2} : p_1, p_2 \in [-10, 10] \setminus \{0\}\}$. Note that the blank region in the bottom left corner in Figure 3.2 can be filled by slowly changing $(p_1, p_2)$ from $(1, 1)$ to $(10, -10)$.

By Fact 3.13, the convergence of Algorithm 3.5, and the fact that all functions in $\Phi_{\bar{p}}$ as well as their inverses are Lipschitz continuous, we can choose any $\mathbf{x}_i[t+1]$ from the "shrunk cube hull" supported by neighbors' variables, given in Algorithm 3.6. Its convergence guarantee is then ensured by Theorem 3.11.

---

**Algorithm 3.6** Inexact NEXT with Consensus in Cube Hull

---

In Line 9 of Algorithm 2.1: pick any $\mathbf{x}_i[n+1] \in \delta \circ cb(\{\mathbf{z}_j[t] : j \in Nb(i)\})$

---

### 3.5.2 Application: Gradient-Oriented Consensus Schemes

In the numerical simulations presented in Section 3.6, max and min operations can converge either faster or slower than the linear consensus scheme depending on the initial value of $\mathbf{x}_0$. When the initial value is generally component-wise smaller (resp. larger) than the optimal value, max (resp. min) works better and min (resp. max) works poorly. When the initial value is around the same range as the optimal value, with some coordinates larger and some smaller, then usually linear consensus is better than max and min versions of nonlinear consensus. Since max and min consensus schemes are generally faster subprojections onto the consensus plane, the described phenomena are better explained by the aggregate convergence part. Specifically, the aggregate convergence basically involves descending in directions *similar* to the negative gradient. When the initial value is smaller (resp. larger) than the optimal value, the direction of change of taking max (resp. min) operation is more similar to the negative gradient, and thus it further reduces the objective function while linear scheme is only taking average. This gives rise to the idea of the "*gradient-oriented consensus scheme*," where we try to align the consensus steps with the negative gradients within the hull of consensus choices.

We start by considering the hull of consensus choices being the convex hull spanned by the neighboring iterates as in Algorithm 3.3. Note that we would like to choose a point such that the direction of change is similar to the negative total gradient in order to get faster aggregate convergence. In the distributed setting, a node does not have the information of objectives of other nodes and hence does not know total gradient. Fortunately, NEXT uses the $\mathbf{y}$ variable to track the average gradient, which we take advantage of in Algorithm 3.7.

---

**Algorithm 3.7** Inexact NEXT with Gradient-oriented Consensus in Convex Hull

---

In Line 9 of Algorithm 2.1: $\mathbf{x}_i[t+1] \in \operatorname{argmax}_{\mathbf{x}\in\delta\circ co(\{\mathbf{z}_j[t]:j\in Nb(i)\})} \frac{\mathbf{y}_i[t]^\top(\mathbf{x}-\mathbf{z}_i[t])}{\|\mathbf{x}-\mathbf{z}_i[t]\|}$

---

Basically, Algorithm 3.7 is doing angle minimization within the convex hull supported by the variables from the neighbors. It is a special case of Algorithm 3.3 and uses the gradient information. The idea of gradient angle minimization was proposed for gradient descent methods in constrained settings [139]. One could also consider projecting the gradient or optimizing any other objective function using gradient information within the constraint set $\delta \circ co(\{\mathbf{z}_j[t] : j \in Nb(i)\})$.

As we combine the convergence results of nonlinear transformation and convex hull in Section 3.5 and enlarge the hull of consensus choices to the shrunk cube hull, we can adopt the same gradient-oriented idea but with the new iterate lying in the constraint set $\delta \circ cb(\{\mathbf{z}_j[t] : j \in Nb(i)\})$ as in Algorithm 3.8.

---

**Algorithm 3.8** Inexact NEXT with Gradient-oriented Consensus in Cube Hull

```
In Line 9 of Algorithm 2.1: $\mathbf{x}_i[t+1] \in \operatorname{argmax}_{\mathbf{x} \in \delta \circ cb(\{\mathbf{z}_j[t] : j \in Nb(i)\})} \frac{\mathbf{y}_i[t]^\top (\mathbf{x} - \mathbf{z}_i[t])}{\|\mathbf{x} - \mathbf{z}_i[t]\|}$
```

Just as Algorithm 3.7 is a special case of Algorithm 3.3, Algorithm 3.8 is a special case of Algorithm 3.6 (which is in turn a special case of Algorithm 3.5). One could optimize any other objective, and choose any other family of transformations, and the convergence of the algorithms will be gauranteed by Theorem 3.10 and Theorem 3.11. We remark that cube hull is the largest possible set one could get, as the notion of mean often requires that the average lies between min and max [11]. Numerical simulations of these methods will be given in Section 3.6. Finally, our conjecture is the gradient-oriented idea within shrunk cube hull or shrunk convex hull also works for DGD with gradient tracking as explained in Section 3.4.2; the convergence guarantee and performance evaluation for this algorithm are left as future work.

## 3.6 Simulation Results

We assume the underlying graph is the 19 cell wrap-around implementation [53], which is a classic example widely used in simulations for cellular networks. Specifically, each BS in the implementation is a node in our graph, and an edge exists between a pair of nodes if and only if the corresponding to BSs are neighbors. The graph contains 19 nodes, and is symmetric and regular of degree 6. An illustration figure from [53] is provided in Figure 3.4.

We consider the objective functions having a partial dependency structure. This is one scenario when the ordinary linear consensus might not work well. We assume that there are 19 local networks, and for any $i$ we have $\mathcal{N}_i = Nb(i)$, i.e. the neighbors of $i$ and $i$ itself. This can be thought of as each node $i$ having a local variable $\mathbf{x}^i$, and $f_i$ only depending on its neighbors' as well as its own local variables.

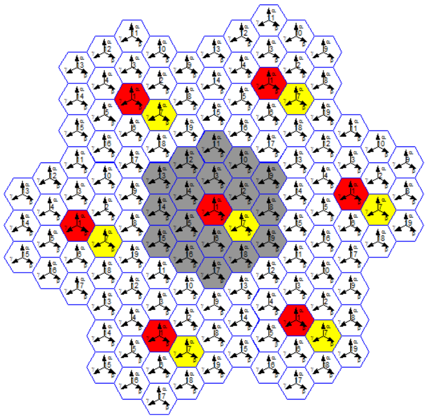


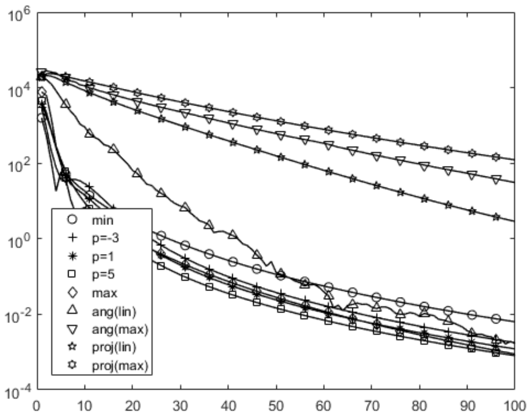
Figure 3.3: Network setting.

We assume for all $i$, the local variable $\mathbf{x}^i \in \mathbb{R}_+^{d'}$ where $d' = 2$. The whole tuple variable $\mathbf{x}$ lies in $\mathbb{R}_+^{38}$. We use the convention that $\mathbf{x}^{Nb(i)}$ denotes the ensemble vector with elements in $\{\mathbf{x}^j : j \in Nb(i)\}$. For all $i$, $f_i$ is then given as

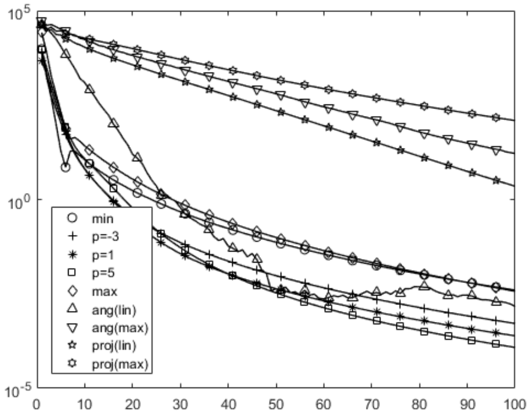
$$f_i(\mathbf{x}^{Nb(i)}) = \frac{1}{2}\mathbf{x}^{Nb(i)^\top}\mathbf{M}^\top\mathbf{M}\mathbf{x}^{Nb(i)} + \mathbf{b}^\top\mathbf{x}^{Nb(i)}, \tag{3.15}$$

where $\mathbf{M}$ is a $14 \times 14$ matrix with *i.i.d.* elements from Unif$[-1, 1]$, and $\mathbf{b}$ is a $14 \times 1$ vector with *i.i.d.* elements from Unif$[-150, -50]$. We consider a convex quadratic objective, since less can be said for the convergence rate for non-convex objectives, and this is the simplest convex objective one can consider. The negative choices of $\mathbf{b}$ is to make the optimal solution on $\mathbb{R}^d$ typically lie in $\mathbb{R}_+^d$. The mean of the optimal solution for each coordinate is around 25. The initial choice of $\mathbf{x}[0]$ for every node is drawn *i.i.d.* for all coordinates from $\chi^2(k)$ (chi-squared distribution with parameter $k$), where we choose $k = 5$, $k = 25$, and $k = 100$ for our three cases, corresponding to low, median, and high initial values relative to the optimal value. The surrogate function of $f_i$ is formed by direct linearization plus a quadratic regularization term with coefficient $\tau_i$ to be specified later on. We let the regularization function $G$ be 0. The value of other parameters are: $\alpha_0 = 0.8$, $\beta = 0.53$, $\tau_i = 100$ for all $i$, and $\delta = 0.9$. Recall that $\delta$ is the "shrinking factor" given in Definition 3.9 and Theorem 3.11.

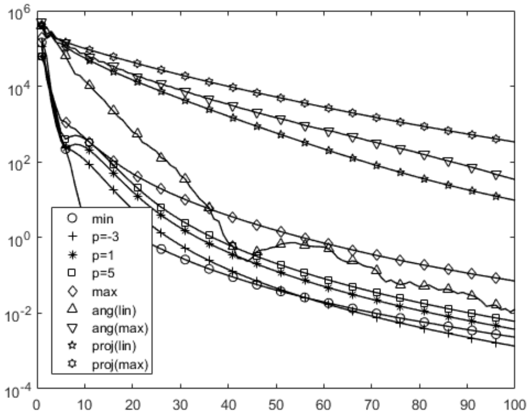
Denote $\mathbf{x}_j^i$ to be the variable part of $\mathbf{x}^i$ stored at node $j$, and $\mathbf{x}_a := [\cdots\ \ \mathbf{x}_i^{i\top}\ \ \cdots]^\top$. Note that the unique optimal value $F^* = F(\mathbf{x}^*)$ achieved at $\mathbf{x}^*$ can be easily solved by QP. Figure 3.4 illustrates how $\frac{|F(\mathbf{x}_a)-F^*|}{|F^*|}$ decreases as a function of number of iterations, and Figure 3.5 shows the variation of $\sum_{i=1}^n \|\mathbf{x}_i^i - \mathbf{x}^*\|^2$. The plotted consensus



(a) $k = 5$

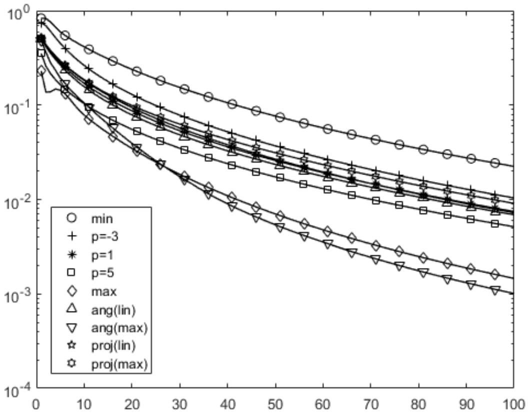


(b) $k = 25$

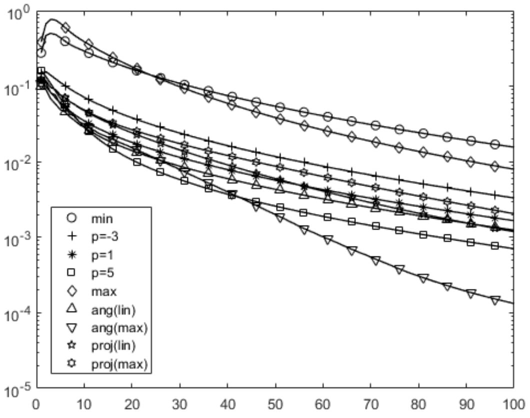


(c) $k = 100$

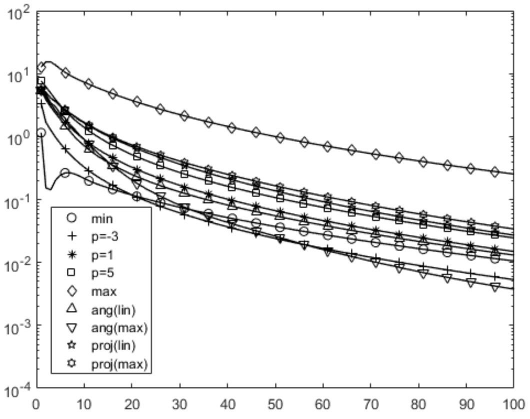
Figure 3.4: The convergence rates of the objective value with respect to number of iterations for different consensus methods for randomly generated quadratic objective functions.



(a) $k = 5$



(b) $k = 25$



(c) $k = 100$

Figure 3.5: The convergence rates of the total deviation from the optimum with respect to number of iterations for different consensus methods for randomly generated quadratic objective functions.

schemes include the following: (1) the component-wise $(p, w)$-mean scheme, which is essentially Algorithm 3.1 with $\varphi_{i,t,l} = \hat{\varphi}_p$ for all $i, t, l$ and taking linear average with weights $\mathbf{W}$ given by

$$W_{ij} = \begin{cases} 0 & \text{if } j \notin Nb(i) \\ \frac{1}{\bar{d}} & \text{if } j \in Nb(i) \text{ and } i \neq j \\ \frac{\bar{d}-d_i}{\bar{d}} & \text{if } j \in Nb(i) \text{ and } i = j \end{cases},$$

where $\bar{d} = \max_i d_i + 1$, (2) component-wise max and min operations, which are just $(p, w)$-mean and taking $p \to \infty$ and $p \to -\infty$[3], (3) angle minimization within shrunk convex hull given in Algorithm 3.7, and (4) angle minimization within shrunk cube hull given in Algorithm 3.8.

Generally, when the initial value is component-wise lower than optimal value ($k = 5$), the optimal $p$ for fastest convergence is large; when the initial value is component-wise higher than optimal value ($k = 100$), the optimal $p$ is small; when the initial value is around optimal value ($k = 25$), the optimal $p$ is around 1. Let us focus on the $(p, w)$-mean first. Figure 3.4 shows that when $k = 5$, choosing $p = 5$ makes the convergence to the optimum fastest while max operation is the second best and min is the worst; when $k = 25$, $p = 5$ is still the fastest but $p = 1$ is the second best, whereas max and min are the worst; when $k = 100$, $p = -3$, min, and max are the best, second best, and worst. The gradient-oriented consensus in convex hull method decreases slower at first, but later catches up and lies somewhere between the group of $(p, w)$-mean schemes in all cases. In contrast, the gradient-oriented consensus within cube hull method is the poorest, having a big gap from the group. Note that the convergence curves of the gradient-oriented methods can be adjusted by $\delta$ – the smaller $\delta$ is, the closer the curves are to the linear method ($p = 1$). The reason that min does not work well in Figure 3.4 (a) is as follows. At each iteration, at node $i$ the component $\mathbf{x}_i^j$ for $j \notin Nb(i)$ is dragged upward by its neighbors towards the optimum, and the min operation somehow drags the value back. Therefore, even if min is the fastest subprojection similar to max, it does not work well in this scenario.

---

[3] Note that our theory developed in Section 3.5 does not guarantee the convergences of max and min operations but a shrunk version of them.

For the same reason max does not work well in Figure 3.4 (c).

In Figure 3.5 and for the $(p, w)$-mean family, max is the best and min is the worst for low initial value; $p = 5$ is the best and max and min are the worst for median initial value; $p = -3$ is the best followed by min while max is the worst for high initial value. The angle minimization within convex hull algorithm outperforms the linear scheme slightly in all cases, but does not surpass the best method of $(p, w)$-mean family in each case. As we have mentioned, this is not a fair comparison as the result of $(p, w)$-mean can fall outside convex hull and hence have larger "consensus step size." We find the cube hull angle minimization method descends faster than all the others in all cases. This is not surprising, as it tries to descend in the gradient direction as much as possible within the largest possible choice set, leading to the fastest decrease in objective value.

The main contribution of proposing these different consensus scheme is providing the flexibility for the algorithm so that the best choice is available for the given requirements. For example, when running distributed optimization algorithms we usually need two termination criteria: one regarding the convergence of $\mathbf{x}_i$ to $\mathbf{x}^*$ for all $i$, and another concerning the convergence of $U(\bar{\mathbf{x}})$ to $U(\mathbf{x}^*)$. If one cares about the latter much more than the former, say one only wants to find the optimal value $U(\mathbf{x}^*)$ within a tolerance of $10^{-3}$, then gradient-oriented within cube hull is clearly the best choice. If one has some tolerance requirement of the former, i.e., convergence to optima, we can also use the cube hull method with smaller $\delta$. Even if one only cares about the former, from Figure 3.4 we know that usually there is a better scheme in the $(p, w)$-mean family than the linear consensus. If the application focus on a part of the objective function rather than the whole, we can also revise the objective function in Algorithm 3.8 by tracking a gradient of that part. For set $\mathcal{K}$ lying outside $\mathbb{R}^d_+$ where $(p, w)$-mean family is no longer feasible, Algorithm 3.5 endows us to choose any other valid family of transformations on the set to construct a "consensus choice hull."

## 3.7 Conclusion and Future Directions

In this chapter, we studied various nonlinear consensus schemes for distributed optimization. As illustrated in our alternative proof of the convergence of the NEXT algorithm, the consensus step is a subprojection onto the consensus plane which is the fast process, while the consensus variable descends to the optimum on the plane which is the slow process. From this perspective, many subprojections outside the paradigm of linear consensus can be considered for distributed optimization. We considered taking monotonic nonlinear transformations before taking linear averages; we established convergence when such consensus schemes are combined with NEXT, and analyzed the convergence rate when it is combined with DGD. We further reestablished the convergence result for NEXT when the averaging matrices are no longer column stochastic. Combining this relaxation with the nonlinear transformation idea allows us to choose any point in the “shrunk cube hull” as the next iterate during the consensus step, which is a very general consensus scheme. Numerical results show that depending on the relation between initial variables and the optimum, various proposed schemes can outperform traditional linear consensus schemes.

However, the underlying reasons for nonlinear consensus sometimes surpassing linear consensus are not entirely clear. The alignment with the gradient is one of the reasons, but it does not seem to be the complete story as the proposed gradient-oriented algorithms do not always perform the best in the comparisons. The nonlinear transformation scheme does not show any theoretical advantage in our convergence rate analysis either, which is not too surprising as it is a worst case analysis and the bound could be loose. To better understand how consensus schemes may affect the speed of convergence, closer investigation of the local properties of the iterates and objective functions as well as the consensus schemes themselves is needed as the results seem to be data dependent.

## CHAPTER IV

# Multi-Agent Multi-Armed Bandit with Hierarchical Information Structure

## 4.1 Introduction

Most reinforcement learning (RL) literature assumes that the underlying model is Markovian. In the single-agent case, these include Markov Decision Processes (MDPs), and when the agent only has noisy observations on the states, the Partially Observable MDP (POMDP). Parallel to the two models in the multi-agent case are the decentralized MDP (Dec-MDP) (similar to a stochastic game (SG)) and the decentralized POMDP (Dec-POMDP) (similar to a Partially Observable SG (POSG) [48]) have been proposed; among them, Dec-POMDP is the most general model [15]. Depending on the agents' objectives, the models range from fully cooperative, where the agents share a common goal, to fully competitive, where the agents' objectives either sum up to zero, such as the game of Go [58], or can be more complex [97]. In keeping with the theme of this thesis, including in this chapter and the next chapter, we focus on the fully cooperative case.

Due to *information asymmetry*, which refers to the mismatch in the set of information each agent has in a multi-agent environment, finding optimal policies for Dec-POMDPs is particularly hard. In fact, a finite-horizon Dec-POMDP with more than one agent is NEXP-complete [15], implying a doubly exponential complexity growth in the horizon length. Even finding an "$\epsilon$-approximate solution" is NEXP-complete [102, 95]. In decentralized control theory, theoretical solutions have been proposed to find the optimal control laws for Dec-POMDPs. Notably among them is the common information (CI) approach [88], a framework that decomposes the decision of a full policy into the decision of a "prescription policy" from the CI known by all the agents, and the "prescription" itself which is a full characterization of how the agents should act based on any realization of their own private information (PI). This approach effectively transforms the decentralized model back to a centralized one from the view of a fictitious "coordinator" who only observes the CI, and permits a coordinator level sequential decomposition using a belief state in a manner similar to POMDPs [66].

The challenge increases greatly in the multi-agent reinforcement learning (MARL) setting where the *model* – transition probabilities, observation kernel, and reward function – is unknown. When the agents learn concurrently, information asymmetry causes another issue called the "non-stationarity issue," since the effective environment observed by each agent is time-varying as the other agents learn and update their policies. The issue can be alleviated in principle by the "centralized learning and distributed execution" scheme [37] as the learning is from the coordinator's viewpoint; indeed, if agents only update their policies using CI, they can perfectly track others' policies. However, there is still a big gap in applying the CI approach to the MARL setting. First, the Bayesian updates of the belief state in the CI approach require the knowledge of the model, which is not available in the MARL setting. Moreover, the linear growth of length of private histories leads to the doubly exponential growth of the space of prescriptions in time, which is explosively large even for toy-size environments and forbids any practical explorations in such space. One natural question is whether we can restrict attention to some policies (and prescriptions) that take some state variable as inputs without losing much performance, where the state variables encapsulate the crucial information relevant to future decisions in a time-invariant domain, and where the representations (ways of encapsulation) can be learned without the knowledge of the model.

In this chapter, we formulate good approximate common and private state representations for learning close-to-optimal policies in unknown finite-horizon Dec-POMDPs, where each agent receives its own private information plus a *common* observation. The agents also share the same commonly observed rewards; however, they may not know each others' actions. We propose conditions for an approximate sufficient private state (ASPS), which compresses an agent's private information, i.e., its action observation history (AOH), and conditions for an approximate sufficient common state (ASCS), which compresses the fictitious coordinator's AOH, with the actions being ASPS-based prescriptions and the observations being common observations. Critically, we derive the optimality gap in terms of the error parameters of our compression and the remaining time steps, between the value functions of two dynamic programmings (DPs): one in Algorithm 4.1 for the optimal policy using the CI approach without compression, with states being the complete coordinator's AOHs and actions being the prescriptions from [88]; and the other in Algorithm 4.6 using our framework with the states being any *valid*[1] ASCSs and the actions being ASPS-based prescriptions for valid ASPS. Our framework generalizes a number of results in the literature: first, it extends the approximate information state (AIS) framework [112, 113] to the multi-agent setting; second, it extends the CI approach [88] and the follow-up sufficient private information (SPI) framework that compresses the private states [119], to their general approximate state representation counterpart; third, it generalizes the work by [84] to include *non-injective* compressions and a general approximate common state representation. Our results can provide guidance on designing Deep Learning (DL) structures to learn the (compressed) state representations and the optimal policies (using learned representations) under the centralized learning distributed execution scheme, which applies to practical offline or online MARL settings.

The rest of the chapter is organized as follows. In the rest of this section, we review the literature and give two motivating examples. We summarize the Dec-POMDP model, the AIS framework, and a few existing DP approaches for decentralized control in Section 4.2. In Section 4.3, we introduce our conditions for general common and private state compressions, and provide a DP-equivalence proof. We extend the results to the approximate cases and derive the optimality gap for value functions in Section 4.4. In Section 4.5, we compare our results to existing schemes, and discuss the practical implications of our results. We discuss a change detection (CD) scheme for piece-wise continuous time-varying environments in Section 4.6, and draw conclusion in Section 4.7.

---

[1] Satisfying the approximation criteria.

### 4.1.1 Related Work

The problem of state representation is well studied in the single-agent POMDP case. Stochastic control theory details the conditions an *information state* (IS) needs to satisfy so that it acts as the Markov state in an equivalent MDP so one may only consider IS-based policies without loss of generality [82]; the belief state is an example of such IS [66]. The paper [113] extends the idea to an *approximate information state* (AIS), where the IS conditions hold approximately; importantly, the optimality gap of running DP with any valid AIS is quantified. Based on their AIS scheme, they propose a DL framework that learns the AIS representation without knowing the model. Recent work on *Deep Bisimulation for Control* (DBC) [133] in the DL literature uses similar ideas: they train an encoder to predict well the instantaneous rewards and transitions, and use the encoder output to train the policies. The encoder is an encapsulation or a compression. The optimality gap established is similar to the result of the infinite horizon case in [112]. There are more representation learning schemes not requiring model knowledge in the DL or RL literature, e.g. [47], with the bulk without theoretical guidance or guarantees. Other work that addresses learning the optimal policies of POMDPs includes [63, 54]. The authors of [63] consider a special type of AIS – the $N$-memory, which contains the information from the last $N$ steps. Here, the compression function is fixed but in contrast to [113], the approximation error given each history need not be uniform. When the model is known, they provide conditions that bound the regret of $N$-memory policies (policies that depend on $N$-memory), and an algorithm that finds optimal policies within this class. The first algorithm to learn the optimal policies of POMDPs with sub-linear regret in an online setting is proposed in [54]. Using a posterior sampling-based scheme, the algorithm maintains the posterior distribution on the unknown parameters of the considered POMDP, and adopts the optimal policy with respect to a set of parameters sampled from the distribution in each episode. The posterior update in the algorithm, however, heavily relies on the knowledge of the observation kernel, which is usually unknown in RL settings.

In the multi-agent context, [88] proposes a belief IS for the coordinator using the CI approach, without compressing agents' private information. The authors of [117] further compress private histories to *sufficient private information* (SPI) so that the corresponding spaces of the belief IS and prescriptions are time-invariant. They identify conditions such that restricting attention to SPI-based policies is without loss of optimality. However, not only do they consider a control setting where the model is required, but also only present compression of the common history to a belief state, which is a narrow class of compression schemes. Nevertheless, this work will be a starting point of our work. The authors of [84] consider an information state embedding that injectively maps agents' histories to representations in a fixed domain, and quantify the effect of the embedding on the value function like [112]. However, their requirement that the mapping is injective is impractical for two reasons: one, an injective mapping does not reduce the policy complexity; and two, real world applications often demand non-injective encapsulations - e.g., tiger [59] where one IS is the number of right observations minus the number of left observations, which is a non-injective compression of the AOH. Moreover, they also compress the common state to a belief state, but it is unclear how this can be done in practice without model information.

Another line of work in deep-MARL literature also applies the notion of CI (also known as the common knowledge) to solving MARL problems [35, 42, 73, 111]. They search for optimal policies for a Dec-POMDP when the model is known, while we consider designing sample efficient and lower regret learning algorithms in an offline or online MARL setting for an unknown model. Moreover, many of them involve heuristic or approximation methods without knowing the potential loss from the approximations or apply a variety of machine learning schemes without a theoretical basis or understanding.

### 4.1.2 Motivating Examples

In this subsection, we introduce two motivating examples that call for the modeling of the multi-agent system by Dec-POMDPs. The common feature is the asymmetric information structure among the agents that makes finding optimal policies without model knowledge difficult.

#### 4.1.2.1 Multiple Access Communication with Collision Channel

One example of multi-agent team problems widely used in communication system applications is the multiple access problem where the users share a common communication channel [98]. Fixed assignment protocols such as the Time-Division Multiple Access (TDMA) protocol divide the channel resource into transmission units and pre-allocate to users to avoid collision [98]. Here, we focus on contention based protocols, where collisions could occur and users obtain partial observations from the system and make transmission decisions on the fly without pre-coordination, much like the MARL setting we are considering. Among them, with channel sensing, Carrier Sense Multiple Access (CSMA) type protocols have been proposed, and employed in the renowned IEEE 802.11 wireless local area network (WLAN) protocols [98].

Here, we describe the system model described in [98], and then explain how it fits in the Dec-POMDP model later in Section 4.2.1. There are $N$ users with generic index $n$; each user $n$ is equipped with an infinite size buffer, and we let $Q_t^n$ denote the queue length at time $t$, which can only be observed by $n$. At the beginning of time $t$, each user $n$ observes an *i.i.d.* (independent across time steps and across users as well) Bernoulli packet arrival with parameter $p^n$; that is, user $n$ will receive a packet $W_{t-1}^n = 1$ with probability $p^n$ and $W_{t-1}^n = 0$ otherwise[2]. Each user $n$ then decides whether to make a transmission, denoted by $U_t^n = 1$, or not, denoted by $U_t^n = 0$. If there is a single transmission, the packet will reach the access point successfully and be removed from the associated queue; on the other hand, multiple transmissions will collide. All the users will be notified either there was no transmission, there was a successful transmission, or there were multiple transmissions such that a collision occurred; i.e., the users will get a feedback $F_t \in \{0, 1, e\}$ corresponding to the described three cases. It is easy to see that the queue of user $n$ evolves as follows

$$Q_{t+1}^n = \mathcal{T}^n(Q_t^n, W_t^n, U_t^{1:N}) \triangleq W_t^n + \max\left(Q_t^n - U_t^n \prod_{m \neq n} (1 - U_t^m), 0\right). \tag{4.1}$$

In the MARL setting, the parameters $\{p^n\}_{n=1}^N$ are unknown to the users. A communication protocol is called stable if for every $\epsilon > 0$, there exists a finite number

---

[2] Note that $W_{t-1}^n$ is already added into the queue $Q_t^n$.

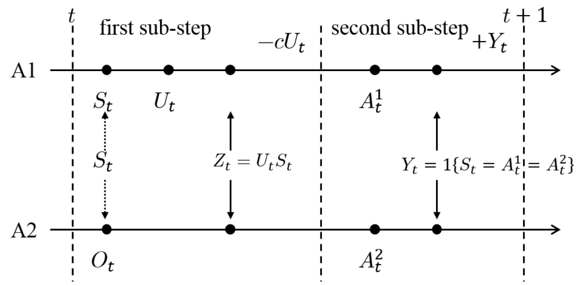


Figure 4.1: Time-line of states, observations, actions, and rewards in the matching problem.

$K$ (possibly depends on $\epsilon$) such that $\mathbb{P}(Q_t^n > K) < \epsilon$ for all time $t$ and user $n$. Obviously, when $p^{1:N} = (p^1, \cdots, p^N)$ is such that $\sum_{n=1}^{N} p^n > 1$, there is no way any protocol can stablize the queues. The goal is thus set to find a decentralized policy that stablizes the communication system for any $p^{1:N}$ such that $\sum_{n=1}^{N} p^n < 1$.

#### 4.1.2.2 Matching Problem

In another example called the "matching problem," two agents have imperfect measurements of the state, and are rewarded if both agents use their actions to "match" the state. The problem was first proposed and studied in [86]. Although simple, the problem captures the information asymmetry aspect in multi-agent problems, as well as communication as a costly action by which agents may actively exploit to increase the CI between agents.

In the matching problem, A1 is the informed agent, and A2 is the uninformed agent. The state space, the observation spaces, and the action spaces are all the same $\mathcal{S} = \mathcal{A}^1 = \mathcal{A}^2 = \mathcal{O}^1 = \mathcal{O}^2 = \{+1, -1\}$, and both agents try to match their actions to the state to get rewards. At time 0 nature draws the initial value of the state variable $S_1$ with prior $\mathbb{P}(S_1 = +1) = 1 - \epsilon$ and $\mathbb{P}(S_1 = -1) = \epsilon$, which is known to both agents. As illustrated in Figure 4.1, each time step is split into two sub-steps. In the first sub-step, the following events occur sequentially:

- The state $S_t$ evolves according to the kernel $\mathbb{P}(S_t = S_{t-1} | Y_{t-1} = 1) = \mathbb{P}(S_t = -S_{t-1} | Y_{t-1} = 0) = 1 - \epsilon$ and $\mathbb{P}(S_t = S_{t-1} | Y_{t-1} = 0) = \mathbb{P}(S_t = -S_{t-1} | Y_{t-1} = 1) = \epsilon$.

- A1 perfectly observes $S_t$, i.e., $O_t^1 = S_t$, while A2 partially observes $O_t^2$, where $\mathbb{P}(O_t^2 = S_t) = \beta$ and $\mathbb{P}(O_t^2 = -S_t) = 1 - \beta$. We will treat $O_t^1$ directly as $S_t$ so that we can drop the superscript in $O_t^2$ and write $O_t$ instead.

- A1 decides $U_t \in \{0, 1\}$, where 1 means to communicate and 0 means not.
- Both agents observe the communication result $Z_t = U_t S_t$. A1 knows what it communicated and so our assumption is that A2 gets the message without any errors if it was sent.
- They receive reward $-cU_t$ where $c \in [0, 1)$ reflects the communication cost.

In the second sub-step, the following events occur sequentially:

- The state makes a dummy change, i.e. remains in $S_t$.
- A1 decides its action $A_t^1$ and A2 decides its action $A_t^2$. This is done simultaneously and neither agent observes the action of the other agent.
- Both agents observe the coordinating result $Y_t \in \{0, 1\}$, where $Y_t(S_t, A_t) = \mathbb{I}\{S_t = A_t^1 = A_t^2\}$ and $A_t = (A_t^1, A_t^2)$.
- They receive reward $Y_t$.

Our model differs from that of [86] in the following aspects. First, in their model the state transits deterministically, in which case agents may just apply the trivial policy of playing $+1$ at all times for both agents to achieve optimality; this corresponds to the case of $\epsilon = 0$ in our model. Second, in our model communication is costly but in [86] there is no such penalty. Since A1 has perfect observations, if at a given time-step it sends its observation to A2, then A2 can directly follow, and both will achieve the reward of 1 at that time-step. Therefore, if the communication costs nothing, then A1 could just send its observation in every time-step and the agents will get the maximum possible long-term reward; note that this is one simple means to achieve optimal performance with communication in [86].

The problem will be revisited in Section 5.6, where we derive its optimal policy using the CI approach, based on which a near-optimal regret achieving bandit algorithm is constructed when the model is unknown.

## 4.2 Preliminaries

We introduce the Dec-POMDP model in Section 4.2.1, and the non-stationarity issue of the model that arises in the RL context in Section 4.2.2. In Section 4.2.3 we introduce the approximate information state (AIS) framework [112, 113] that learns the state representation in the (single-agent) POMDP setting when the model is unknown. Then in Section 4.2.4 we explain the CI approach as the solution to the non-stationarity issue in the decentralized control context, and give three DPs that solve the decentralized control problem in the Dec-POMDP setting using the CI approach: the DP with no compression for any state in Section 4.2.4.2, the DP with the common state compressed to a belief state proposed in [88] in Section 4.2.4.3, and the DP with the common state compressed to a belief state proposed and the private state compressed to sufficient private information (SPI) proposed in [117, 119] in Section 4.2.4.4.

### 4.2.1 Dec-POMDP Model

The model of Dec-POMDP is commonly used in the fields of decentralized control and MARL. Suppose there are $N$ agents in the system. A Dec-POMDP model is a tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}_T, \mathcal{R}, \mathcal{O}, \mathbb{P}_O, T, \kappa, \mathbb{P}_I)$ for the finite horizon case or $(\mathcal{S}, \mathcal{A}, \mathbb{P}_T, \mathcal{R}, \mathcal{O}, \mathbb{P}_O, \lambda, \kappa, \mathbb{P}_I)$ for the infinite horizon case, where the quantities are explained in the following:

- $\mathcal{S}$ is the state space.

- $\mathcal{A} = \mathcal{A}^1 \times \cdot \times \mathcal{A}^N$ is the joint action space whose elements are joint actions $A = (A^1, \dots, A^N)$ where agent $n$'s action $A^n \in \mathcal{A}^n$ comes from the action space of agent $n$.

- $\mathbb{P}_T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel mapping a current state and a joint action to a distribution of new states. Recall the notation $\Delta(\cdot)$ denotes a distribution on the space.

- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ is the reward function mapping a current state and a joint action to a probability distribution on the real values.

- $\mathcal{O} = \mathcal{O}^1 \times \cdot \times \mathcal{O}^N$ is the joint observation space whose elements are joint observations $O = (O^1, \ldots, O^N)$ where agent $n$'s observation $O^n \in \mathcal{O}^n$ comes from the observation space of agent $n$ and is only observed by agent $n$. In this chapter, we consider an enlarged joint observation space $\mathcal{O} = \mathcal{O}^0 \times \mathcal{O}^1 \times \cdot \times \mathcal{O}^N$ where $O = (O^1, O^{1:N})$ and $O^0 \in \mathcal{O}^0$ is commonly observed. Details will be explained later.

- $\mathbb{P}_O : \mathcal{S} \to \Delta(\mathcal{O})$ is the observation kernel mapping a current state to a distribution of joint observations. In some literature, $\mathbb{P}_O$ is a function of the current joint actions, and even the new state as well, i.e., $\mathbb{P}_O : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \Delta(\mathcal{O})$; we will however consider the more concise former model here.

- $T$ is the time horizon when finite horizon problems are considered, while $\lambda$ is the discount factor when infinite horizon problems are considered.

- $\kappa$ is a function specifying what will be shared in agents' histories to become common information to all agents. In this chapter, we assume the common information revealed at time $t$ only contains the common observation $O_t^0$ and generated reward $R_t$ (which will be included as part of $O_t^0$). The justification of such choice will be explained later.

- $\mathbb{P}_I \in \Delta(\mathcal{S})$ is the initial state distribution.

In comparison to the standard Dec-POMDP model [95], we have an additional common observation (including the reward), and our observations depend only on the current state.

In this thesis, we assume the agents have perfect recall. At time $t$, agent $n$ observes $(O_t^0, O_t^n)$ generated from $\mathbb{P}_O(S_t)$, then uses the policy $A_t^n = g_t^n(O_{1:t}^0, g_{1:t-1}, H_t^n)$ to select its action, where $g_s = g_s^{1:N}$ and $H_t^n = (A_{1:t-1}^n, O_{1:t}^n)$ is agent $n$'s private history and known as its AOH. The agents receive a reward $R_t \triangleq R(S_t, A_t)$ sampled from $\mathcal{R}(S_t, A_t)$, and the next state $S_{t+1}$ is generated from $\mathbb{P}_T(S_t, A_t)$. We also assume $\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$, and $T$ (or $\lambda$ for the infinite horizon setup) are finite and known in advance. On the other hand, $\mathbb{P}_T$, $\mathcal{R}$, $\mathbb{P}_O$, and $\mathbb{P}_I$, which are collectively referred to as the "model," may be known or unknown depending on the problem context.

In the context of control problems where the model is known, the objective is to find a policy $g = \{g_t\}_{t=1}^T$ (finite horizon) or $g = \{g_t\}_{t=1}^\infty$ (infinite horizon), such that the common cumulative reward

$$\mathbb{E}\left[\sum_{t=1}^{T} R(S_t, A_t)\right] \text{ (finite horizon) or } \mathbb{E}\left[\sum_{t=1}^{\infty} \lambda^t R(S_t, A_t)\right] \text{ (infinite horizon)} \quad (4.2)$$

is maximized, where the expectation is taken over the measure generated by policy $g$ applied to the model $(\mathbb{P}_T, \mathcal{R}, \mathbb{P}_O, \mathbb{P}_I)$. Such a policy is called an optimal policy and denoted as $g^*$. On the other hand, in the context of RL problems, the model is assumed to be unknown, and the objective is to minimize "regret," which is the actual realized cumulative reward deducted from the best cumulative reward in hindsight (or simply the reward under optimal policy) in expectation; in other words, the objective of RL can be thought of as finding the procedure (learning algorithm) that figures out $g^*$ as fast as possible. In some situations, the agents might know certain structures of $(\mathbb{P}_T, \mathcal{R}, \mathbb{P}_O, \mathbb{P}_I)$ but do not have the information of some values, which we call "parameters" in the structures; such situations still fall into the category of RL, but are easier than the cases where the model is completely unknown.

The information $H_t^n$ is presumed to be private and only known to agent $n$. The common information (CI) specification function $\kappa = \{\kappa_t\}_{t=1}^T$ or $\kappa = \{\kappa_t\}_{t=1}^\infty$ is a function such that at time $t$, $\kappa_t$ specifies what information among $H_t^{1:N} = (H_t^1, \ldots, H_t^N)$ should be shared to become common information at time $t$, which is denoted as $C_t$ and is a subset of $H_t^{1:N}$. The specification $\kappa_t$ is a function as what is shared may potentially depends on all the things that happen in the system so far, i.e. $C_t = \kappa_t(H_t^{1:N}, S_{1:t})$. This is the model used in [88]. Since the agents have perfect recall, once an information is shared, it cannot be taken back. That is to say, if we let the common informations from different time steps form a sequence of sets $\{C_t\}_t$, then $\{C_t\}_t$ is an increasing sequence of sets, i.e. $C_t \subseteq C_{t+1}$ for all $t$.

In the standard Dec-POMDP setting [95, 101], however, nothing is shared throughout so that $C_t = \emptyset$ for all $t$. The two models can be bridged if the spaces and kernels are allowed to be *time-varying.* Then the state can first transit to a 'dummy preparation state" and the agents can observe the shared common information *through their private observations.* In the control context, the two are equivalent; however, in the RL context, the two models are different, since it is not *common knowledge* that the private observations from the dummy state are the common information coming from a subset of $H_t^{1:N}$ and are the same for all agents. Moreover, the time-varying model and the CI specification model have a major drawback – their complexities grow exponentially with time, as they try to specify the dependency of what should be shared on history. Therefore, in this chapter we consider a third model, which is used in [117, 119, 84] as well; in this model, the observation space is enlarged to $\mathcal{O} = \mathcal{O}^0 \times \mathcal{O}^1 \times \cdots \times \mathcal{O}^N$, where at time $t$ a common (or public) observation $O_t^0 \in \mathcal{O}^0$ is observed by all agents in addition to their private observations. This way the agents can make decisions based on common information, while keeping the model time-invariant so that the complexity does not grow with time. Note that this model corresponds to the standard Dec-POMDP model and the CI model in the sense that at time $t$, agent $n$ observes $(O_t^0, O_t^n)$, and $O_t^0$ is immediately shared as common information, so that $C_t = O_{1:t}^0$.

As mentioned earlier, throughout the thesis, we only consider the "team problem," i.e., the reward or utility function $\mathcal{R}$ is same for all agents[3]. In this setting, the global optimal policy (or policies) that achieves the maximal long-term reward exists. When the reward function can potentially be different for different agents, more complex solution concepts such as Nash Equilibrium or Perfect Bayesian Equilibrium have to be introduced; the readers are referred to [117, 118, 97, 116] for more details on this topic.

#### 4.2.1.1 Linkage to the Motivating Examples

The multiple access problem introduced in Section 4.1.2.1 can be modeled as a Dec-POMDP. Specifically, at time $t$, the state is the tuple of queue lengths, new packet arrivals, and the transmission decisions from $t-1$ $S_t = (Q_t^{1:N}, W_{t-1}^{1:N}, U_{t-1}^{1:N})$, the action of agent $n$ is the transmission decision $A_t^n = U_t^n$, the transition probability

---

[3] We further assume the sampled reward $R_t$ is also the same for all agents; this is actually not required for team problems, but will save us an additional layer of concentration bounds.

can be completely characterized by the parameters of the Bernoulli random variables

$$\begin{aligned}&\mathbb{P}_T((Q_{t+1}^{1:N}, W_t^{1:N}, U_t^{1:N})|(Q_t^{1:N}, W_{t-1}^{1:N}, U_{t-1}^{1:N}))\\ &= \mathbb{I}\{Q_{t+1}^{1:N} = \mathcal{T}^{1:N}(Q_t^{1:N}, W_t^{1:N}, U_t^{1:N})\} \cdot \prod_{n=1}^{N} p_n^{W_t^n}(1-p_n)^{(1-W_t^n)}\end{aligned}$$

where $\mathcal{T}^n$ is given in (4.1), the common observation is the feedback $O_t^0 = F_{t-1}$ generated by

$$F_{t-1}(Q_t^{1:N}, W_{t-1}^{1:N}, U_{t-1}^{1:N}) = F_{t-1}(U_{t-1}^{1:N}) = \begin{cases} 0, & \text{if } \prod_{n=1}^{N}(1-U_{t-1}^n) = 1, \\ 1, & \text{if } \prod_{n=1}^{N}\Big(1 - U_{t-1}^n \prod_{m\neq n}(1-U_{t-1}^m)\Big) = 0, \\ e, & \text{otherwise,} \end{cases}$$

the private observation of agent $n$ contains its own queue length and packet arrival status $O_t^n = (Q_t^n, W_{t-1}^n)$. The reward function is not clearly defined in the problem. However, we can easily identify an instantaneous reward as the indicator function of a successful transmission $R_t(S_t, A_t) = R_t(F_t) = \mathbb{I}\{F_t = 1\}$, so that the overall objective is throughput maximization.

For the matching problem introduced in Section 4.1.2.2, it is nearly already in the form of Dec-POMDP, except the inclusion of the two sub-steps makes it time-varying. If one insists a time-invariant model, we can simply adopt a "superstate" consisting of a mode $M_t$ and the original state $S_t$, where the mode deterministically transits between $M_t = 1$ (first sub-step) and $M_t = 2$ (second sub-step). The total reward for the entire time step is $R_t = R_t(S_t, A_t, U_t) = \mathbb{I}\{S_t = A_t^1 = A_t^2\} - cU_t = Y_t - cU_t$. Just before A1 takes action in the first sub-step at time $t$, the agents' information structure is as follows: the common information is $(Z_{1:t-1}, Y_{1:t-1})$[4], A1's private information includes $(S_{1:t}, A_{1:t-1}^1)$, and A2's private information includes $(O_{1:t}, A_{1:t-1}^2)$. Just before both agents take actions in the second sub-step at time $t$, the agents' information structure is as follows: the common information is $(Z_{1:t}, Y_{1:t-1})$, A1's private information includes $(S_{1:t}, A_{1:t-1}^1)$, and A2's private information includes $(O_{1:t}, A_{1:t-1}^2)$.

---

[4] Notice that $U_t$ can be recovered from $Z_t$ losslessly: $U_t = |Z_t|$; hence, $U_{1:t-1}$ is also common information. We do not put it in the common information tuple specifically since all the information it carries is a subset of that carried by $Z$.

### 4.2.2 Non-stationarity Issue in MARL

Having understood the Dec-POMDP model, we can reiterate the non-stationarity issue that arises in the MARL setting. Note that agent $n$'s private AOH $H_t^n$ is not shared as CI and is kept private to agent $n$ itself; as agent $n$ uses this private information to update its policy $g^n$ in MARL context, this "asymmetric information structure" causes an issue. In multi-agent systems, the "effective environment" agent $n$ experiences not only depends on the true underlying environment but also the policies of other agents $g^{-n} = (g^1, \ldots, g^{n-1}, g^{n+1}, \ldots, g^N)$. As other agents update their policies based on their private information unknown to agent $n$, these policies are changing in an intractable way to agent $n$ and the effective environment to agent $n$ is non-stationary. It is known that if all agents just perform single-agent RL algorithms, their joint policy may not converge to an optimum [33]. In an alternative method called the peron-by-person (PBP) approach [104, 50, 16], one agent (say agent $n$) updates its policy $g^n$ to best respond to other agents' joint policy $g^{-n}$ and the underlying environment, and this is performed iteratively for all agents one at a time. However, with the PBP approach, one may only end in a locally optimal joint policy instead of a globally optimal one. To overcome this issue, the "centralized learning and decentralized execution" scheme [37] is proposed. Such scheme is supported in control theory by the CI approach, which will be the main theme of this chapter and formally introduced in Section 4.2.4.

### 4.2.3 AIS Framework

We start with the single-agent POMDP, which is a tuple of $(\mathcal{S}, \mathcal{A}, \mathbb{P}_T, R, \mathcal{O}, \mathbb{P}_O, T, \mathbb{P}_I)$ as in Section 4.2.1 where the action space and observation space are no longer product spaces, and there is no common observation or information sharing as there is only one agent. At time $t$, the agent's policy is of the form $g_t : \Omega(H_t) \to \Omega(A_t)$, where $H_t = (A_{1:t-1}, O_{1:t})$ is the agent's AOH. Note the policy space grows exponentially in $t$ as the length of $H_t$ grows linearly in $t$.

It is well-established in stochastic control theory that when the model is known, the POMDP can be converted into an MDP with a "belief state," and solved by DP theoretically with the belief state [66]. In the setting of POMDP, the belief state is a conditional distribution on the true state given the entire history, i.e., it is of the form $\mathbb{P}(S_t|H_t)$, and can evolve recursively based on Bayesian updates. The mapping from an AOH to its belief state $\mathbb{P}(S_t|\cdot) : \Omega(H_t) \to \Delta(\mathcal{S})$ can be seen as a lossless compression for decision purposes. With the belief state, the policy now only depends on an input from a time-invariant space $\Delta(\mathcal{S})$ as we get away from the curse of dimensionality.

When the model is unknown, however, the belief state cannot be formed. The belief state is initialized by the initial distribution $\mathbb{P}_I$, and then updated from the previous belief state by the Bayesian update

$$\begin{aligned} \mathbb{P}(s_t|h_{t-1}, a_{t-1}, o_t) &= \frac{\sum_{s_{t-1}} \mathbb{P}(s_t, s_{t-1}, o_t|h_{t-1}, a_{t-1})}{\sum_{s'_t, s_{t-1}} \mathbb{P}(s'_t, s_{t-1}, o_t|h_{t-1}, a_{t-1})} \\ &= \frac{\sum_{s_{t-1}} \mathbb{P}(s_{t-1}|h_{t-1})\mathbb{P}_T(s_{t-1}, a_{t-1}, s_t)\mathbb{P}_O(s_t, o_t)}{\sum_{s'_t, s_{t-1}} \mathbb{P}(s_{t-1}|h_{t-1})\mathbb{P}_T(s_{t-1}, a_{t-1}, s'_t)\mathbb{P}_O(s'_t, o_t)}. \end{aligned} \tag{4.3}$$

Without the knowledge of $\mathbb{P}_T$ and $\mathbb{P}_O$, the equation cannot be computed. The AIS framework proposed in [112] provides us the conditions for a compression mapping to be sufficient for DP purposes, and a way to learn an approximately sufficient compression mapping in practice without knowing the model.

From stochastic control theory, an information state (IS) which encapsulate the AOH $H_t$, is a state that evolves in a state-like manner, suffices for performance evaluation, and suffices for transition prediction.

**Definition 4.1:** *An information state $Z_t$ is the output of a function $Z_t = \vartheta_t(H_t)$ that satisfies the following properties:*

***(IS1)*** *It evolves recursively $Z_{t+1} = \phi_t(Z_t, A_t, O_{t+1})^5$.*

***(IS2)*** *It suffices for performance evaluation $\mathbb{E}[R_t|h_t, a_t] = \mathbb{E}[R_t|z_t, a_t] \ \forall \ h_t, a_t$.*

***(IS3)*** *It suffices for self-prediction $\mathbb{P}(Z_{t+1}|h_t, a_t) = \mathbb{P}(Z_{t+1}|z_t, a_t) \ \forall \ h_t, a_t$.*

Alternatively, it is equally good if the state can predict the new observation.

**Definition 4.2:** *An information state $Z_t$ is the output of a function $Z_t = \vartheta_t(H_t)$ that satisfies (IS1) and (IS2) in Definition 4.1 and the following property:*

---

[5] In [112], this condition is not required for their main results when Definition 4.1 is used. However, it is required when Definition 4.2 is used, as (IS1) and (IS3') imply (IS3).

***(IS3')*** *It suffices for predicting the observation* $\mathbb{P}(O_{t+1}|h_t, a_t) = \mathbb{P}(O_{t+1}|z_t, a_t) \; \forall \; h_t, a_t$.

Examples of IS include: the AOH $H_t$ itself which trivially satisfies all properties; the belief state as we learn from stochastic control theory for partially observed systems; the number of left observations subtracted by right observations in the *tiger* environment [59]. None of them work in the general unknown model RL setting – using $H_t$ itself has the complexity issue, the map from $H_t$ to the belief state is unknown as we mentioned, and that of *tiger* is just a special case. Hence, we need the notion of AIS that satisfies IS properties to some extent, but is easier to be found and implemented in practice. An AIS is a state that satisfies Definition 4.1 or Definition 4.2 approximately. In particular, [112] suggests that AIS is a representation encapsulating the information in $H_t$ that is approximately sufficient for decision purposes into a time-invariant space. We have the following two definitions that correspond to Definition 4.1 and Definition 4.2, respectively.

**Definition 4.3:** *An* $(\epsilon, \delta)$*-approximate information state* $\widehat{Z}_t$*is the output of a function* $\widehat{Z}_t = \widehat{\vartheta}_t(H_t)$ *that satisfies the following properties:*

***(AIS1)*** *It evolves recursively* $\widehat{Z}_{t+1} = \widehat{\phi}_t(\widehat{Z}_t, A_t, O_{t+1})$.

***(AIS2)*** *It suffices for approximate performance evaluation* $|\mathbb{E}[R_t|h_t, a_t] - \mathbb{E}[R_t|\widehat{z}_t, a_t]| \leq \epsilon \; \forall \; h_t, a_t$.

***(AIS3)*** *It suffices for approximate self-prediction* $\mathcal{K}(\mathbb{P}(\widehat{Z}_{t+1}|h_t, a_t), \mathbb{P}(\widehat{Z}_{t+1}|\widehat{z}_t, a_t)) \leq \delta \; \forall \; h_t, a_t$, *where* $\mathcal{K}(\cdot, \cdot)$ *is a distance between two distributions*[6].

We also have the alternative option for the state to predict observations.

**Definition 4.4:** *An* $(\epsilon, \delta)$*-approximate information state* $\widehat{Z}_t$ *is the output of a function* $\widehat{Z}_t = \widehat{\vartheta}_t(H_t)$ *that satisfies (AIS1) and (AIS2) in Definition 4.3 and the following property:*

***(AIS3')*** *It suffices for approximately predicting the observation* $\mathcal{K}(\mathbb{P}(O_{t+1}|h_t, a_t), \mathbb{P}(O_{t+1}|\widehat{z}_t, a_t)) \leq \delta \; \forall \; h_t, a_t$.

The value function at $t$ obtained from Bellman equations with $\widehat{Z}$'s as states falls behind the optimal value function at the most by an expression linear in $T - t$, $\epsilon$, and $\delta$ [112]. When $\epsilon = \delta = 0$, the expression is 0, and the AIS $\widehat{Z}$ degenerates to an IS $Z$.

---

[6] For example, Wasserstein and total variation distance [46].

When the discounted infinite horizon setting is considered, the value will converge to the true optimal value as the horizon of the computed Bellman equations approaches infinity (due to discounting).

In [112], a DL framework is provided to find an "approximate mapping" $\widehat{\vartheta}_t(\cdot)$ for any given POMDP model. The idea is to interpret the quantities in the LHS of (AIS2) and (AIS3) as driving the learning loss in DL, and let existing DL optimization algorithms find good mappings. The resulting AIS can then be used as the state in common policy approximation methods to find a near-optimal policy.

### 4.2.4 DPs Using CI Approach

#### 4.2.4.1 Common Information Approach

We now introduce the common information (CI) approach proposed in [88] that effectively transforms a Dec-POMDP back to POMDP from the "coordinator's view." Note the policy can be decomposed into two parts: the part depending on CI, and the part depending on private information (PI). The approach incorporates the latter part into a new action called "prescription," and designs an equivalent decision rule that solely depends on the former part.

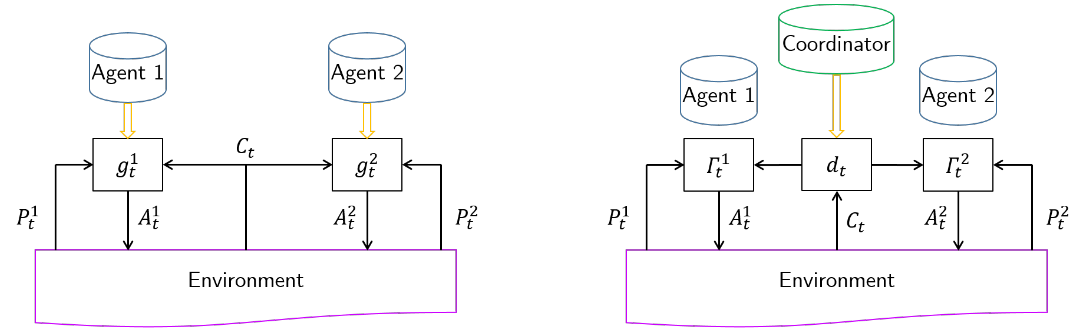
We learn from Section 4.2.1 that at time $t$, the CI specification $\kappa_t$ divides the union set of information available to all agents into the CI $C_t$ that is known to all agents, and agent $n$'s PI $P_t^n \triangleq H_t^n \setminus C_t$ that is known only to agent $n$ for all $n \in [N]$. We only consider *decentralized policies* to be applicable in decentralized systems, which are of the form $g_t = g_t^{1:N}$ where $A_t^n = g_t^n(H_t^n) = g_t^n(C_t, P_t^n)$ for all $n \in [N]$. The decision of $A_t^n$ can be split into two steps. In the first step, given any common information $C_t$, the agent decides $g_t^n$ and forms

$$d_t^n(C_t)(\cdot) \triangleq g_t^n(C_t, \cdot) \triangleq \Gamma_t^n(\cdot) \quad \forall \ n \in [N] \tag{4.4}$$

based on some decision rule[7] $d_t^n$; then in the second step, it simply applied $\Gamma_t^n$ to $H_t^n$ to obtain the action $A_t^n = \Gamma_t^n(H_t^n)$. For a fixed CI $C_t$, $g_t^n(C_t, \cdot) \triangleq \Gamma_t^n(\cdot)$ is a function

---

[7] One may consider deciding $g_t^n$ based on the past policies $g_{1:t-1}$ as well. However, with the past policy also further depends on $C_{t-1}$ and $g_{1:t-2}$, iterative expansion implies that this is equivalent to considering policies that only depend on $C_t$.



(a) The decision flow before decomposition. (b) The decision flow after decomposition.

Figure 4.2: Illustration of the common information approach.

that maps agent $n$'s PI $P_t^n$ to its new action $A_t^n$:

$$\Gamma_t^n \triangleq d_t^n(C_t) : \Omega(P_t^n) \to \Omega(A_t^n) \quad \forall\ n \in [N]. \tag{4.5}$$

Such function is called the "prescription," since it *prescribes* what agent $n$ should do based on any possible realization of its PI. The policy or decision rule $d_t^n$ maps any CI $C_t$ from $\Omega(C_t)$ to the space of prescriptions $\Omega(\Gamma_t^n) = \Omega(\Omega(P_t^n) \to \Omega(A_t^n))$ for agent $n$ (where $\Omega(A \to B)$ is the functional space of all functions with domain $A$ and codomain $B$). Notice that the aggregated policy

$$d_t = (d_t^1, \dots, d_t^N) : \Omega(C_t) \to \Omega(\Gamma_t^1) \times \cdots \times \Omega(\Gamma_t^N) \tag{4.6}$$

only depends on the CI, and will be referred to as the "coordinator's policy" or "coordinator's decision rule." The decomposition technique transforms the decision flow from individual agents deciding $g_t^{1:N}$ to the coordinator deciding $d_t$, and is called the CI approach, which is shown to be without loss of generality in the clss of decentralized policies (so without loss of optimality too) [88]. The transformation of the decision flows by the CI approach is illustrated in Figure 4.2.

Note that the actual decision is carried out in the the coordinator's policy from the first step and solely upon CI. We can thus eliminate the information asymmetry by transforming what is decentralized in the model back to being centralized.

One may imagine there is a fictitious coordinator who only has access to the CI, and can tell the agents how they should act based on their local PIs, that is, the coordinator decides the prescriptions. From the coordinator's perspective, the problem is now a centralized POMDP at time $t$:

- The state is the Dec-POMDP state combined with agents' PIs $(S_t, P_t^{1:N})$ unseen by the coordinator.
- The observation is the incoming CI $\delta C_t := C_t - C_{t-1}$, that is, the new information shared by the agents, where $C_{t-1} \subseteq C_t$ is guaranteed.
- The action is the tuple of prescriptions $(d_t^1(C_t), \ldots, d_t^N(C_t))$ that specify what the agents should do based on their PIs.

In practice, the coordinator is virtual and the computation of the coordinator is carried out in all agents – this is viable since the coordinator's computation only requires CI, which every agent has access to. Basically, they use only the CI to decide the prescriptions to take for all the agents in the system, and come up with their final actions based on their own prescriptions and PIs.

In stochastic control theory, a POMDP is solvable using dynamic programming (DP) with the belief state. Now that we transform the Dec-POMDP considered into a POMDP, we can then use DP to solve the problem with the state being a distribution on $(S_t, P_t^{1:N})$. Notice that the transformed POMDP is no longer time-invariant, but this does not forbid the usage of DP. Hence, the problem of Dec-POMDP is considered solvable in the stochastic control context. The details of the DP formulations will be given next.

#### 4.2.4.2 DP with No Compression

Throughout the rest of the chapter, we will concentrate on the Dec-POMDP setting with horizon $T$ and common observations, that is, agent $n$ observes $(O_t^0, O_t^n)$ at time $t$ where $O_t^0 \in \mathcal{O}^0$ is the common observation and $O_t^n \in \mathcal{O}^n$ is agent $n$'s private observation[8]. We will keep the notation that $H_t^n = (A_{1:t-1}^n, O_{1:t}^n)$ stands for agent $n$'s PI, which does not include $O_{1:t}^0$. Using the CI approach, the following is what happens at time $t$. The coordinator observes common observation $O_t^0$, selects a prescription function $\Gamma_t = \Gamma_t^{1:N}$ based on CI $C_t = O_{1:t}^0$, and sends to the agents. Agent $n$ observes

---

[8] Note that in the MARL setting we consider, the common reward is observed by all agents at the end of the time step, so that we treat $R_{t-1}$ as part of $O_t^0$.

private observation $O_t^n$, forms its AOH $H_t^n = (O_1^n, A_1^n, \dots, O_t^n)$, and applies $\Gamma_t^n$ to the AOH to obtain its action $A_t^n = \Gamma_t^n(H_t^n)$ when PI is not compressed.

We label the coordinator as agent 0. Denote the tuple of all observations of agent $n$ until time $t$ as $C_t^n$ for all $n = 0 : N$, i.e. $C_t^n = O_{1:t}^n$. For agent $n$ we consider policies of the form $A_t^n = g_t^n(C_t^0, H_t^n)$, or equivalently the policies of the form $A_t^n = g_t^n(C_t^0, C_t^n, g_{1:t-1}) = g_t^n(C_t^0, H_t^n, g_{1:t-1})$ where $g_s = g_s^{1:N}$ for $s \in [t-1]$ (via iterative expansion)[9]. Then from the perspective of the coordinator, the coordinator's policy is equivalently of the form $d_t : \Omega(H_t^0) \to \Omega(\Gamma_t)$, where $H_t^0 \triangleq (O_1^0, \Gamma_1^0, \dots, O_t^0)$ is equivalent to $(O_{1:t}^0, g_{1:t-1})$. Since this $H_t^0$ contains all the past common observations (coordinator's observations) up to $t$ and all the past prescriptions (coordinator's actions) up to $t-1$, it can be seen as the *coordinator's AOH*. We will refer to this $H_t^0$ term as the full common state (FCS), because it contains all the common information without any compression, and is the *IS* (Section 4.2.3) of the transformed-POMDP[10]. Similarly, for agent $n \in [N]$, the term $H_t^n$ contains all the PI agent $n$ has without any compression, and we will refer to it as the full private state (FPS).

From the perspective of the coordinator, the problem is now a centralized POMDP, and the goal is to find a policy $d = d_{1:T}$ that maximizes the expected cumulative reward. This permits a sequential decomposition with FCS as the state and FPS-based prescription (meaning the prescription takes FPS as its input) as the action, which is presented in Algorithm 4.1.

**Algorithm 4.1** Dynamic Programming with FCSs and FPS-based Prescriptions

```
$V_{T+1}(h_{T+1}^0) \triangleq 0$
for $t = T, \dots, 1$ do
    $Q_t(h_t^0, \gamma_t) = \mathbb{E}\left[R(S_t, \Gamma_t(H_t^{1:N})) + V_{t+1}((H_t^0, \Gamma_t, O_{t+1}^0)) | H_t^0 = h_t^0, \Gamma_t = \gamma_t\right]$
    $V_t(h_t^0) = \max_{\gamma_t \in \Omega(\Gamma_t)} Q_t(h_t^0, \gamma_t)$
```

In each time step of the algorithm, the computations are done for all $h_t^0$ and $\gamma_t$ whenever applicable, and same for all the remaining DPs in the rest of this chapter. In this case, updating the old FCS to the new one is done by direct concatenation of

---

[9] From the perspective of the designer's approach [128], the policy $g_{1:T}$ is decided even before the process starts, so adding the iterative dependence makes no difference. The reader may refer to [88] Proposition 3 for the details of the iterative expansion.

[10] It is the complete history or AOH of the transformed-POMDP (from the Dec-POMDP), which is in turn the state of the transformed-MDP (from the transformed-POMDP).

the incoming FPS-based prescription $\Gamma_t$ and common observation $O^0_{t+1}$ to the back of $H^0_t$.

#### 4.2.4.3 DP with BCS

In [88], the FCS is further compressed to the belief common state (BCS) $\Pi_t = \mathbb{P}(S_t, H^{1:N}_t | H^0_t)$, which is the conditional distribution on the state and the FPSs given the FCS. It is shown that restricting attention to coordinator's policy of the form $\dot{d}_t : \Omega(\Pi_t) \rightarrow \Omega(\Gamma_t)$ is without loss of optimality. The DP presented thus uses this BCS as the state and an FPS-based prescription as the action, which is presented in Algorithm 4.2.

**Algorithm 4.2** Dynamic Programming with BCSs and FPS-based Prescriptions

```
$\dot{V}_{T+1}(\pi_{T+1}) \triangleq 0$
for $t = T, \dots, 1$ do
    $\dot{Q}_t(\pi_t, \gamma_t) = \mathbb{E}\left[R(S_t, \Gamma_t(H^{1:N}_t)) + \dot{V}_{t+1}(\eta_t(\Pi^0_t, \Gamma_t, O^0_{t+1}))|\Pi_t = \pi_t, \Gamma_t = \gamma_t\right]$
    $\dot{V}_t(\pi_t) = \max_{\gamma_t \in \Omega(\Gamma_t)} \dot{Q}_t(\pi_t, \gamma_t)$
```

Note the BCS is updated through *Bayesian update* with the function $\eta_t$ [88].

There are two problems with this approach when applied to the MARL setting. First, the BCS is updated via a Bayesian update using $\mathbb{P}_T$ and $\mathbb{P}_O$, which requires model knowledge. Second, the growing length of $H^{1:N}_t$ makes the spaces of $\Pi_t$ and $\Gamma_t$ explosively large and impossible to explore. Specifically, the space $\Omega(\Pi_t) = \Delta(\mathcal{S} \times \Omega(H^{1:N}_t))$ grows linearly in time, and the space $\Omega(\Gamma_t) = \prod_{n=1}^{N} \Omega(\Omega(H^n_t) \rightarrow \Omega(A^n_t))$ grows doubly exponentially in time. At a conceptual level we can apply the AIS framework to the centralized POMDP of the coordinator[11]; however, the underlying decentralized information structure coupled with increasing domain of private information makes practical implementations of this scheme challenging.

#### 4.2.4.4 DP with BCS and SPI

To alleviate the aforementioned dimensionality issue, [117] further compresses the FPS to a representation called the sufficient private information (SPI) lying in a time-

---

[11] Strictly speaking, this requires a straightforward extension to time-varying action spaces for different time steps – see [113] Section 5 for details.

invariant domain. They identify a set of conditions for the compression so that the SPI is sufficient for decision making purposes, just like conditions of $Z_t$ in Definition 4.1 and Definition 4.2.

**Definition 4.5:** *A sufficient private information (SPI)* $Z_t^{1:N}$ *is a tuple of outputs of a set of functions* $Z_t^n = \vartheta_t^n(H_t^0, H_t^n)$ *for all* $n \in [N]$ *satisfying the following properties:*

***(SPI1)*** *It evolves recursively* $Z_t^n = \phi_t^n(Z_{t-1}^n, \Gamma_{t-1}, O_t^0, A_{t-1}^n, O_t^n, g_{1:t-1}) \ \forall \ n$.

***(SPI2)*** *It suffices for performance evaluation for SPI-based policies*

$$\mathbb{E}[R(S_t, A_t)|h_t^0, h_t^n, a_t] = \mathbb{E}[R(S_t, A_t)|h_t^0, z_t^n, a_t] \ \forall \ n, h_t^0, h_t^n, a_t.$$

***(SPI3)*** *It suffices for predicting itself and the common observation*

$$\mathbb{P}(Z_{t+1}^{1:N}, O_{t+1}^0|h_t^0, h_t^{1:N}, \gamma_t, a_t) = \mathbb{P}(Z_{t+1}^{1:N}, O_{t+1}^0|h_t^0, z_t^{1:N}, \gamma_t, a_t) \ \forall \ h_t^{0:N}, \gamma_t, a_t.$$

***(SPI4)*** *It suffices for predicting other agents' SPI for SPI-based policies*

$$\mathbb{P}(Z_t^{-n}|h_t^0, h_t^n) = \mathbb{P}(Z_t^{-n}|h_t^0, z_t^n) \ \forall \ n, h_t^0, h_t^n.$$

In the conditions, (SPI2) and (SPI4) hold for SPI-based policies; this means the equalities only need to hold when "SPI-based prescriptions," which are explained in the following, are executed throughout. The coordinator now considers SPI-based prescriptions $\Lambda_t = \Lambda_t^{1:N}$ where $A_t^n = \Lambda_t^n(Z_t^n)$, and the BCS is changed to $\tilde{\Pi}_t = \mathbb{P}^{\tilde{g}_{1:t-1}}(S_t, Z_t^{1:N}|C_t^0) = \mathbb{P}(S_t, Z_t^{1:N}|\tilde{H}_t^0)$, where agent $n$'s policy is now of the form $A_t^n = \tilde{g}_t^n(\tilde{\Pi}_t, Z_t^n)$ and coordinator's AOH is $\tilde{H}_t^0 = (O_1^0, \Lambda_1, \ldots, O_t^0)$. It is shown in [117] that restricting attention to coordinator's policy of the form $\tilde{d}_t : \Omega(\tilde{\Pi}_t) \to \Omega(\Lambda_t)$ is without loss of optimality. The resulting DP uses BCS (now a belief on true state and SPI) as the state and SPI-based prescription as the action given in Algorithm 4.3. Note that the compression actually leads to an action compression for the coordinator – from FPS-based prescriptions to SPI-based prescriptions – which has no loss in performance.

**Algorithm 4.3** Dynamic Programming with BCSs and SPI-based Prescriptions

```
$\tilde{V}_{T+1}(\tilde{\pi}_{T+1}) \triangleq 0$
for $t = T, \ldots, 1$ do
    $\tilde{Q}_t(\tilde{\pi}_t, \lambda_t) = \mathbb{E}\left[R(S_t, \Lambda_t(Z_t^{1:N})) + \tilde{V}_{t+1}(\tilde{\eta}_t(\tilde{\Pi}_t, \Lambda_t, O_{t+1}^0))|\tilde{\Pi}_t = \tilde{\pi}_t, \Lambda_t = \lambda_t\right]$
    $\tilde{V}_t(\tilde{\pi}_t) = \max_{\lambda_t \in \Omega(\Lambda_t)} \tilde{Q}_t(\tilde{\pi}_t, \lambda_t)$
```

With $\tilde{\Pi}_t$, $Z_t^{1:N}$, and $\Lambda_t$ all lying in time-invariant spaces, the complexity no longer grows with time. However, it is unclear how to find mappings satisfying Definition 4.5

and update the BCSs in an MARL setting. Further, the solution focuses on a decentralized setting wherein the (lossless) compression functions are common knowledge, and the performance assessments and predictions are based only on the information of any particular agent (in (SPI2) and (SPI4)). Ensuring these properties in the MARL context would require significant communication, particularly during searching the compression functions.

## 4.3 General Common and Private State Representations

We seek to extend the idea of identifying representations sufficient for approximately optimal decision making from Section 4.2.3 to the multi-agent setting, and develop a general compression framework for common states and private states (hence also prescriptions) whose mappings can be learned from samples obtained by interacting with the environment alone.

The results in [117] imply that the BCS along with SPI is sufficient for DP purposes; in other words, one may obtain the same optimal value function if one runs the DP with BCSs and SPI-based prescriptions. The proof provided in [117], however, is by showing that $Z_t^n$ with the BCS is an IS for agent $n$ so that one can restrict attention to SPI-based strategies, and is not helpful for extending the concept to the approximate cases. Moreover, the BCS is also impossible to compute when the model is unknown. In this section, we present a DP-based proof of the same result – we show that the DP in Algorithm 4.1 is equivalent to the DP in Algorithm 4.3, with our own definition of the private state representation, which we call the sufficient private state (SPS). The relations of our definition with SPI proposed in [117] are discussed in Section **??**. Following the idea of the IS in Definition 4.1 and Definition 4.2, we propose a more general common state representation called the sufficient common state (SCS), which includes BCS as one of its special case.

We will first introduce the *supervisor's perspective* in Section 4.3.1, which is a key tool in the analysis of the rest of the chapter. We show that the compression from FPS to our proposed SPS, which induces an action compression from FPS-based prescriptions to SPS-based prescriptions, is without loss of optimality in Section 4.3.2. Then based on SPS-based prescriptions, we further compress FCS to SCS and show that doing so is without loss of optimality as well in Section 4.3.3.

### 4.3.1 Supervisor's Functions

For better exposition, we introduce another set of $Q/V$ functions from an omniscient *supervisor*'s perspective, for the original decision problem. The supervisor can access the *union* of the information of all agents: at time $t$ the supervisor knows $H_t^{0:N}$. In contrast, coordinator's information is the *intersection* of the information of all agents: $H_t^0$. The supervisor, however, only observes what is happening, lets the coordinator decide all the policies and prescriptions, and implements the coordinator's policies. Let $d_{1:T}^*$ be a coordinator's optimal policy solved using Algorithm 4.1, i.e. $d_t^*(h_t^0) \in \operatorname{argmax}_{\gamma_t \in \Omega(\Gamma_t)} Q_t(h_t^0, \gamma_t)$. Then the $Q/V$ functions defined in Algorithm 4.1 can be rewritten as

$$Q_t(h_t^0, \gamma_t) = \mathbb{E}\left[\sum_{\tau=t}^{T} R_\tau \middle| h_t^0, \gamma_t, d_{t+1:T}^*\right], \tag{4.7}$$

$$V_t(h_t^0) = \mathbb{E}\left[\sum_{\tau=t}^{T} R_\tau \middle| h_t^0, d_{t:T}^*\right]. \tag{4.8}$$

In other words, given $h_t^0$, the coordinator's $V$ function is the cumulative reward of executing the optimal policy from $t$ on, and its $Q$ function given the prescription $\gamma_t$ is the cumulative reward of performing $\gamma_t$ first then executing the optimal policy from $t+1$ on. The supervisor's $Q/V$ functions use similar concepts, but with *supervisor's states* and *coordinator's optimal policies.*

**Definition 4.6:** *For any $h_t^{0:N} \in \Omega(H_t^{0:N})$, $\gamma_t \in \Omega(\Gamma_t)$, define the supervisor's $Q$ function as*

$$\begin{aligned} Q_t^S(h_t^0, h_t^{1:N}, \gamma_t) &\triangleq \mathbb{E}\left[\sum_{\tau=t}^{T} R_\tau \middle| h_t^0, h_t^{1:N}, \gamma_t, d_{t+1:T}^*\right] \\ &= \mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N})) + V_{t+1}^S((H_t^0, \Gamma_t, O_{t+1}^0), H_{t+1}^{1:N}) | h_t^0, h_t^{1:N}, \gamma_t\right], \end{aligned} \tag{4.9}$$

*and the supervisor's* $V$ *function as*

$$V_t^S(h_t^0, h_t^{1:N}) \triangleq \mathbb{E}\left[\sum_{\tau=t}^{T} R_\tau \middle| h_t^0, h_t^{1:N}, d_{t:T}^*\right] = Q_t^S(h_t^0, h_t^{1:N}, \gamma_t^*), \tag{4.10}$$

*where* $\gamma_t^* \in \operatorname{argmax}_{\gamma_t \in \Omega(\Gamma_t)} Q_t(h_t^0, \gamma_t)$. *The supervisor's* $Q$ *function and* $V$ *function are only defined when the FPS* $h_t^{1:N}$ *is* admissible *under* $h_t^0$, *i.e.* $\mathbb{P}(h_t^{1:N}|h_t^0) > 0$. *We denote the space of admissible FPSs in* $\Omega(H_t^{1:N})$ *under* $h_t^0$ *as* $\mathfrak{H}_t(h_t^0)$.

Then the coordinator's $Q/V$ functions can be expressed as the expectation of supervisor's $Q/V$ functions taken over the conditional distribution on FPSs given the FCS:

$$\begin{aligned} Q_t(h_t^0, \gamma_t) &= \sum_{h_t^{1:N} \in \mathfrak{H}_t(h_t^0)} \mathbb{P}(h_t^{1:N}|h_t^0) \mathbb{E}\left[\sum_{\tau=t}^{T} R_\tau \middle| h_t^0, h_t^{1:N}, \gamma_t, d_{t+1:T}^*\right] \\ &= \sum_{h_t^{1:N} \in \mathfrak{H}_t(h_t^0)} \mathbb{P}(h_t^{1:N}|h_t^0) Q_t^S(h_t^0, h_t^{1:N}, \gamma_t), \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} V_t(h_t^0) &= \sum_{h_t^{1:N} \in \mathfrak{H}_t(h_t^0)} \mathbb{P}(h_t^{1:N}|h_t^0) \mathbb{E}\left[\sum_{\tau=t}^{T} R_\tau \middle| h_t^0, h_t^{1:N}, d_{t:T}^*\right] \\ &= \sum_{h_t^{1:N} \in \mathfrak{H}_t(h_t^0)} \mathbb{P}(h_t^{1:N}|h_t^0) V_t^S(h_t^0, h_t^{1:N}). \end{aligned} \tag{4.12}$$

Throughout the rest of the chapter, we assume that only admissible FPSs are considered, so that we use the convention that "for any $h \in \Omega(H_t^{1:N})$" means "for any $h \in \mathfrak{H}_t(h_t^0)$ when $h_t^0$ is clear in the context."

### 4.3.2 Compressing Private States

In this subsection, we give an alternative proof of the DP equivalence between Algorithm 4.1, the no compression scheme, and a revised Algorithm 4.3 with BCSs and SPS-prescriptions, where the SPS is the compressed private state defined in Definition 4.7 in the following. This implies that the compression to SPS is without loss of optimality.

**Definition 4.7:** *A sufficient private state (SPS)* $Z_t^{1:N}$ *is a tuple of outputs of a set of functions* $Z_t^n = \vartheta_t^n(H_t^0, H_t^n)$ *for all* $n \in [N]$ *that satisfies the following properties:*

***(SPS1)*** *It evolves recursively* $Z_t^n = \phi_t^n(Z_{t-1}^n, \Gamma_{t-1}, O_t^0, O_t^n) \ \forall \ n$.

***(SPS2)*** *It is sufficient for performance evaluation*

$$\mathbb{E}[R(S_t, A_t)|h_t^0, h_t^{1:N}, a_t] = \mathbb{E}[R(S_t, A_t)|h_t^0, z_t^{1:N}, a_t] \ \forall \ h_t^{0:N}, a_t.$$

***(SPS3)*** *It is sufficient for predicting observations*

$$\mathbb{P}(O_{t+1}^{0:N}|h_t^0, h_t^{1:N}, a_t) = \mathbb{P}(O_{t+1}^{0:N}|h_t^0, z_t^{1:N}, a_t) \ \forall \ h_t^{0:N}, a_t.$$

**Remark 4.8:** *For (SPS1), in our DP equivalence proof we only require* $Z_t^{1:N} = \phi_t^{1:N}(Z_{t-1}^{1:N}, \Gamma_{t-1}, O_t^{0:N})$, *that is, jointly the SPS evolves recursively; but for fully decentralized execution, it has to be updated component-wise recursively. Note that* $Z_t^n = \phi_t^n(Z_{t-1}^n, \Gamma_{t-1}, O_t^0, O_t^n)$ *implies* $Z_t^{1:N} = \phi_t^{1:N}(Z_{t-1}^{1:N}, \Gamma_{t-1}, O_t^{0:N})$ *as we define* $\phi_t^{1:N}(Z_{t-1}^{1:N}, \Gamma_{t-1}, O_t^{0:N}) \triangleq (\phi_t^1(Z_{t-1}^1, \Gamma_{t-1}, O_t^0, O_t^1), \ldots, \phi_t^N(Z_{t-1}^N, \Gamma_{t-1}, O_t^0, O_t^N))$.

The compression is characterized by functions $\vartheta_t^{1:N}$. These functions also relate $\Omega(\Gamma_t)$ and $\Omega(\Lambda_t)$ as SPS-based prescriptions are a strict subset of FPS-based prescritions, which will be explained next. Note that here the conditions we set for the action compression from $\Omega(\Gamma_t)$ to $\Omega(\Lambda_t)$ are on the private states instead of defining an encapsulation directly on the actions (i.e., prescriptions); moreover, the compression may depend on the common state $h_t^0$ as well. Hence, this falls outside of the action compression scheme studied in [113].

We now determine the relationship between the space of FPS-based prescriptions $\Omega(\Gamma_t)$ and the space of SPS-based prescriptions $\Omega(\Lambda_t)$. Consider a fixed $h_t^0$. Since the compression mappings $Z_t^{1:N} = \vartheta_t^{1:N}(H_t^0, H_t^{1:N})$ are functions, there could be multiple $h_t^{1:N}$'s that are mapped to the same $z_t^{1:N}$. A $\lambda_t \in \Omega(\Lambda_t)$ can thus be thought of as a special element of $\Omega(\Gamma_t)$ that prescribes the same action for all the FPSs $h_t^{1:N}$'s mapped to the same SPS $z_t^{1:N}$. Hence, we can construct an injective *extension mapping* from $\Omega(\Lambda_t)$ to $\Omega(\Gamma_t)$ in this sense.

**Definition 4.9:** *For any* $h_t^0 \in \Omega(H_t^0)$, *define the extension mapping* $\psi_t : \Omega(\Lambda_t) \times \Omega(H_t^0) \to \Omega(\Gamma_t)$ *as follows: for any* $h_t^{1:N}$ *and* $\lambda_t$, $\gamma_t = \psi_t(\lambda_t, h_t^0)$ *will first compress* $h_t^{1:N}$ *to* $z_t^{1:N} = \vartheta_t^{1:N}(h_t^0, h_t^{1:N})$ *(hence* $\psi_t$ *implicitly depends on* $\vartheta_t$*), then choose the action according to* $\lambda_t(z_t^{1:N})$. *That is,*

$$\gamma_t(h_t^{1:N}) = \psi_t(\lambda_t, h_t^0)(h_t^{1:N}) \triangleq \lambda_t(\vartheta_t^{1:N}(h_t^0, h_t^{1:N})).$$

*Given the compression* $\vartheta_t^{1:N}$*,* $\psi_t$ *is well-defined. Under this circumstance and with an abuse of notation,* $\gamma_t = \psi_t(\lambda_t, h_t^0)$ *will be written as* $\gamma_{\lambda_t, h_t^0}$ *when the considered compression is clear from the context and will be referred to as the* $\gamma_t$ *extended from* $\lambda_t$ *under* $h_t^0$*.*

The following proposition says for any FCS one can find an SPS-based prescription whose extension achieves the same $Q$-value as an optimal prescription. This implies that it suffices to consider the class of prescriptions extended from SPS-based prescriptions for DP purposes.

**Proposition 4.10:** *For any* $h_t^0 \in \Omega(H_t^0)$*, assume* $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$ *is an optimal prescription. Then there exists a* $\lambda \in \Omega(\Lambda_t)$ *such that*

$$Q_t(h_t^0, \gamma^*) = Q_t(h_t^0, \gamma_{\lambda, h_t^0}), \tag{4.13}$$

*which leads to*

$$V_t(h_t^0) = \max_{\gamma \in \Omega(\Gamma_t)} Q_t(h_t^0, \gamma) = \max_{\lambda \in \Omega(\Lambda_t)} Q_t(h_t^0, \gamma_{\lambda, h_t^0}). \tag{4.14}$$

Before proving the proposition we need a few intermediate results. The first key lemma says that with the same supervisor's state, the supervisor's $Q$-values for two different prescriptions will be the same as long as they prescribe the same action for the given PI.

**Lemma 4.11:** *For any* $h_t^0 \in \Omega(H_t^0)$ *and* $h \in \Omega(H_t^{1:N})$*, let* $\gamma_1, \gamma_2 \in \Omega(\Gamma_t)$ *be two prescriptions that choose the same action on the FPS* $h$*, i.e.* $\gamma_1(h) = \gamma_2(h) = a$*. Then*

$$Q_t^S(h_t^0, h, \gamma_1) = Q_t^S(h_t^0, h, \gamma_2). \tag{4.15}$$

**Proof:** See Appendix C.1. ■

Next we show that the supervisor's $Q$-values are also the same for two different FPSs that map to the same SPS and a prescription that prescribes the same action on these two FPSs.

**Lemma 4.12:** *For any* $h_t^0 \in \Omega(H_t^0)$*, let* $h_1, h_2 \in \Omega(H_t^{1:N})$ *be two FPSs under* $h_t^0$ *that map to the same SPS* $z \in \Omega(Z_t^{1:N})$*, i.e.* $z = \vartheta_t^{1:N}(h_t^0, h_1) = \vartheta_t^{1:N}(h_t^0, h_2)$*, and* *let* $\gamma \in \Omega(\Gamma_t)$ *be a prescription that chooses the same action on these two FPSs* $\gamma(h_1) = \gamma(h_2) = a$. *Then*

$$Q_t^S(h_t^0, h_1, \gamma) = Q_t^S(h_t^0, h_2, \gamma). \tag{4.16}$$

**Proof:** See Appendix C.1. ■

Using the above two lemmas, we show that the supervisor's $V$ function will be the same for two supervisor's states with the same compression of private states.

**Corollary 4.13:** *For any* $h_t^0 \in \Omega(H_t^0)$, *let* $h_1, h_2 \in \Omega(H_t^{1:N})$ *be two FPSs under* $h_t^0$ *that map to the same SPS* $z \in \Omega(Z_t^{1:N})$, *i.e.* $z = \vartheta_t^{1:N}(h_t^0, h_1) = \vartheta_t^{1:N}(h_t^0, h_2)$, *and let* $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$ *be an optimal prescription. Then we have*

$$V_t^S(h_t^0, h_1) - V_t^S(h_t^0, h_2) \triangleq Q_t^S(h_t^0, h_1, \gamma^*) - Q_t^S(h_t^0, h_2, \gamma^*) = 0. \tag{4.17}$$

**Proof:** See Appendix C.1. ■

We are now ready to prove Proposition 4.10.

**Proof of Proposition 4.10:** Given an optimal prescription $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$, we will specifically construct a $\lambda \in \Omega(\Lambda_t)$ such that $Q_t(h_t^0, \gamma^*) = Q_t(h_t^0, \gamma_{\lambda, h_t^0})$. For each $z \in \Omega(Z_t^{1:N})$, define

$$\mathcal{H}_z = \left\{ h \in \Omega(H_t^{1:N}) : \vartheta_t^{1:N}(h_t^0, h) = z \right\} \tag{4.18}$$

to be the class of $h$'s in $\Omega(H_t^{1:N})$ that are compressed to $z$ under the considered compression $\vartheta_t^{1:N}$. By the Axiom of Choice, for each $z \in \Omega(Z_t^{1:N})$ there exists a representative of the class $\mathcal{H}_z$ coming from arbitrary choice function, which we denote as $\bar{h}_z$ (or we do not really have to use the Axiom of Choice since all the classes $\mathcal{H}_z$ considered here are finite). We then construct the $\lambda$ by $\lambda(z) = \gamma^*(\bar{h}_z)$; the corresponding extension in $\Omega(\Gamma_t)$ will be

$$\gamma_{\lambda, h_t^0}(h) = \gamma^* \left( \bar{h}_{\vartheta_t^{1:N}(h_t^0, h)} \right) \qquad \forall \ h \in \Omega(H_t^{1:N}), \tag{4.19}$$

that is, the prescription first compresses the input FPS and finds the representative of the corresponding compression class, then it mimics what the optimal prescription would have done with the representative. For any $h \in \Omega(H_t^{1:N})$, we have

$$\begin{aligned}
&\left|Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, h, \gamma_{\lambda, h_t^0})\right| \\
\leq &\left|Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, \bar{h}_{\vartheta_t^{1:N}(h_t^0, h)}, \gamma^*)\right| \\
&\quad + \left|Q_t^S(h_t^0, \bar{h}_{\vartheta_t^{1:N}(h_t^0, h)}, \gamma^*) - Q_t^S(h_t^0, \bar{h}_{\vartheta_t^{1:N}(h_t^0, h)}, \gamma_{\lambda, h_t^0})\right| \\
&\quad + \left|Q_t^S(h_t^0, \bar{h}_{\vartheta_t^{1:N}(h_t^0, h)}, \gamma_{\lambda, h_t^0}) - Q_t^S(h_t^0, h, \gamma_{\lambda, h_t^0})\right| \\
= &\, 0,
\end{aligned}$$

as the first term is 0 due to Corollary 4.13, the second term is 0 due to Lemma 4.11, and the third term is also 0 due to Lemma 4.12. Taking the conditional expectation on $h$ (only over admissible $h$'s) given $h_t^0$, we obtain

$$\begin{aligned}
&\left|Q_t(h_t^0, \gamma^*) - Q_t(h_t^0, \gamma_{\lambda, h_t^0})\right| \\
= &\left|\sum_h \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) - \sum_h \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma_{\lambda, h_t^0})\right| \\
\leq &\sum_h \mathbb{P}(h|h_t^0) \left|Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, h, \gamma_{\lambda, h_t^0})\right| = 0.
\end{aligned}$$

■

Proposition 4.10 effectively restricts one's attention to SPS-based prescriptions from the bigger class of FPS-based prescriptions without loss of optimality. Now based on that, we further compress FCSs to BCSs while maintaining the DP outcomes.

**Proposition 4.14:** *Let* $h_1^0, h_2^0 \in \Omega(H_t^0)$ *be two FCSs. If* $\tilde{\Pi}_t(h_1^0) = \tilde{\Pi}_t(h_2^0)$*, i.e.* $\mathbb{P}(S_t, Z_t^{1:N}|h_1^0) = \mathbb{P}(S_t, Z_t^{1:N}|h_2^0)$*, then for any* $\lambda \in \Omega(\Lambda_t)$*,*

$$Q_t(h_1^0, \gamma_{\lambda, h_1^0}) = Q_t(h_2^0, \gamma_{\lambda, h_2^0}). \tag{4.20}$$

**Proof:** See Appendix C.1. ■

Combining Proposition 4.10 and Proposition 4.14, we can now conclude the Algorithm 4.1 and Algorithm 4.3 are equivalent in terms of the DP quantities.

**Theorem 4.15:** *For any* $h_t^0 \in \Omega(H_t^0)$ *and* $\lambda \in \Omega(\Lambda_t)$, *we have*

$$Q_t(h_t^0, \gamma_{\lambda,h_t^0}) = \tilde{Q}_t(\tilde{\Pi}_t(h_t^0), \lambda) \tag{4.21}$$

*and*

$$V_t(h_t^0) = \tilde{V}_t(\tilde{\Pi}_t(h_t^0)). \tag{4.22}$$

**Proof:** The result follows directly from Proposition 4.10 and Proposition 4.14. The result in Proposition 4.10 establishes the implication from (4.21) to (4.22) in the claim, as

$$V_t(h_t^0) = \max_{\lambda \in \Omega(\Lambda_t)} Q_t(h_t^0, \gamma_{\lambda,h_t^0}) = \max_{\lambda \in \Omega(\Lambda_t)} \tilde{Q}_t(\tilde{\Pi}_t(h_t^0), \lambda) = \tilde{V}_t(\tilde{\Pi}_t(h_t^0)).$$

The proof of Proposition 4.14 establishes that

$$\begin{aligned}
&\mathbb{E}\left[R_t(S_t, A_t)|\tilde{\Pi}_t(h_t^0), \gamma_{\lambda,\tilde{\Pi}_t(h_t^0)}\right] \\
&= \sum_{\hat{h}_t^0 : \tilde{\Pi}_t(\hat{h}_t^0) = \tilde{\Pi}_t(h_t^0)} \mathbb{P}(\hat{h}_t^0 | \tilde{\Pi}_t(h_t^0)) \mathbb{E}[R_t(S_t, A_t)|\hat{h}_t^0, \gamma_{\lambda,\hat{h}_t^0}] \\
&= \sum_{\hat{h}_t^0 : \tilde{\Pi}_t(\hat{h}_t^0) = \tilde{\Pi}_t(h_t^0)} \mathbb{P}(\hat{h}_t^0 | \tilde{\Pi}_t(h_t^0)) \mathbb{E}[R_t(S_t, A_t)|h_t^0, \gamma_{\lambda,h_t^0}] \\
&= \mathbb{E}[R_t(S_t, A_t)|h_t^0, \gamma_{\lambda,h_t^0}],
\end{aligned}$$

and that

$$\begin{aligned}
&\mathbb{E}[\tilde{V}_{t+1}(\tilde{\Pi}_{t+1}(H_{t+1}^0))|\tilde{\Pi}_t(h_t^0), \lambda] = \mathbb{E}\left[V_{t+1}(H_{t+1}^0)|\tilde{\Pi}_t(h_t^0), \gamma_{\lambda,\tilde{\Pi}_t(h_t^0)}\right] \\
&= \sum_{\hat{h}_t^0 : \tilde{\Pi}_t(\hat{h}_t^0) = \tilde{\Pi}_t(h_t^0)} \mathbb{P}(\hat{h}_t^0 | \tilde{\Pi}_t(h_t^0)) \mathbb{E}[V_{t+1}(H_{t+1}^0)|\hat{h}_t^0, \gamma_{\lambda,\hat{h}_t^0}] \\
&= \sum_{\hat{h}_t^0 : \tilde{\Pi}_t(\hat{h}_t^0) = \tilde{\Pi}_t(h_t^0)} \mathbb{P}(\hat{h}_t^0 | \tilde{\Pi}_t(h_t^0)) \mathbb{E}[V_{t+1}(H_{t+1}^0)|h_t^0, \gamma_{\lambda,h_t^0}] \\
&= \mathbb{E}[V_{t+1}(H_{t+1}^0)|h_t^0, \gamma_{\lambda,h_t^0}].
\end{aligned}$$

Using a mathematical induction argument with the two equalities validates (4.21). ■

The same DP equivalence proof presented in Theorem 4.15 will hold between the

DP in Algorithm 4.1 and the DP with FCS as the state and SPS-based prescription as the action, with $\widetilde{\Pi}_t(h_t^0)$ replaced by $h_t^0$ (so that the probability $\mathbb{P}(\hat{h}_t^0|h_t^0) = \mathbb{I}\{\hat{h}_t^0 = h_t^0\}$) is simply an indicator function). We will see that both DP equivalences are special cases of Theorem 4.18, where we propose a general lossless compression for the common states.

### 4.3.3 Compressing Common States

Adopting BCSs also requires the model information when performing the Bayesian updates. We hence look for a more general representation of the common state. Under the scheme of SPS-based prescriptions, which form the action space in this context, the generalization is an extension of the result in [112].

**Definition 4.16:** *A sufficient common state (SCS) $Z_t^0$ is the output of a function $Z_t^0 = \vartheta_t^0(H_t^0)$ that satisfies the following properties:*

***(SCS1)*** *It evolves recursively $Z_t^0 = \phi_t^0(Z_{t-1}^0, \Lambda_{t-1}, O_t^0)$.*

***(SCS2)*** *It suffices for performance evaluation*

$$\mathbb{E}[R_t(S_t, A_t)|h_t^0, \lambda_t] = \mathbb{E}[R_t(S_t, A_t)|\vartheta_t^0(h_t^0), \lambda_t] \;\forall\; h_t^0, \lambda_t.$$

***(SCS3)*** *It suffices for predicting the common observation*

$$\mathbb{P}(O_{t+1}^0|h_t^0, \lambda_t) = \mathbb{P}(O_{t+1}^0|\vartheta_t^0(h_t^0), \lambda_t) \;\forall\; h_t^0, \lambda_t.$$

In Algorithm 4.4, SCS serves as the common state and SPS serves as the private state. In the DP, the states are SCSs and the actions are SPS-based prescriptions.

**Algorithm 4.4** Dynamic Programming with SCSs and SPS-based Prescriptions

```
$\breve{V}_{T+1}(z_{T+1}^0) \triangleq 0$
for $t = T, \ldots, 1$ do
    $\breve{Q}_t(z_t^0, \lambda_t) = \mathbb{E}\left[R(S_t, \Lambda_t(Z_t^{1:N})) + \breve{V}_{t+1}(\phi_t^0(Z_t^0, \Lambda_t, O_{t+1}^0))|Z_t^0 = z_t^0, \Lambda_t = \lambda_t\right]$
    $\breve{V}_t(z_t^0) = \max_{\lambda_t \in \Omega(\Lambda_t)} \breve{Q}_t(z_t^0, \lambda_t)$
```

With these general representations, the DP is still equivalent to the original DP without any compression in Algorithm 4.1.

**Proposition 4.17:** *Let $h_1^0, h_2^0 \in \Omega(H_t^0)$ be two FCSs. If $\vartheta_t^0(h_1^0) = \vartheta_t^0(h_2^0)$, then for any $\lambda \in \Omega(\Lambda_t)$,*

$$Q_t(h_1^0, \gamma_{\lambda, h_1^0}) = Q_t(h_2^0, \gamma_{\lambda, h_2^0}).$$

**Proof:** See Appendix C.2. ■

**Theorem 4.18:** *For any* $h_t^0 \in \Omega(H_t^0)$ *and* $\lambda \in \Omega(\Lambda_t)$, *we have*

$$Q_t(h_t^0, \gamma_{\lambda, h_t^0}) = \breve{Q}_t(\vartheta_t^0(h_t^0), \lambda) \tag{4.23}$$

*and*

$$V_t(h_t^0) = \breve{V}_t(\vartheta_t^0(h_t^0)). \tag{4.24}$$

**Proof:** The result follows directly from Proposition 4.10 and Proposition 4.17. The exact proof is the same as that of Theorem 4.15. ■

Note that BCS is actually a special case of SCS, so that Proposition 4.17 and Theorem 4.18 imply Proposition 4.14 and Theorem 4.15, respectively. We proved Proposition 4.14 and Theorem 4.15 directly to provide more intuition. For more details see Section 4.5.1.

## 4.4 General Common and Private Approximate State Representations

The theory developed in Section 4.3 does not guarantee the existences of such SCS and SPS compression mappings, nor does it provide a way to find such mappings if they exist[12]. Following the concept of the AIS framework in [112], we extend the theory in Section 4.3 to its approximate correspondence, so that it is more applicable to designing MARL algorithms without the model knowledge.

In this section, we propose our general states representation framework for approximate planning and control in partially observable MARL problems. We start by compressing private histories to ASPS in Section 4.4.1; for the coordinator, this induces an action compression from FPS-based prescriptions to ASPS-based prescriptions. Then based on this compression, the common history is further compressed to ASCS in Section 4.4.2. The results in Section 4.3 are nearly in parallel with and are special cases of the results developed in this section with error parameters being 0

---

[12] In the literature, e.g. [117, 116, 87], various examples of SCS and SPS are provided, but usually for some particular information structures.

(see Definition 4.19 and Definition 4.25), just as the relation between IS and AIS in Section 4.2.3.

The framework we develop will be consistent with the "centralized training with distributed execution" philosophy of recent empirical MARL work wherein there is a centralized agent called the supervisor (Section 4.3.1). The supervisor observes all the quantities and develops good compression of PI and CI that the coordinator can use to produce close-to-optimal prescriptions (using the compressed CI), which can be implemented by the agents using just their own compressed PI.

### 4.4.1 Compressing Private States

We start with the notion of an approximate sufficient private state (ASPS), which is a private representation that satisfies the conditions of a SPS approximately. We then provide a DP with the FCS as the state and the ASPS-based prescription as the action in Algorithm 4.5, and construct "approximate DP equivalence" by bounding the optimality gap of such a DP as in [112].

**Definition 4.19:** *An $(\epsilon_p, \delta_p)$-approximate sufficient private state (ASPS) $\widehat{Z}_t^{1:N}$ is a tuple of outputs of a set of functions $\widehat{Z}_t^n = \widehat{\vartheta}_t^n(H_t^0, H_t^n)$ for all $n \in [N]$ satisfying:*

***(ASPS1)*** *It evolves recursively $\widehat{Z}_t^n = \widehat{\phi}_t^n(\widehat{Z}_{t-1}^n, \Gamma_{t-1}, O_t^0, O_t^n)$ $\forall$ $n$.*

***(ASPS2)*** *It suffices for approximate performance evaluation*

$$\left|\mathbb{E}[R(S_t, A_t)|h_t^0, h_t^{1:N}, a_t] - \mathbb{E}[R(S_t, A_t)|h_t^0, \widehat{z}_t^{1:N}, a_t]\right| \le \epsilon_p/4 \;\forall\; h_t^{0:N}, a_t.$$

***(ASPS3)*** *It suffices for approximately predicting observations*

$$\mathcal{K}\left(\mathbb{P}(O_{t+1}^{0:N}|h_t^0, h_t^{1:N}, a_t), \mathbb{P}(O_{t+1}^{0:N}|h_t^0, \widehat{z}_t^{1:N}, a_t)\right) \le \delta_p/8 \;\forall\; h_t^{0:N}, a_t.$$

**Remark 4.20:** *Again, for (ASPS1), in our approximate DP equivalence proof we only require $\widehat{Z}_t^{1:N} = \widehat{\phi}_t^{1:N}(\widehat{Z}_{t-1}^{1:N}, \Gamma_{t-1}, O_t^{0:N})$, that is, jointly the SPS evolves recursively; but for fully decentralized execution it has to be updated component-wise recursively. Note that $\widehat{Z}_t^n = \widehat{\phi}_t^n(\widehat{Z}_{t-1}^n, \Gamma_{t-1}, O_t^0, O_t^n)$ implies $\widehat{Z}_t^{1:N} = \widehat{\phi}_t^{1:N}(\widehat{Z}_{t-1}^{1:N}, \Gamma_{t-1}, O_t^{0:N})$ as we define $\widehat{\phi}_t^{1:N}(\widehat{Z}_{t-1}^{1:N}, \Gamma_{t-1}, O_t^{0:N}) \triangleq (\widehat{\phi}_t^1(\widehat{Z}_{t-1}^1, \Gamma_{t-1}, O_t^0, O_t^1), \ldots, \widehat{\phi}_t^N(\widehat{Z}_{t-1}^N, \Gamma_{t-1}, O_t^0, O_t^N))$.*

With this definition, an ASPS-based prescription is a mapping $\widehat{\lambda}_t : \Omega(\widehat{Z}_t^{1:N}) \to \Omega(A_t)$ that prescribes the joint actions for all ASPSs $A_t = \widehat{\lambda}_t(\widehat{Z}_t^{1:N})$, of course, in a component-wise manner $\widehat{\lambda}_t(\widehat{Z}_t^{1:N}) \triangleq (\widehat{\lambda}_t^1(\widehat{Z}_t^1), \ldots, \widehat{\lambda}_t^N(\widehat{Z}_t^N))$.

**Algorithm 4.5** Dynamic Programming with FCSs and ASPS-based Prescriptions

```
$\widehat{V}_{T+1}(h^0_{T+1}) \triangleq 0$
for $t = T, \dots, 1$ do
    $\widehat{Q}_t(h^0_t, \widehat{\lambda}_t) = \mathbb{E}\left[R_t(S_t, \widehat{\lambda}_t(\widehat{Z}^{1:N}_t)) + \widehat{V}_{t+1}((H^0_t, \widehat{\Lambda}_t, O^0_{t+1}))|H^0_t = h^0_t, \widehat{\Lambda}_t = \widehat{\lambda}_t\right]$
    $\widehat{V}_t(h^0_t) = \max_{\widehat{\lambda}_t \in \Omega(\widehat{\Lambda}_t)} \widehat{Q}_t(h^0_t, \widehat{\lambda}_t)$
```

**Proposition 4.21** (Extension of Proposition 4.10)**:** *Assume the reward function $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. For any $h^0_t \in \Omega(H^0_t)$ and $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h^0_t, \gamma)$, there exists a $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$ with*

$$\left|Q_t(h^0_t, \gamma^*) - Q_t(h^0_t, \gamma_{\widehat{\lambda}, h^0_t})\right| \leq (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p, \tag{4.25}$$

*which leads to*

$$\left|V_t(h^0_t) - \max_{\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)} Q_t(h^0_t, \gamma_{\widehat{\lambda}, h^0_t})\right| \leq (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p. \tag{4.26}$$

We will again leave the proof until later, and instead present the intermediate results that are needed for the proof. Lemma 4.22 and Corollary 4.23 are extended versions of Lemma 4.12 and Corollary 4.13 to the approximate cases. Now the supervisor's $Q/V$ functions on different supervisor's states with the same private compression no longer align exactly, as we see the error grows linearly in the number of the remaining time steps $\bar{t} \triangleq (T-t)$.

**Lemma 4.22** (Extension of Lemma 4.12)**:** *Assume $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. For any $h^0_t \in \Omega(H^0_t)$, let $h_1, h_2 \in \Omega(H^{1:N}_t)$ be two FPSs under $h^0_t$ that map to the same ASPS $\widehat{z} \in \Omega(\widehat{Z}^{1:N}_t)$, i.e. $\widehat{z} = \widehat{\vartheta}^{1:N}_t(h^0_t, h_1) = \widehat{\vartheta}^{1:N}_t(h^0_t, h_2)$, and let $\gamma \in \Omega(\Gamma_t)$ be a prescription that chooses the same action on these two FPSs $\gamma(h_1) = \gamma(h_2) = a$. Let $\bar{t} = T - t$. Then we have*

$$\left|Q^S_t(h^0_t, h_1, \gamma) - Q^S_t(h^0_t, h_2, \gamma)\right| \leq \bar{t}(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2. \tag{4.27}$$

**Proof:** See Appendix C.3. ■

Note that Lemma 4.11 is irrelevant to the compression schemes, and thus continues to hold in the current approximate state representations context. Using it and Lemma 4.22, we show that the supervisor's $V$ function will differ little for two supervisor's states with the same compression of private states.

**Corollary 4.23** (Extension of Corollary 4.13)**:** *Assume $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. For any $h_t^0 \in \Omega(H_t^0)$, let $h_1, h_2 \in \Omega(H_t^{1:N})$ be two FPSs under $h_t^0$ that map to the same ASPS $\widehat{z} \in \Omega(\widehat{Z}_t^{1:N})$, i.e. $\widehat{z} = \widehat{\vartheta}_t(h_t^0, h_1) = \widehat{\vartheta}_t(h_t^0, h_2)$, and let $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$ be an optimal prescription. Let $\bar{t} = T - t$. Then we have*

$$\left|V_t^S(h_t^0, h_1) - V_t^S(h_t^0, h_2)\right| \triangleq \left|Q_t^S(h_t^0, h_1, \gamma^*) - Q_t^S(h_t^0, h_2, \gamma^*)\right| \leq \bar{t}(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2. \tag{4.28}$$

**Proof:** See Appendix C.3. ■

**Proof of Proposition 4.21:** Given an optimal prescription $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$, we will specifically construct a $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$ that serves for the claim. For each $\widehat{z} \in \Omega(\widehat{Z}_t^{1:N})$, define

$$\mathcal{H}_{\widehat{z}} = \left\{h \in \Omega(H_t^{1:N}) : \widehat{\vartheta}_t^{1:N}(h_t^0, h) = \widehat{z}\right\} \tag{4.29}$$

to be the class of $h$'s in $\Omega(H_t^{1:N})$ that are compressed to $\widehat{z}$ under the considered compression $\widehat{\vartheta}_t^{1:N}$. By the Axiom of Choice, for each $\widehat{z} \in \Omega(\widehat{Z}_t^{1:N})$ there exists a representative of the class $\mathcal{H}_{\widehat{z}}$ coming from arbitrary choice function, which we denote as $\bar{h}_{\widehat{z}}$. We then construct the $\widehat{\lambda}$ by $\widehat{\lambda}(\widehat{z}) = \gamma^*(\bar{h}_{\widehat{z}})$; the corresponding extension in $\Omega(\Gamma_t)$ will be

$$\gamma_{\widehat{\lambda}, h_t^0}(h) = \gamma^*\left(\bar{h}_{\widehat{\vartheta}_t^{1:N}(h_t^0, h)}\right) \qquad \forall\ h \in \Omega(H_t^{1:N}), \tag{4.30}$$

that is, the prescription first compresses the input FPS and finds the representative of the corresponding compression class, then it mimics what the optimal prescription would have done with the representative. For any $h \in \Omega(H_t^{1:N})$, we have

$$\begin{aligned}
&\left|Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, h, \gamma_{\widehat{\lambda}, h_t^0})\right| \\
\leq &\left|Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, \bar{h}_{\widehat{\vartheta}_t^{1:N}(h_t^0, h)}, \gamma^*)\right| \\
&\quad + \left|Q_t^S(h_t^0, \bar{h}_{\widehat{\vartheta}_t^{1:N}(h_t^0, h)}, \gamma^*) - Q_t^S(h_t^0, \bar{h}_{\widehat{\vartheta}_t^{1:N}(h_t^0, h)}, \gamma_{\widehat{\lambda}, h_t^0})\right| \\
&\quad + \left|Q_t^S(h_t^0, \bar{h}_{\widehat{\vartheta}_t^{1:N}(h_t^0, h)}, \gamma_{\widehat{\lambda}, h_t^0}) - Q_t^S(h_t^0, h, \gamma_{\widehat{\lambda}, h_t^0})\right| \\
\leq &\ (T - t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p,
\end{aligned}$$

as the first term is bounded by $(T-t)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2$ due to Corollary 4.23, the second term is 0 due to Lemma 4.11, and the third term is bounded by $(T-t)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2$ due to Lemma 4.22. If it happens to be the case that $h = \bar{h}_{\widehat{\vartheta}_t^{1:N}(h_t^0, h)}$, i.e. $h$ is the representative, then the original term $|Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, h, \gamma_{\widehat{\lambda}, h_t^0})|$ is 0. Taking the conditional expectation on $h$ given $h_t^0$ yields

$$\begin{aligned}
&\left| Q_t(h_t^0, \gamma^*) - Q_t(h_t^0, \gamma_{\widehat{\lambda}, h_t^0}) \right| \\
&= \left| \sum_{h_t} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) - \sum_{h_t} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma_{\widehat{\lambda}, h_t^0}) \right| \\
&\leq \sum_{h_t} \mathbb{P}(h|h_t^0) \left| Q_t^S(h_t^0, h, \gamma^*) - Q_t^S(h_t^0, h, \gamma_{\widehat{\lambda}, h_t^0}) \right| \\
&\leq \sum_{h_t} \mathbb{P}(h|h_t^0) \cdot \left[ (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p \right] = (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p.
\end{aligned}$$

■

The following main theorem characterizes the cost incurred by adopting ASPS-based prescriptions, that is, it characterizes the gap of the DP in Algorithm 4.5 in contrast to the optimal DP in Algorithm 4.1. The optimality gap scales quadratically with $\bar{t} = (T-t)$.

**Theorem 4.24:** *Assume $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. For any $h_t^0 \in \Omega(H_t^0)$ and $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$, there exists a $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$ such that*

$$Q_t(h_t^0, \gamma^*) - \widehat{Q}_t(h_t^0, \widehat{\lambda}) \leq \frac{\bar{t}(\bar{t}+1)}{2}(\epsilon_p + T\bar{R}\delta_p) + (\bar{t}+1)\epsilon_p \tag{4.31}$$

*and*

$$V_t(h_t^0) - \widehat{V}_t(h_t^0) \leq \frac{\bar{t}(\bar{t}+1)}{2}(\epsilon_p + T\bar{R}\delta_p) + (\bar{t}+1)\epsilon_p. \tag{4.32}$$

**Proof:** See Appendix C.3. ■

### 4.4.2 Compressing Common States

While restricting attention to ASPS-based prescriptions, we further compress the common history to an approximate representation by applying the state compression result of [112], in order to construct a general framework suitable for designing practical MARL algorithms.

**Definition 4.25:** *An $(\epsilon_c, \delta_c)$-approximate sufficient common state (ASCS) $\widehat{Z}^0_t$ is the output of a function $\widehat{Z}^0_t = \widehat{\vartheta}^0_t(H^0_t)$ satisfying the properties:*

***(ASCS1)*** *It evolves recursively $\widehat{Z}^0_t = \widehat{\phi}^0_t(\widehat{Z}^0_{t-1}, \widehat{\Lambda}_{t-1}, O^0_t)$.*

***(ASCS2)*** *It suffices for approximate performance evaluation*

$$\left|\mathbb{E}[R_t(S_t, A_t)|h^0_t, \widehat{\lambda}_t] - \mathbb{E}[R_t(S_t, A_t)|\widehat{\vartheta}^0_t(h^0_t), \widehat{\lambda}_t]\right| \leq \epsilon_c \ \forall \ h^0_t, \widehat{\lambda}_t.$$

***(ASCS3)*** *It suffices for approximately predicting common observation*

$$\mathcal{K}\left(\mathbb{P}(O^0_{t+1}|h^0_t, \widehat{\lambda}_t), \mathbb{P}(O^0_{t+1}|\widehat{\vartheta}^0_t(h^0_t), \widehat{\lambda}_t)\right) \leq \delta_c/2 \ \forall \ h^0_t, \widehat{\lambda}_t.$$

---

**Algorithm 4.6** Dynamic Programming with ASCSs and ASPS-based Prescriptions

```
$\check{V}_{T+1}(\widehat{z}^0_{T+1}) \triangleq 0$
for $t = T, \dots, 1$ do
    $\check{Q}_t(\widehat{z}^0_t, \widehat{\lambda}_t) = \mathbb{E}\left[R(S_t, \widehat{\Lambda}_t(\widehat{Z}^{1:N}_t)) + \check{V}_{t+1}(\widehat{\phi}^0_t(\widehat{Z}^0_t, \widehat{\Lambda}_t, O^0_{t+1}))|\widehat{Z}^0_t = \widehat{z}^0_t, \widehat{\Lambda}_t = \widehat{\lambda}_t\right]$
    $\check{V}_t(\widehat{z}^0_t) = \max_{\widehat{\lambda}_t \in \Omega(\widehat{\Lambda}_t)} \check{Q}_t(\widehat{z}^0_t, \widehat{\lambda}_t)$
```

---

**Proposition 4.26:** *Assume $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. Let $h^0_1, h^0_2 \in \Omega(H^0_t)$ be two FCSs. If $\widehat{z}^0 = \widehat{\vartheta}^0_t(h^0_1) = \widehat{\vartheta}^0_t(h^0_2)$, then for any $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$, with $\bar{t} = T - t$,*

$$\left|\widehat{Q}_t(h^0_1, \widehat{\lambda}) - \widehat{Q}_t(h^0_2, \widehat{\lambda})\right| \leq 2\bar{t}(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c. \tag{4.33}$$

**Proof:** See Appendix C.4. ■

The following result is nearly a direct application of [112] to this context, with Algorithm 4.5 as the original DP, Algorithm 4.6 as the new DP, the set of ASPS-based prescriptions as the action space, and $\widehat{\vartheta}^0_t$ as the compression of the state. We still provide the proof for completeness.

**Theorem 4.27:** *Assume $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. For any $h^0_t \in \Omega(H^0_t)$ and $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$, with $\bar{t} = T - t$, we have*

$$\widehat{Q}_t(h^0_t, \widehat{\lambda}) - \check{Q}_t(\widehat{\vartheta}^0_t(h^0_t), \widehat{\lambda}) \leq \bar{t}(\epsilon_c + T\bar{R}\delta_c) + \epsilon_c \tag{4.34}$$

*and*

$$\widehat{V}_t(h^0_t) - \check{V}_t(\widehat{\vartheta}^0_t(h^0_t)) \leq \bar{t}(\epsilon_c + T\bar{R}\delta_c) + \epsilon_c. \tag{4.35}$$

**Proof:** See Appendix C.4. ■

From Algorithm 4.1 to Algorithm 4.5, the FPSs are compressed to the ASPSs, which introduces a quadratic gap in $\bar{t}$ as characterized by Theorem 4.24. Then under ASPSs, from Algorithm 4.5 to Algorithm 4.6, the FCSs are further compressed to the ASCSs, which introduces a linear gap in $\bar{t}$ as characterized by Theorem 4.27. These two theorems bridge the relation between the original non-compressed $(Q, V)$-system and the approximately compressed $(\breve{Q}, \breve{V})$-system with ASCSs and ASPSs. In particular, in the following theorem which is the main result of this chapter, we bound the optimality gap of value functions obtained from performing DP with the general common and private representations satisfying the conditions in Definition 4.19 and Definition 4.25 as in Algorithm 4.6, in comparison to the optimal value functions computed from Algorithm 4.1.

**Theorem 4.28:** *Assume $\mathcal{R}$ is uniformly bounded almost surely by $\bar{R}$. For any $h_t^0 \in \Omega(H_t^0)$ and $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$, there exists a $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$ such that*

$$Q_t(h_t^0, \gamma^*) - \breve{Q}_t(\widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}) \leq \frac{\bar{t}(\bar{t}+1)}{2}(\epsilon_p + T\bar{R}\delta_p) + (\bar{t}+1)(\epsilon_c + \epsilon_p) + \bar{t}T\bar{R}\delta_c, \quad (4.36)$$

$$V_t(h_t^0) - \breve{V}_t(\widehat{\vartheta}_t^0(h_t^0)) \leq \frac{\bar{t}(\bar{t}+1)}{2}(\epsilon_p + T\bar{R}\delta_p) + (\bar{t}+1)(\epsilon_c + \epsilon_p) + \bar{t}T\bar{R}\delta_c, \quad (4.37)$$

*where $\bar{t} = T - t$.*

**Proof:** Combine Theorem 4.24 and Theorem 4.27. ■

Notice that Theorem 4.27 actually implies Proposition 4.26:

$$\begin{aligned}\left|\widehat{Q}_t(h_1^0, \widehat{\lambda}) - \widehat{Q}_t(h_2^0, \widehat{\lambda})\right| &\leq \left|\widehat{Q}_t(h_1^0, \widehat{\lambda}) - \breve{Q}_t(\widehat{\vartheta}_t^0(h_1^0), \widehat{\lambda})\right| + \left|\breve{Q}_t(\widehat{\vartheta}_t^0(h_2^0), \widehat{\lambda}) - \widehat{Q}_t(h_2^0, \widehat{\lambda})\right| \\ &\leq 2(T-t)(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c.\end{aligned}$$

We can see that mixing up two common states with approximately the same compression incurs a linear cost instead of a constant one (Proposition 4.26), just as switching to the DP using the approximate state (Theorem 4.27). The two common states are different states, and they will still transit to different future states, but again with approximately the same compression; as each transition incurs a constant cost, overall the gap grows linearly. If we switch to use the approximate (compressed) states, the two states will then be the same (compressed state), but the switching itself incurs a linear gap (Theorem 4.27). The results are in line with those of [112].

We saw earlier, that the compression of private states incurs a quadratic cost (Theorem 4.24). This is because when we compress FPSs to ASPSs, we are compressing the action space not the state space, as the set of prescriptions considered shrinks from FPS-based to ASPS-based. Each time an ASPS-based prescription is chosen, different *supervisor's states* still transit to different future supervisor's states (see the proof of Lemma 4.22), whose difference grows linearly. Since in the DP of Algorithm 4.5, each time an ASPS-based prescription is adopted potentially incurs a linear cost, the total gap will then be quadratic, in contrast to the case of state compression.

Note that in the above discussion, the optimality gap complexity is in terms of the number of remaining time steps until the episode ends, i.e. $(T-t)$. If one considers the complexity in terms of the time horizon $T$ for the value of the whole episode $\mathbb{E}[V_1]$, then the complexity rises to *cubic* in $T$. One may wonder how is it possible that the optimality gap be larger than $2T\bar{R}$, which is only linear in $T$? Indeed, when the parameters $(\epsilon_c, \delta_c)$ and $(\epsilon_p, \delta_p)$ are too large, the results in Theorem 4.28 can say nothing more than the trival bound of $2T\bar{R}$; the cubic dependence only appears when the approximate parameters are small enough. The results in Theorem 4.28 still does not reflect actual regret in practical implementation though, it is only the optimality gap given the approximate parameters *in one episode*. In practical implementation, one uses ML algorithms (e.g. function approximation methods) to learn effective representations of ASCSs and ASPSs and minimize the approximation errors, while using common policy gradient methods [115] to learn the optimal policy (instead of solving DP directly). The true regret in practical implementation will then depend on both the convergence rate of the state representation learning and the policy gradient method used.

## 4.5 Discussions

In this section, we compare the differences between existing schemes and our methods in Section 4.5.1, discuss the practical implications of our approximate state representation framework in Section 4.5.2, and propose an algorithmic framework in Section 4.5.3.

### 4.5.1 Comparisons to Existing Schemes

We compare our methods to the results proposed in [88], [117], and [84].

**Relation to [88].** The private history is not compressed in [88], so it is clearly a special case of SPS. The BCS proposed in [88] is a special case of SCS as well.

**Proposition 4.29:** *The BCS* $\Pi_t = \mathbb{P}(S_t, H_t^{1:N} | H_t^0)$ *satisfies the conditions of an SCS in Definition 4.16.*

**Proof:** See Appendix C.5. ■

**Relation to [117].** Our conditions of SPS and [117]'s conditions of SPI both lead to performance sufficiency of the space of SPI-based (or SPS-based) prescriptions. The two sets of conditions are similar but not exactly the same. Condition (SPI1) corresponds to (SPS1); however, (SPS1) is stricter since we require *policy-independent compression*, while [117] allow policy-dependent compression. Condition (SPI3) ensures *future sufficiency* as does (SPS3). Conditions (SPI2) and (SPI4) together ensure *present sufficiency* as does (SPS3). In [117], $h_t^n$ is used instead of $h_t^{1:N}$ here because the paper considers potentially more general cases where utilities can be different for different agents, while we only focus on *team problems*. The relations of these conditions are described below.

**Proposition 4.30:** *(SPS1) and (SPS3) imply (SPI3).*

**Proof:** See Appendix C.5. ■

**Proposition 4.31:** *(SPS2) and (SPI4) imply (SPI2).*

**Proof:** See Appendix C.5. ■

Table 4.1: Dynamic programming comparison.

| Work | Algorithm /Definition | Agent | Common State | Private State | Action | Compression Common/Private | Incurred DP Gap Common/Private |
|---|---|---|---|---|---|---|---|
| | | Single | AOH | - | Action | None | 0 |
| Belief State | (4.3) | Single | BS | - | Action | Lossy | 0 |
| [112] | 4.1 | Single | IS | - | Action | Lossy | 0 |
| [112] | 4.3 | Single | AIS | - | Action | Lossy | Linear |
| CI Approach (No Compression) | 4.1 | Multi | FCS | FPS | FPS-pres. | None/None | 0/0 |
| [88] | 4.2 | Multi | BCS | FPS | FPS-pres. | Lossy/None | 0/0 |
| [117] | 4.3 | Multi | BCS | SPI | SPI-pres. | Lossy/Lossy | 0/0 |
| [113] | - | Multi | ASCS | FPS | FPS-pres. | Lossy/None | Linear/0 |
| [84] | - | Multi | BCS | ASPS | ASPS-pres. | Lossy/Lossless | 0/Linear |
| This work | 4.4 | Multi | SCS | SPS | SPS-pres. | Lossy/Lossy | 0/0 |
| This work | 4.5 | Multi | FCS | ASPS | ASPS-pres. | None/Lossy | 0/Quadratic |
| This work | 4.6 | Multi | ASCS | ASPS | ASPS-pres. | Lossy/Lossy | Linear/Quadratic |

Restricting to SPS, their BCS $\tilde{\Pi}_t = \mathbb{P}(S_t, Z_t^{1:N} | H_t^0)$ is a special case of SCS as well.

**Proposition 4.32:** *The BCS $\tilde{\Pi}_t = \mathbb{P}(S_t, Z_t^{1:N} | H_t^0)$ satisfies the conditions of an SCS in Definition 4.16.*

**Proof:** The proof is the same as the proof of Proposition 4.29, with $\Pi_t$ replaced by $\tilde{\Pi}_t$, $H_t^{1:N}$ replaced by $Z_t^{1:N}$, and $\gamma_t$ replaced by $\lambda_t$. ■

**Relation to [84].** Their private state embedding does not require a recursive update (ASPS1), but demands injective functions $\widehat{\vartheta}_t^{1:N}$. With this additional assumption they show linearity of the optimality gap in remaining time. For the common state, the BCS they consider is a special case of our SCS, just as the BCS of [117].

A thorough comparison of the DPs proposed in this chapter and in the literature is summarized in Table 4.1.

### 4.5.2 Practical Implications

The theory developed in Section 4.4 provides a nice theoretical support of designing practical low-regret deep-MARL algorithms, just as the way the DL schemes solves POMDP RL proposed in [112, 113]. Algorithms using this approximate state representation framework will consists of two steps when learning the optimal policies. In the first step, the agents use DL models (or other function approximation methods) to learn good representations of the common states and the private states.

Our contribution is to identify measures for "good representations" – they should satisfy the conditions of ASCS and ASPS with low error parameters $(\epsilon_c, \delta_c, \epsilon_p, \delta_p)$. In particular, the DL models will try to predict the instantaneous reward and new observations while quantities in the LHS of the conditions are good candidates of the loss functions. In the second step, assuming the agents have learned good state representations, they will use policy approximation theory in RL [115] to learn good policies from the *coordinator's view*, thus alleviating the non-stationarity issue. The agents, from the perspective of the coordinator, will first feed the common state representation to the policy function approximator, which can be suitably implemented by a DL model as well, and generate an ASPS-based prescription; they choose their actions prescribed by the prescription based on their ASPSs, and get the feedback reward from the environment. The parameters of the function policy approximator, now usually a DL model, can then be updated by policy gradient theorem [115] with an approximation of the long-term reward. In [112, 113], the two steps are performed concurrently using the concept of two time-scale algorithm [20], with the first step being the fast time-scale and the second step being the slow time-scale. Indeed, the policy function approximator learns an optimal policy *based on the learned state representations* in the first step. A more concrete algorithmic framework is given in Section 4.5.3.

It should be noted that in the learning of the first step, the agents need to have all the information, including the private states of other agents. This is because in (ASPS2) and (ASPS3), the full tuple of ASPS $\widehat{z}_t^{1:N}$ is needed to predict the instantaneous reward and new observations. This necessitates "centralized learning decentralized execution" scheme, where the difference between being centralized or decentralized is judged from the information sharing scheme – a centralized scheme requires full information sharing. For the first step we mentioned above, i.e. the representation learning step, we need a centralized learning scheme. The centralized learning decentralized execution scheme was first proposed to tackle the non-stationarity issue, as when the learning or exploration is centralized, there is no information asymmetry and agents are able to keep good track of others' policies; and when it comes to execution or exploitation stage, the issue no longer exists as the agents have stopped updating their policies. In our case, however, the second step we mentioned above, i.e. the policy learning step, does *not* require centralized learning thanks to the application of the CI approach in our theory.

The centralized learning decentralized execution scheme is good enough for many applications, such as gaming AI or autonomous vehicle, where an "environment simulator" that is identical (or nearly) to the real environment is accessible, so that the agents can learn through the simulator with full information sharing offline. There are some applications that demand decentralized learning though; for example, a sensor network may only starts to learn the environment when deployed, a time when information sharing may be hard or even impossible. What could be done in this case then? From the theory perspective, we need conditions like (SPI2) where only $z_t^n$ instead of the whole $z_t^{1:N}$ is required for the representation learning for agent $n$. But as Definition 4.5 and Proposition 4.31 suggest, in that case further "consistency condition(s)" such as (SPI4) will be required. Such condition does not solve the problem, as it requires agent $n$ to predict $z_t^{-n}$ and compare the prediction with the ground truth, which agent $n$ does not have access to (otherwise agent $n$ just knows $z_t^{1:N}$); in other words, the condition is impossible to be implemented in the decentralized learning setup too. The networked MARL schemes [136, 135] provide another possible direction. The MARL problem is much like a distributed optimization one in the sense that the agents try to optimize long-term reward by choosing the *optimal policy parameters*. However, sharing the parameters also requires communication, and the amount of information may not be smaller that of $z_t^{-n}$; then why do not the agents just share $z_t^{-n}$? In conclusion, whether it is possible to find general fully decentralized MARL algorithms that is guaranteed to learn (near) optimal policies without any communication, and if not, how to design *communication efficient* general decentralized MARL algorithms with guaranteed convergence to optimality, remain to be open problems, and we leave them as future directions.

### 4.5.3 Algorithmic Framework

In this subsection we propose an MARL algorithmic framework using the theory developed in Section 4.4; the designing detail is left as future work. The framework adopts the "centralized learning distributed execution" scheme, i.e., the agents assume the omniscient supervisor's view when they learn the compressions and policies.

**Algorithm 4.7** Deep-MARL Framework

```
Common part: coordinator computes (done in each agent in execution phase) $\widehat{Z}^0_t = \rho^0(O^0_t, \widehat{\Lambda}_{t-1})$, $\widehat{\Lambda}_t = \varphi^0(\widehat{Z}^0_t)$.
Private part: agent $n$ computes $\widehat{Z}^n_t = \rho^n(O^n_t, A^n_{t-1})$, $A^n_t = \varphi^n(\widehat{Z}^n_t, \widehat{\Lambda}^n_t)$.
if in learning phase then
    Coordinator computes $(\hat{R}_t, \hat{O}^0_{t+1}) = \psi^C(\widehat{Z}^0_t, \widehat{\Lambda}_t)$.
    $(\hat{R}_t, \hat{O}^0_{t+1})$ is compared with ground truth $(R_t, O^0_{t+1})$ and loss is back-propagated to $(\rho^0, \psi^C)$.
    Supervisor computes $(\hat{R}_t, \hat{O}^{0:N}_{t+1}) = \psi^S(\widehat{Z}^{0:N}_t, A^{1:N}_t)$.
    $(\hat{R}_t, \hat{O}^{0:N}_{t+1})$ is compared with ground truth $(R_t, O^{0:N}_{t+1})$ and loss is back-propagated to $(\rho^{0:N}, \psi^S)$.
    Coordinator computes $\sum_{\tau=t-W}^{t} R_\tau$ and performs policy gradient on $\varphi^{0:N}$.
```

There are three types of functions within: the state networks $\rho^{0:N}$ modeled by recurrent neural networks (RNNs), the policy networks $\varphi^{0:N}$ modeled by deep neural networks (DNNs), and the prediction networks $\psi^C$ and $\psi^S$ also modeled by DNNs. The state networks $\rho^{0:N}$ serve the purpose of the compression mappings $\widehat{\phi}^{0:N}_t$ in Definition 4.19 and Definition 4.25, and their recursive evolution structures suggest an RNN modeling. The common policy network $\varphi^0$ takes $\widehat{Z}^0_t$ as input and gives the prescription $\widehat{\Lambda}_t$ as suggested by Section 4.4.2; the private policy network $\varphi^n$ takes $\widehat{Z}^n_t$ and $\widehat{\Lambda}^n_t$ and outputs $A^n_t$. Here, we have to use a *variable* to represent the *prescription function*; hence, it cannot be directly applied to $\widehat{Z}^n_t$. Effective design of representing prescription function is left as future work, even though Lemma 4.11 provides a nice decomposition. Finally, the policy networks $\psi^C$ and $\psi^S$ are used to produce the predicted reward and new observations. In the learning phase, the predictions are compared with the ground truth and errors are back-propagated through the state and prediction networks. This requires full knowledge of $\widehat{Z}^{1:N}_t$ and $O^{1:N}_t$; consequently, the learning has to be centralized. A windowed (with length $W$) cumulative reward is summed for the computation of the loss in policy gradient methods [115], which is back-propagated through the policy networks; actor-critic methods can also be employed here. In the execution phase, only state and policy networks are required, and everything can be performed in a decentralized fashion. Note that in our proof of Theorem 4.24 only the fact that $\widehat{Z}^{1:N}_t$ can be updated from $\widehat{Z}^{1:N}_{t-1}$ is needed.

## 4.6 Change Detection in Time-Varying Environments

In this section, we consider MARL problems where the environment is non-stationary. In particular, we consider the easier case of a piece-wise stationary environment instead of a constantly-changing one, and investigate how the agents can adapt to discrete changes in the environment. In Section 4.6.1, we consider the cases where the model structure is partially known, and propose a change detection (CD) framework for MARL. In Section 4.6.2, we apply the framework we propose to the matching problem introduced in Section 4.1.2.2 for changing environment. We discuss possible ways to go beyond partially known structure in the model by combining the approximate state representations framework with the CD framework in Section 4.6.3.

### 4.6.1 Partially Known Structure

We consider changing environments setting where the models of the environments are partially known, so that we can decompose the problem into an estimation problem and a control problem; the agents estimate the parameters in the first part, and then adopt the optimal policy corresponding to the estimated parameters based on the solution of the control problem. In particular, we assume all underlying environments conform to certain structure $E = \mathcal{E}(\xi)$, and when the environment changes, it is the parameter $\xi$ that changes. We first consider a simpler setting where the environment changes from $E^1 = \mathcal{E}(\xi^1)$ to $E^2 = \mathcal{E}(\xi^2)$, where the two parameters $\xi^1$ and $\xi^2$ are completely unknown. We propose a change detection (CD) framework, which is an extension to the CD-RL framework of the single-agent case studied in [79], for the agents to adapt to the change and minimize regret. Our framework conceptually consists of three parts: (1) first it estimates the parameter $\xi^1$ and adopts the corresponding optimal policy, (2) it monitors the transition behavior and announces an alert when an environment change is believed to have occurred, (3) it adapts to the new environment by re-estimating the new parameter $\xi^2$ and adopts the corresponding optimal policy. To use the framework, there are three key assumptions that need to be satisfied.

**Assumption 4.33:**

*(a) Once entering a mode (characterized by an environmental parameter), the en-*

*vironment will stay in the mode for at least $W$ time steps. Since now we only consider $E^1$ changes to $E^2$, this implies the environment will stay in $E^1$ for at least $W$ steps.*

(b) *The gap between the two different parameters is larger than a fixed positive number $\Delta = \|\xi^1 - \xi^2|/3 > 0$.*

(c) *An unbiased estimate of the environmental parameter is available from the AOH of the coordinator's view $H_t^0$ at any time $t$, i.e. there exists $\hat{\xi}(H_t^0)$ such that $\mathbb{E}[\hat{\xi}(H_t^0)] = \xi_t$, where $\xi_t$ is the true environmental parameter at time $t$ (here $\xi_t$ is either $\xi^1$ or $\xi^2$).*

Denote $\bar{\xi}(t) \triangleq \frac{1}{t}\sum_{\tau=1}^{t}\hat{\xi}_\tau$ as the average of first $t$ samples of $\hat{\xi}$. Note that from Assumption 4.33 (a) and (c), $\bar{\xi}(W)$ is an unbiased estimate of $\xi^1$. Moreover, denote $\bar{\xi}(t_1, t_2) \triangleq \frac{1}{t_2 - t_1}\sum_{\tau=t_1+1}^{t_2}\hat{\xi}_\tau$ as the average of the samples of $\hat{\xi}$ from $t_1 + 1$ to $t_2$. Now we formally introduce the CD-MARL framework in Algorithm 4.8.

**Algorithm 4.8** CD-MARL Framework

```
input: horizon $T$, window length $W$, threshold $\theta$, optimal policy $g^*(\cdot)$
for $t = 1, \dots, W$ do
    Collect $\hat{\xi}_t$, then compute $\bar{\xi}(t)$ and adopt $g^*(\bar{\xi}(t))$
for $t = W + 1, \dots, T$ do
    Collect $\hat{\xi}_t$, then compute $\bar{\xi}(t)$ and adopt $g^*(\bar{\xi}(t))$
    if $\|\bar{\xi}(t) - \bar{\xi}(W)\| > \theta$ then
        Assign switching time $\tau \to t$
        break
for $t = \tau + 1, \dots, T$ do
    Collect $\hat{\xi}_t$, then compute $\bar{\xi}(\tau, t)$ and adopt $g^*(\bar{\xi}(\tau, t))$
```

The algorithm can be trivially extended to the case of multiple environment changes (instead of only two). Intuitively, the average $\bar{\xi}(t)$ will perform a random walk around $\xi^1$ until the switch, when it will start to have mean drifts toward $\xi^2$. Hence, when $\bar{\xi}(t)$ falls outside the $\theta$-ball centered at $\bar{\xi}(W)$ where $\theta$ is the given threshold, the framework alerts a change, and the agents have to re-estimate the parameter of the new environment.

### 4.6.2 Application to the Matching Problem

The matching problem (Section 4.1.2.2) is in line with the setting of having a known model structure while a few parameters within may be unknown. In particular, it is characterized by the transition parameter $\epsilon$ and the observation parameter $\beta$, which the agents may not know in advance[13]. Here, we can treat the optimal policy under $\epsilon$ and $\beta$ as a black box $g^*(\epsilon, \beta)$; the detailed analysis is deferred to Section 5.6.

We consider the simple case that only the parameter $\epsilon$ changes. Now that it is understood that we consider the environment changes from $E^1 = (\epsilon^1, \beta)$ to $E^2 = (\epsilon^2, \beta)$, we examine how Assumption 4.33 is satisfied in our setting. For Assumption 4.33 (a), we simply assume it is the case that the environment will stay in $E^1$ for $W$ steps. Assumption 4.33 (b) is not a big issue either since it only requires $\epsilon^1 \neq \epsilon^2$; the gap parameter $\Delta$ is related to how the threshold $\theta$ should be selected, which will not be discussed here. The crucial assumption is Assumption 4.33 (c). Indeed, as analyzed in Section 5.6, $S_t$ becomes common information at time $t+1$, while $\bar{S}_t$ becomes common information at time $t$ (since it is computed from $S_{t-1}$ and $Y_t$). Define

$$\bar{\epsilon}_t(S_t, \bar{S}_t) = \begin{cases} 1, & \text{if } S_t = -\bar{S}_t, \\ 0, & \text{if } S_t = \bar{S}_t. \end{cases} \tag{4.38}$$

From Section 5.6, we know that $\bar{\epsilon}_t(S_t, \bar{S}_t)$ is an unbiased estimate of $\epsilon_t$. Technically, this unbiased estimate requires $H^0_{t+1}$. However, we can view this as a one-step delay from the framework, and the framework can still be applied completely in the problem. Note that the reason we assume $\beta$ is fixed is because an unbiased estimate of $\beta_t$ is unavailable from the AOH $H^0_t$, so the framework does not apply.

We propose four methods and simulate them for performance comparison. Based on whether $\beta$ is known in advance, we divide them into two groups. In the case that $\beta$ is known, the Estimate and Adopt (EA) method computes the long-term average $\bar{\epsilon}(t)$ and directly adopts the optimal policy $g^*(\bar{\epsilon}(t), \beta)$ based on this estimation; on the other hand, the Change Detection Estimate and Adopt (CD-EA) method uses

---

[13] The agents may not know the cost parameter $c$ at first as well, but they can learn its value by letting A1 have a one-time try on the communication option then observing the reward value. Hence, we will assume it is known.

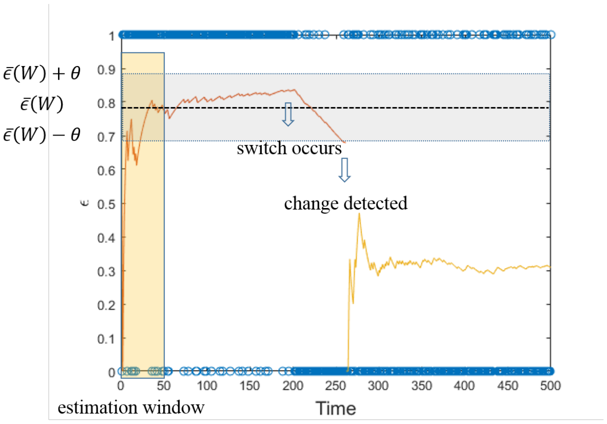


Figure 4.3: An illustration of the tracked $\epsilon$ average of the CD-EA algorithm being used in the matching problem.

$g^*(\bar{\epsilon}(t),\beta)$ at first, but alerts a change when $\bar{\epsilon}(t)$ drifts far away enough from $\bar{\epsilon}(W)$

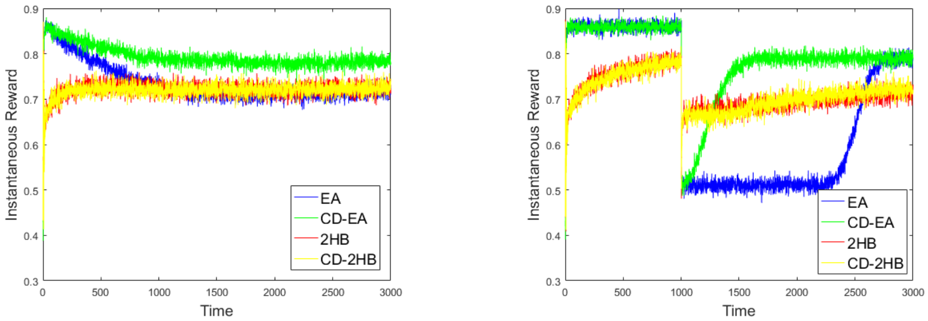
$$|\bar{\epsilon}(t)-\bar{\epsilon}(W)|>\theta \tag{4.39}$$

at $t=\tau$ ($\tau$ depending on the actual process), and uses $g^*(\bar{\epsilon}(\tau,t),\beta)$ thereafter. Note that CD-EA is exactly Algorithm 4.8.

An illustration of how CD-EA works is given in Fig. 4.3 (resulted from real simulation experiment described below). The blue circles are unbiased samples of $\epsilon$, which are either 1 or 0. In the first $W=50$ time steps, the agents gather the sampled $\epsilon$'s, take their average $\bar{\epsilon}(W)$, set a $\theta$ range around the average, and keep track of the average as new samples come in. The environment switch occurs at $t=200$, and the sampled $\epsilon$'s begin to have a negative mean shift until around $t=260$ when the change is detected. The agents then discard all old samples and take the average of only the samples that comes after $t=260$, and adopt optimal policy based on new averages.

In the case that $\beta$ is unknown too, we consider the direct Two-stage Hierarchical Bandit (2HB) method proposed in Section 5.3 without change detection, and change detection working with Two-stage Hierarchical Bandit (CD-2HB) method. Specifically, in CD-2HB, the bandit algorithm is used to search the optimal policy while the agents keep monitoring $\bar{\epsilon}(t)$; once a change is detected, i.e. (4.39) is satisfied, the CD-2HB will reset recorded rewards and counters for the bandits and start over.

Figure 4.4 shows how the four change detection algorithms perform in the matching problem. The simulation setting is quite similar to the previous one. We have



(a) A geometrically distributed switch of environments.

(b) A switch of environments occurring at $t = 1000$.

Figure 4.4: Instantaneous rewards of four advanced adaptive algorithms in the matching problem with unknown $\epsilon$'s.

$E^1 = (\epsilon^1, \beta) = (0.3, 0.7)$ and $E^2 = (\epsilon^2, \beta) = (0.8, 0.7)$, $c = 0.7$, and $\gamma = 0.999$. The estimation window length is $W = 50$, and the allowed fluctuating range is $\theta = 0.1$. In Figure 4.4 (a), the underlying environment starts with $E^1$, but in each time step there is a chance of $p = 0.001$ that it will switch to $E^2$; in Figure 4.4 (b), the environment starts with $E^1$ and switch to $E^2$ exactly at $t = 1000$. The final results are averaged through 1000 Monte-Carlo trials. The optimal value of $E^1$ is 0.86, and that of $E^2$ is 0.79.

In Figure 4.4 (a), the EA methods outperform the 2HB methods, since they exploit the structure of the model and optimal policy so the only task is to estimate the parameter $\epsilon$, while the 2HB methods assume no prior knowledge. The CD-EA method further outperforms the EA method, because it discards $\epsilon$ samples from $E^1$ after the change which allows it to quickly adapt to $E^2$ while EA does not. The CD-2HB method is hardly better than the 2HB method, since bandit methods are intrinsically *soft*, which means they keep exploring. Whether or not a change is detected and the algorithm is reset to start over, they spend a portion of steps exploring. On one hand, they are adaptive even without change detection mechanism; on the other hand, they only converge to optimal policy as time goes to infinity.

In order to gain more insight, we let the switch occur at exactly $t = 1000$ and simulate their instantaneous rewards, which is plotted in Figure 4.4 (b). Before the switch, the EA methods, knowing the structure, are operating the optimal policy as expected, while the 2HB methods quickly pick up. The CD-EA method detects the switch around $t = 1100{\sim}1400$, and approaches optimum around $t = 1500$. The EA method does not discard old samples, so while the averages are shifting negatively towards the true $\epsilon^2$, they correspond to a suboptimal policy until around $t = 2500$ when the averages are finally close enough to $\epsilon^2$ so that agent adopt the optimal policy of $E^2$.

Right after the switch, the 2HB methods are actually better than the EA methods due to the "softness" of bandit algorithms. They are surpassed by both EA methods near the end as EA methods exploit the additional knowledge; however, they will converge to optimum too as time grows. Around $t = 1100{\sim}1400$, CD-2HB detects the switch and completely re-explores, causing lower instantaneous rewards than 2HB in the period; but the re-exploration starts to pay off around $t = 2000$ when CD-2HB marginally outperforms 2HB.

### 4.6.3 Completely Unknown Structure

Going beyond partially known model structure to completely unknown models, our proposed approximate state representation framework for MARL can come into play. The framework provides a natural way to monitor the environment, as the conditions define "how well the representations encapsulate the histories" by their *abilities to predict instantaneous reward and future observations.* If at some time, the agents find their predictions have become worse than usual, it is a good indication that the environment might have changed. Specifically, the differences between the predicted instantaneous reward $\hat{R}_t$ (by the state representation DL models) and the corresponding ground truth $R_t$, and the predicted future observations $\hat{O}^{0:N}_{t+1}$ and the corresponding ground truth $O^{0:N}_{t+1}$, provide good estimations of the error parameters $(\epsilon_c, \delta_c, \epsilon_p, \delta_p)$ we want to minimize. As the representation models capture the environment, the parameters (or at least their averages within a window) should stay steady and low. Once they change by more than some threshold, the agents may alert an environmental change and start to re-learn the state representations and policies, just as in Algorithm 4.8.

In our approximate state conditions (Definition 4.25 and Definition 4.19), however, full ASPS tuple $\widehat{z}^{1:N}_t$ is required to predict instantaneous reward and future observations. In the decentralized execution stage, this is not possible, while we do wish the agents to be able to detect environmental change at this stage. This goes back to the open question discussions in Section 4.5.2. The agents either need to communicate their information, or one has to find similar conditions that only require $\widehat{z}_t^n$ for agent $n$'s predictions. In the latter case, the agents may declare an environmental change only when a majority of the agents believe that is the case – the change alert signals would require least amount of communication in comparison to other schemes.

## 4.7 Conclusion and Future Directions

In this chapter, we developed a general approximate state representation framework for MARL problems and characterized the episodic optimality gap in terms of the approximation errors. We first re-established the result of [117] that confines to SPS-based prescriptions from a DP perspective, and proposed the notion of SCS that incorporates the existing results in decentralized control that based on the CI approach, including [88], [117], and the no compression scheme. Inspired by the AIS framework proposed in [112], we extended the state representation framework for the Dec-POMDP setting to its approximate correspondent, where we bounded the episodic optimality gap by a linear growth for common state error, and a quadratic growth for private state error, in terms of number of remaining steps. Such result can be seen as a generalization of the result in [112] to the multi-agent cases. Finally, we proposed a CD framework for MARL problems and discussed possible ways to combine it with the approximate state representation framework when the environment is non-stationary.

The approximate state representation framework opens up innumerable future directions. The most direct one that is not carried out in this thesis is to design practical DL algorithms whose structures are based on the framework. This is by no means trivial, as even with ASPS now being in a fixed space, the space of its prescriptions is still huge and grows exponentially in $N$. Another notable direction is to go beyond the centralized learning scheme, which may involve investigating new private state representation conditions that only require individual private information, or designing efficient communication scheme between the agents to maintain consistency. Yet another direction is to combine the ASCS with individual agent's ASPS to only one state for the agent and characterize the optimality gap; such an idea is already proposed in [117], but without any approximations with performance guarantee that could guide practical implementations. Other directions which include DL implementation details, simplifications that cater to specific information structures, concise and possibly lossy representation of prescriptions, generalizations beyond team problems to game scenarios, etc. necessitate more research in this promising field.

## CHAPTER V

# General Common and Private Approximate State Representations in Multi-Agent Reinforcement Learning

## 5.1 Introduction

Multi-agent reinforcement learning (MARL) has received great attention due to its wide variety of applications and the tremendous advances in single-agent RL techniques [58]. In a multi-agent environment, each agent has different observations and may have different sets of information. This is referred to as the *information asymmetry* property [30]. One straightforward method used with information asymmetry is to let every agent concurrently learn based on its own information using single-agent algorithms. However, this creates the *non-stationarity issue* since the effective environment observed by each agent is time-varying, which sometimes causes non-convergence of the learning algorithms. Another line of solutions is to enforce coordination among agents, essentially transforming a multi-agent system back to (or making it more similar to) a single-agent one. One way to achieve this is through communication [110, 135], which introduces extra costs that may be intolerable in some cases. A more broadly applicable scheme is through the common information (CI) approach [88, 30, 37]. The CI approach relies on a set of CI shared by all agents, and all agents need to agree on a protocol that specifies the joint policy updates of all agents upon receiving a certain piece of CI. With this protocol, an agent may be able to infer the actions taken by other agents without observing them. However, while this approach conceptually solves the problem, it has several shortcomings that make it hard to apply in practice: it has high computational complexity (since every agent has to perform policy updates for all the other agents); and it requires all agents to have very tight coordination (e.g., sharing randomization seeds in each round and knowing all details in the algorithms of other agents), which may be infeasible if synchronization among agents or agent privacy is an issue.

In this chapter, we address the aforementioned issues in a special but widely applicable MARL setting. We consider MARL team problems with a particular hierarchical information structure between agents under the settings of multi-agent multi-armed bandits (MAMABs) and multi-agent Markov decision processes (MDPs). In this structure, decisions are made sequentially, and a decision maker has all the decisions from decision makers that act before it in the sequence. In the two-agent case, one of the agents (the "leader") chooses her action first, while the other agent (the "follower") chooses his action after observing the leader's action. This setting is similar to the Stackelberg game but with the players cooperating to achieve the same objective. Such hierarchical information structure arises in many applications. For example, in a cognitive radio (CR) wireless network, the primary user (PU) first decides its resource allocation scheme; then based on this scheme the secondary user (SU) chooses its own resource allocation scheme that minimally interferes with the PU's transmission [93]. While this problem can also be solved using the CI approach or other MAMAB algorithms [61, 30], as discussed earlier, they are intensive either in computation or in communication. In this work, we exploit the hierarchical information structure, and propose simpler and more efficient MARL algorithms that require neither communication nor explicit coordination, while achieving near-optimal regret bounds. Such algorithms could be much easier to deploy in practice.

In more detail, we first consider the two-agent bandit setting, where both agents observe the same reward determined by their *joint action*, but only the follower observes the leader's action but not *vice versa*. For this setting, we propose a decentralized algorithm that achieves a near-optimal gap-independent regret bound of $\widetilde{\mathcal{O}}(\sqrt{ABT})$[1] and a gap-dependent bound of $\mathcal{O}(\log(T))$, where $A$ and $B$ are the numbers of actions of the leader and the follower, respectively, and $T$ is the number of

---

[1] We use $\widetilde{\mathcal{O}}(\cdot)$ to hide poly-logarithmic factors.

time-steps. In our method, both agents perform Upper Confidence Bound (UCB)-based algorithms [5], with a modified bonus term in the leader's algorithm that is related to the follower's regret bound. Without explicit coordination, the agents perform joint exploration over the action space and learn with low regret. We further extend our idea to two more complicated settings. The first is the case of multiple agents simultaneously deciding their actions in each of the two stages. The other is the case with a deep hierarchy, where more than two agents make decisions sequentially based on the decisions made by prior agents, and all agents observe the same reward. In both extensions, our algorithms also achieve near-optimal regret bounds.

Next, we generalize the above idea to the two-agent MDP setting. In this setting, the state evolution and reward observable by both agents are sampled from distributions depending on the current state and the joint action of the agents; as before, the follower observes the leader's action but not vice versa. Similar to the bandit case, we propose a decentralized learning method that enables the agents to perform joint exploration without communication or explicit coordination. Our algorithm is based on an intriguing combination of two exploration strategies developed for single-agent reinforcement learning: UCB-H [57] and UCBVI [7]. By letting the leader execute a UCB-H-styled algorithm and the follower use a UCBVI-styled one, the agents jointly achieve a regret upper bound of $\widetilde{\mathcal{O}}(\sqrt{H^7S^2ABT})$ ($H$ is the horizon length, and $T$ is the number of episodes), while the regret lower bound is $\Omega(\sqrt{H^2SABT})$, inherited from the single-agent MDP setting [7]. Tightening our bound without sacrificing the benefit of decentralized learning is left as an open question.

The rest of the chapter is organized as follows. The related work is reviewed in Section 5.1.1. We describe the hierarchical bandit and the hierarchical MDP settings in Section 5.2. The two-stage UCB algorithm for the hierarchical bandit is presented in Section 5.3 where we show its regret is near-optimal, with two extensions given in Section 5.4. Then we develop an algorithm based on UCB-H and UCBVI for the hierarchical MDP with near-optimal regret in Section 5.5. In Section 5.6, we derive an optimal policy for the matching problem (introduced in Section 4.1.2.2), and demonstrate the application of the two-stage UCB algorithm to the problem in the MARL context, i.e. when the model is unknown. The conclusion is drawn in Section 5.7.

### 5.1.1 Related Work

Algorithms for various MAMAB settings have gained increasing interest recently, but there is only a limited literature that investigates the effects and challenges caused by information asymmetry in the setting where agents jointly interact with the environment as is common in MARL applications, with the corresponding MDP setting receiving even less attention. The authors of [30] study the MAMAB setting where the reward is determined by the joint action with three types of information asymmetry: unobserved actions and common rewards, observed actions and independent rewards, and unobserved actions and independent rewards; the first two settings can be solved by the notion of the CI approach[2], while in the last setting they propose an "explore then commit" type algorithm that achieves an $\mathcal{O}\left(\log(T)\right)$ regret. The paper [9] considers sample-efficient learning in bandit games and bandit-RL games. Their bandit game corresponds to our hierarchical bandits, but under a general reward setup (i.e., in their setting, the rewards of the two agents have different means, unlike the common-reward setting we consider here). They consider centralized and offline learning assuming access to a sample generator, while we consider decentralized and online learning through interactions with the environment. While their results imply a worst case information-theoretic gap to the Stackelberg game value that cannot be closed, it is not the case in our team problem. On the other hand, our hierarchical MDP has a more general transition structure than their bandit-RL game (in their setting, only the follower is involved in an MDP). The authors in [3] study meta-learning over bandit algorithms, which exhibits a similar mathematical structure to our hierarchical bandit problem, though from a very different perspective. With the follower using any algorithm that achieves sub-linear regret, our hierarchical bandit algorithm can be used as an algorithm for their setting [62], and our gap-dependent bound improves their $\mathcal{O}(\log^2(T))$ bound by a factor of $\log(T)$, resolving their question on the tightness of their result.

Most papers on MAMAB consider a set of agents pulling *the same set of arms* simultaneously, and in most of them the agents coordinate through *real-time communications* to collaboratively find the optimal policy, with a few exceptions [19, 18, 27].

---

[2] Their mUCB algorithm for the first setting is equivalently the CI approach in combination with the UCB1 algorithm.

In the former, the communication resource is either costly [81], limited by budget [67, 123, 108, 31], or constrained through communication networks [68, 110], so the main focus is on designing communication efficient schemes that achieve the same performance as if there were no information asymmetry. In another related thread, referred to as the matching bandits problem, agents choosing the same arm collide and obtain zero rewards [61, 19, 18]; here and in a few other works [110], different agents get different distribution of rewards from the same arm, while in other referenced work they get independent and identically distributed samples from the same arm.

Regret minimization in MARL is in general challenging due to the fact that every agent faces a non-stationary environment. It has been shown in [1, 103, 120] that for single-agent non-stationary MDP problems, to have a sub-linear-in-$T$ regret bound against the best policy is both computationally and statistically hard. Therefore, to establish meaningful guarantees in MARL while keeping the algorithm efficient, special properties of the problem have to be considered. The paper [103] considers the same two-agent collaborative setting as ours, but requires that the agents exchange their policies after each episode. The authors of [120] study another two-agent setting where each agent is agnostic about the actions of the other; however, their algorithm is conservative (with the goal of guarding against an adversarial opponent) and does not exploit the cooperative setting of our problem. The authors of [72] study multi-agent Markov potential games (more general than the team problem) and establishes finite convergence bounds; however, their algorithm does not handle the state-space exploration issue (which is a key element in our work) so their regret bound has an extra problem-dependent factor; besides, only convergence to local Nash Equilibria is shown, and there is no guarantee about attaining global optima (social welfare maximization in the game setting).

## 5.2 Preliminaries

In this section, we describe the settings of two-agent hierarchical bandit and hierarchical MDP, with the corresponding optimal regret benchmarks that can be attained by the CI approach. For an integer $n$, we define $n^+ = \max\{n, 1\}$.

### 5.2.1 Two-agent Hierarchical Bandits

Consider a two-agent MAB where the rewards are decided by the joint action of the two agents U1 (leader) and U2 (follower). Let $K$ and $L$ be the numbers of actions (which are arms in the context) of U1 and U2, respectively, and $T$ be the number of time steps. Without loss of generality, we assume that $K, L \leq T$. Under the hierarchical information structure, in round $t \in [T]$, U1 first chooses an action $k_t \in [K]$; after observing U1's action $k_t$, U2 then chooses another action $l_t \in [L]$. However, U1 cannot observe U2's action $l_t$. These two actions jointly generate a noisy reward $R_t(k_t, l_t) \in [0, 1]$ with expectation $\mu_{k_t,l_t}$, and both agents observe[3] $R_t$. The detailed problem setting is given in Protocol 5.1, which we call the "two-stage hierarchical bandit" (2HB) problem.

**Protocol 5.1** Two-stage Hierarchical Bandit

```
input: $K$ (#actions of U1), $L$ (#actions of U2), $\{\mu_{k,l}\}_{k\in[K],l\in[L]}$ (unknown to the agents).
for $t = 1, \ldots, T$ do
    U1 chooses $k_t \in [K]$.
    After receiving $k_t$, U2 chooses $l_t \in [L]$.
    U1 and U2 receive the reward $R_t(k_t, l_t)$ sampled from a distribution with mean $\mu_{k_t,l_t}$.
```

In particular, U2 knows U1's choice of $k_t$, but U1 does not know U2's choice of $l_t$. If U1 knew $l_t$ as well, this would be equivalent to the usual stochastic MAB problem with $KL$ arms where the decision-making is centralized as we described above.

For ease of presentation, we assume without loss of generality that the best action of U2 given any choice of U1 is indexed by 1, i.e., $\mu_{k,1} \geq \mu_{k,l}$ for all $k, l$; similarly, the best action of U1 is indexed by 1, i.e., $\mu_{1,1} \geq \mu_{k,1}$ for all $k$. Then the (common) goal of the agents is to minimize the pseudo-regret defined as follows:

$$\mathrm{Reg}(T) = \sum_{t=1}^{T} (\mu_{1,1} - \mu_{k_t,l_t}).$$

---

[3] Our analysis can straightforwardly handle a more general case where U1 and U2 receive different (independent) noisy copies of the reward with the same mean. For simplicity, we assume that they receive the same copy.

### 5.2.2 Two-agent Hierarchical MDPs

Consider a two-agent $H$-step finite-horizon MDP where the rewards and state transitions depend on the joint action of the two agents U1 and U2, with the process run over $T$ episodes. This generalizes the previous two-agent bandit setting. The state space is $\mathcal{S}$, with a number of $S = |\mathcal{S}|$ states. In each state, U1 and U2 choose actions from $[A]$ and $[B]$, respectively. We assume that $S, A, B, H$ are all upper bounded by $T$. Every episode $t$ starts with an initial state $s_{t,1} \in \mathcal{S}$. In the $h$-th step of the $t$-th episode, the agents first observe $s_{t,h} \in \mathcal{S}$. Under the hierarchical information structure, U1 chooses an action $a_{t,h} \in [A]$, followed by U2 choosing another action $b_{t,h} \in [B]$ upon seeing $a_{t,h}$. After the actions are chosen, both agents receive a reward $r_{t,h} \in [0, 1]$ with $\mathbb{E}[r_{t,h}] = R(s_{t,h}, a_{t,h}, b_{t,h})$, and then the state transitions to the next state $s_{t,h+1} \sim P(\cdot|s_{t,h}, a_{t,h}, b_{t,h})$. The episode ends right after the state transitions to $s_{t,H+1}$. In the RL setting we consider, rewards are commonly observed by both agents, but they do not know the reward function $R$ or the transition probability $P$. The detailed problem setting is given in Protocol 5.2.

```
Protocol 5.2 Two-stage Hierarchical MDP
input: $(\mathcal{S}, \mathcal{A} \times \mathcal{B}, P, R, H)$ (episodic MDP, $\mathcal{A} \times \mathcal{B}$ is the joint action space).
for $t = 1, \ldots, T$ do
    Initial state $s_{t,1}$ is arbitrarily given.
    for $h = 1, \ldots, H$ do
        Both agents observes the state $s_{t,h}$.
        U1 chooses $a_{t,h}$; after receiving $a_{t,h}$, U2 chooses $b_{t,h}$.
        Both agents receive the reward $r_{t,h} = R(s_{t,h}, a_{t,h}, b_{t,h})$.
        The state transits based on $s_{t,h+1} \sim P(\cdot|s_{t,h}, a_{t,h}, b_{t,h})$.
```

Note that the transition kernel and reward function are assumed to be time-invariant.

An $H$-step policy for U1 can be represented as $\pi^1 = \{\pi^1_1, \ldots, \pi^1_H\}$, where $\pi^1_h : \mathcal{S} \to [A]$ specifies the choice of her action on each state when she is at step $h$; a policy for U2 can be represented as $\pi^2 = \{\pi^2_1, \ldots, \pi^2_H\}$, where $\pi^2_h : \mathcal{S} \times [A] \to [B]$ specifies the choice of his action on each state and under each possible choice of U1, when he is at step $h$. We define the state value function at step $h$ under a policy pair $(\pi^1, \pi^2)$ as

$$V_h^{\pi^1,\pi^2}(s) = \mathbb{E}\left[\sum_{k=h}^{H} R(s_k, a_k, b_k) \;\middle|\; s_h = s, a_k = \pi^1_k(s_k), b_k = \pi^2_k(s_k, a_k), s_{k+1} \sim P(\cdot|s_k, a_k, b_k), \forall k \geq h\right].$$

with $V_{H+1}^{\pi^1,\pi^2}(\cdot) \triangleq 0$. Also, we define the state-action value function as

$$Q_h^{\pi^1,\pi^2}(s,a,b) = R(s,a,b) + \mathbb{E}\left[ V_{h+1}^{\pi^1,\pi^2}(s_{h+1}) \middle| s_{h+1} \sim P(\cdot|s,a,b) \right].$$

The optimal value functions are then given by $V_{*,h}(s) = \max_{\pi^1,\pi^2} V_h^{\pi^1,\pi^2}(s)$ and $Q_{*,h}(s,a,b) = \max_{\pi^1,\pi^2} Q_h^{\pi^1,\pi^2}(s,a,b)$. By dynamic programming, we have the following for all $h, s, a, b$:

$$V_{*,h}(s) = \max_{a,b} Q_{*,h}(s,a,b) \quad \text{and} \quad Q_{*,h}(s,a,b) = R(s,a,b) + \mathbb{E}_{s' \sim P(\cdot|s,a,b)}\left[V_{*,h+1}(s')\right],$$

with $V_{*,H+1}(\cdot) \triangleq 0$. We further define $Q_{*,h}(s,a) = \max_b Q_{*,h}(s,a,b)$. With this notation, we can write the regret of the agents as

$$\operatorname{Reg}(T) = \sum_{t=1}^{T} \left( V_{*,1}(s_{t,1}) - V_1^{\pi_t^1,\pi_t^2}(s_{t,1}) \right).$$

### 5.2.3 Regret Benchmarks

In two-agent cases, there are three obvious types of information structure in terms of *action information asymmetry*: the complete information setting, the no information setting, and the hierarchical setting considered in this paper.

- Sequential decision making: U1 first chooses $a_t \in [A]$ (or $k_t \in [K]$ for the bandit setting); after observing $a_t$, U2 then chooses $b_t \in [B]$ (or $l_t \in [L]$ for the bandit setting, same for the rest of this subsection). This is the hierarchical information structure considered in this paper.

- Simultaneous decision making: U1 and U2 choose $a_t \in [A]$ and $b_t \in [B]$ simultaneously, respectively. Depending on their respective feedback afterwards, the setting can be further divided as follows:
  - Complete information sharing: both agents observe $(a_t, b_t)$ directly after they make their choices.
  - No information sharing: both agents do not observe $(a_t, b_t)$. This setting is considered in [30].

Note that in the setting of sequential decision making, U1 also does not observe $b_t$. Otherwise, it will be identical to the setting of complete information sharing in the learning context since after the time-step ends both agents will know $(a_t, b_t)$. Using the CI approach, one may achieve the lower regret bounds of $\widetilde{\mathcal{O}}(\sqrt{KLT})$ for the bandit setting and $\widetilde{\mathcal{O}}(\sqrt{H^2SABT})$ for the MDP setting, with higher complexity and stronger assumptions, which we now explain.

In the complete information sharing setting, it is evident that the agents may treat the joint action space $[A] \times [B]$ as the new action space and learns as if a single agent (the fictitious coordinator) is learning the policy of choosing the joint actions. The learning is centralized as there is no information asymmetry and the non-stationarity issue will not happen. Interestingly, the same approach can be carried through in the other two information structures as well. Suppose U1 learns with algorithm $\mathsf{ALG}^1$ (which should also include any possible tie-breaking rule) and randomization seed $\mathfrak{R}^1$, and U2 learns with $\mathsf{ALG}^2$ and randomization seed $\mathfrak{R}^2$. Suppose both agents have the information of $(\mathsf{ALG}^1, \mathfrak{R}^1, \mathsf{ALG}^2, \mathfrak{R}^2)$. In step 1, U2 can generate $a_1$ from $(\mathsf{ALG}^1, \mathfrak{R}^1)$, and U1 can generate $b_1$, so that $(a_1, b_1)$ becomes CI. Going forward, in step $t$, since $\mathcal{I}_{t-1} = (a_{1:t-1}, b_{1:t-1}, r_{1:t-1})$ (where $a_{1:t-1} = (a_1, \ldots, a_{t-1})$, etc.) is CI, U2 can reproduce $a_t$ from $\mathsf{ALG}^1(\mathcal{I}_{t-1}, \mathfrak{R}^1)$, and U1 can reproduce $b_t$ from $\mathsf{ALG}^2(\mathcal{I}_{t-1}, \mathfrak{R}^2)$, so that $(a_t, b_t)$ is again CI. We can see that with this approach there will be no information asymmetry. Clearly, if U1 and U2 treat $[A] \times [B]$ from the coordinator's perspective and adopt the same single-agent algorithm $\mathsf{ALG}^1 = \mathsf{ALG}^2$ with near-optimal regret guarantee and the same randomization device $\mathfrak{R}^1 = \mathfrak{R}^2$, the problem is equivalent to learning in the standard single-agent $AB$-armed bandit or standard single-agent MDP with action space being $[A] \times [B]$. Using the state of the art algorithms, i.e., UCB1 [5] for the bandit setting, and UCBVI algorithm (with a Bernstein bonus design) [7] (model-based) or the UCB-Advantage algorithm [137] (model-free), one may achieve the lower regret bounds of $\widetilde{\mathcal{O}}(\sqrt{KLT})$ for the bandit setting and $\widetilde{\mathcal{O}}(\sqrt{H^2SABT})$ for the MDP setting.

Both agents knowing $(\mathsf{ALG}^1, \mathfrak{R}^1, \mathsf{ALG}^2, \mathfrak{R}^2)$ and being able to reproduce each other's computation is a strong assumption. In the case of hierarchical information structure, simpler and more efficient alternatives presented in this chapter are possible.

## 5.3 Learning Hierarchical Bandits

Since U1 does not observe U2's actions, it is unclear how U1 can utilize or interpret the samples she receives. For example, if U1 receives a low reward, one possibility is that U1 has chosen a bad action, so whatever action U2 chooses, the reward is going to be low; but it is also possible that the action chosen by U1 is actually good (i.e., the reward would be high if U2 chose a good subsequent action), but U2 has chosen a bad subsequent action. If the identity of U2's action is not revealed, in general, U1 cannot distinguish between these two cases.

In Algorithm 5.3 below, we propose a modified UCB algorithm [2, 5] for the 2HB problem, and show that it achieves near-optimal regret bounds similar to those of UCB for both gap-independent and gap-dependent cases. At time $t$, define

$$\hat{\mu}_t^1(k) \triangleq \frac{1}{n_t^1(k)} \sum_{s=1}^{t-1} R_s(k_s, l_s) \mathbb{I}\{k_s = k\} \tag{5.1}$$

to be the empirical mean of the $k$-th arm in the first level based on it being played $n_t^1(k) \triangleq \sum_{s=1}^{t-1} \mathbb{I}\{k_s = k\}$ times until $t-1$, and define

$$\hat{\mu}_t^2(k, l) \triangleq \frac{1}{n_t^2(k, s)} \sum_{s=1}^{t-1} R_s(k_s, l_s) \mathbb{I}\{k_s = k, l_s = l\} \tag{5.2}$$

to be the empirical mean of the $(k, l)$-th arm based on it being played $n_t^2(k, l) \triangleq \sum_{s=1}^{t-1} \mathbb{I}\{k_s = k, l_s = l\}$ times up until $t-1$.

**Algorithm 5.3** Two-stage UCB

```
input: $K$ (#actions of U1), $L$ (#actions of U2).
for $t = 1, \ldots, KL$ do
    U1 and U2 play each combination of arms $(k, l)$ once, where $k \in [K]$ and $l \in [L]$.
for $t = KL + 1, \ldots, T$ do
    U1 chooses $k_t = \arg\max_{k \in [K]} \ \hat{\mu}_t^1(k) + 11\sqrt{\frac{L \log(LT/\delta)}{n_t^1(k)}}$.
    U2 chooses $l_t = \arg\max_{l \in [L]} \ \hat{\mu}_t^2(k_t, l) + 2\sqrt{\frac{\log(LT/\delta)}{n_t^2(k_t, l)}}$ after receiving $k_t$.
```

The algorithm ensures that the agents converge to playing optimal actions while keeping U1 agnostic to U2's actions during the whole process. The key observation is that when U2 is a no-regret learning agent (i.e., it learns with sub-linear-in-$T$ regret), his choices of action under a given action of U1 will converge to the best one, hence avoiding the second possibility mentioned above in the long run.

Despite both agents learning simultaneously in Algorithm 5.3, U1 explores arms more via a higher UCB so that U2 can discover the best arm conditioned on U1's arm being fixed. This allows us to achieve close-to-optimal regret as summarized in the theorems below. Before presenting one of the main results – the gap-independent regret bound for Algorithm 5.3 – we first state a lemma that is key to the proof and gives intuition of the design of the algorithm in the beginning.

**Lemma 5.1:** *For any* $t \in [T] \setminus [KL]$, $k \in [K]$, *with probability at least* $1 - 2\delta$,

$$\sum_{s=1}^{t} \mathbb{I}\{k_s = k\} \left(\mu_{k,1} - R_s(k, l_s)\right) \leq 11\sqrt{Ln_t^1(k) \log(Ln_t^1(k)/\delta)} \leq 11\sqrt{Ln_t^1(k) \log(LT/\delta)}. \tag{5.3}$$

**Proof:** By the standard $K$-armed UCB regret bound given in Appendix D.1 for the second layer when the $k$-th arm is selected in the first layer. ■

Lemma 5.1 can be thought of as the "regret of lower (second) layer" given the upper (first) layer chooses certain arm $k$. Note that dividing both sides of the inequality in (5.3) by $n_t^1(k)$ gives the following inequality

$$\mu_{k,1} \leq \frac{\sum_{s=1}^{t-1} R_s(k_s, l_s)\mathbb{I}\{k_s = k\}}{n_t^1(k)} + 11\sqrt{\frac{L \log(LT/\delta)}{n_t^1(k)}} = \hat{\mu}_t^1(k) + 11\sqrt{\frac{L \log(LT/\delta)}{n_t^1(k)}} \tag{5.4}$$

holds with probability at least $1 - 2\delta$. This inequality says for any upper layer arm $k$, the actual (mean) value even if the lower layer is playing optimally, will still be not larger than U1's optimistic value (which is the sum of empirical value and the bonus term) of the arm with high probability. The actual value of the best arm, $\mu_{1,1}$, will then be not larger than U1's optimistic value of the arm it actually selects with high probability, since it always chooses the arm with the maximal optimistic value. Having bounded $\mu_{1,1}$ by what U1 actually chooses in the execution of the algorithm, the remaining terms can be bounded by concentration inequalities. The inequality (5.4) is the reason why we use the $11\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k)}}$ bonus term for U1 in

Algorithm 5.3; the bonus term for U2 directly follows from standard UCB. We now give the gap-independent regret bound in the following theorem.

**Theorem 5.2:** *For the 2HB problem given in Protocol 5.1, with probability of at least* $1-\delta$*, Algorithm 5.3 achieves the regret bound of*

$$\mathrm{Reg}(T) \leq O\left(\sqrt{KLT\log(LT/\delta)}\right). \tag{5.5}$$

**Proof:** See Appendix D.2. ■

We remark that the bound in (5.5) is slightly smaller than coordinator's $KL$-armed UCB previously described by a factor of $\sqrt{\log(K)}$; this reduction follows from the feature of *sequential decision making* in the hierarchical information structure. However, the constant term when we remove the big-$O$ notation (which is 74 in this case, see Lemma 5.5) is larger than that of standard UCB (which is 11). This is what the information asymmetry introduces, as without the knowledge of U2's actions, U1 only tries to learn slower in the slow time-scale and gives U2 more time to explore and find the optimal arms for the lower layer first by using a larger bonus term. Note that the difference is trivial since we consider the setting of $T \gg K, L$, so that $\log(KLT)/\log(LT) \approx 1$. Hence, with this fact, all the $\log(KL)$ terms can be removed under the big-$O$ notation.

Another benefit of performing hierarchical UCB is that now U1 still performs the $K$-armed UCB (only with a different bonus term), in contrast to the scenario of coordinator's $KL$-armed UCB where both agents have to perform $KL$-armed UCB. The memory storage and computational complexity requirements for U1 are then reduced. This benefit becomes prominent when U1 resource-limited node in the system, and when the depth of the hierarchical becomes larger (see Section 5.4).

In the following, we present the result of gap-dependent regret bound; we again start by giving a key lemma characterizing the lower layer regret similar to Lemma 5.1.

**Lemma 5.3:** *Let* $k \neq 1$ *be a sub-optimal arm of U1. Then with probability at least* $1-2\delta$,

$$\sum_{t=1}^{T} \mathbb{I}\{k_t = k\} \leq O\left(\frac{L\log(LT/\delta)}{(\mu_{1,1}-\mu_{k,1})^2}\right).$$

**Proof:** See Appendix D.2. ■

**Theorem 5.4:** *For the 2HB problem given in Protocol 5.1, with probability of at least $1-\delta$, Algorithm 5.3 achieves the regret bound of*

$$\operatorname{Reg}(T) \leq L \sum_{k\in[K]\setminus\{1\}} O\left(\frac{\log(LT/\delta)}{\mu_{1,1}-\mu_{k,1}}\right) + \sum_{k\in[K]} \sum_{l\in[L]\setminus\{1\}} O\left(\frac{\log(LT/\delta)}{\mu_{k,1}-\mu_{k,l}}\right). \tag{5.6}$$

**Proof:** See Appendix D.2. ■

## 5.4 Wider and Deeper Hierarchical Bandits

In this section, we generalize the hierarchical bandit problem to be wider and larger. Specifically, in Section 5.4.1, we consider there are multiple agents in the same layer so that the *width* of the bandit problem is increased; in Section 5.4.2, we consider there are multiple layers instead of only two in the problem so that the *depth* of the bandit problem is increased.

### 5.4.1 Wider Hierarchical Bandits

We extend the 2HB problem to the scenario where there are multiple agents simultaneously decide their actions in each of the two stages. For notation simplicity we assume the numbers of agents in the two stages are the same $W$, the width of the problem. The agents indexed by $w \in [W]$ in the upper layer choose their actions $k_t^w$ simultaneously; and after receiving the action tuple from the upper layer, the agents in the lower layer again indexed by $w \in [W]$ choose their actions $l_t^w$ simultaneously. The problem setting is given below.

**Protocol 5.4** Two-stage Wide Hierarchical Bandit

```
input: $K = K^1 \times \cdots \times K^W$ (#actions of Group 1), $L = L^1 \times \cdots \times L^W$ (#actions of Group 2), $\{\mu_{k,l}\}_{k\in[K],l\in[L]}$ (unknown to the agents).
for $t = 1, \ldots, T$ do
    All agents $w \in [W]$ in group 1 choose $k_t^w \in [K^w]$ simultaneously.
    All agents $w \in [W]$ in group 2 receive $k_t = (k_t^1, \ldots, k_t^W)$ and choose $l_t^w \in [L^w]$ simultaneously.
    Action tuple formed by group 2 $l_t = (l_t^1, \ldots, l_t^W)$ is not seen by group 1.
    All agents receive the reward $R_t(k_t, l_t)$, which is sampled from $Ber(\mu_{k_t,l_t})$.
```

In the problem, there are two smaller groups, i.e. group 1 consisting of the agents in the upper layer, and group 2 consisting of the agents in the lower layer. There is no information asymmetry within each group. Hence, this corresponds to the first special case we discussed in Section 5.3, where the equivalent virtual coordinator has the complete information and chooses tuples of actions as its equivalent actions. Treating the coordinator of group 1 as U1 with $k_t = (k_t^1, \ldots, k_t^W) \in [K]$ as its form of actions, and the coordinator of group 2 as U2 with $l_t = (l_t^1, \ldots, l_t^W) \in [L]$ as its form of actions in the 2HB problem, we can again solve it with the two-stage UCB in Algorithm 5.3, leading to a (gap-independent) regret of

$$O\left(\sqrt{KLT\log(LT/\delta)}\right) = O\left(\sqrt{K^1 \times \cdots \times K^W L^1 \times \cdots \times L^W T \log(L^1 \times \cdots \times L^W T/\delta)}\right).$$

Consider a simple case where $K^1 = \cdots = K^W = \bar{K}$ and $L^1 = \cdots = L^W = \bar{L}$, that is, assuming the number of actions are the same for all agents within each group. Then the regret bound can be simplified to $O\left(\sqrt{\bar{K}^W \bar{L}^W T \log(\bar{L}^W T/\delta)}\right)$. We can see that the regret grows exponentially in the width $W$. While the exponential growth is definitely unfavorable, it is also inevitable due to the complex setting – an “arm” is one complete action tuple in this setting, and without any further assumption or constraint made for $\{\mu_{k,l}\}_{k\in[K],l\in[L]}$, the number of arms just grows exponentially in $W$, and the agents have to explore all of them to find the optimal one.

### 5.4.2 Deeper Hierarchical Bandits

In this subsection, we generalize the 2HB problem in another direction. We consider there are $D$ layers of the problem with the hierarchical information structure, that is, upper layers will decide first, and their choices will be seen by lower layers. For notation simplicity we assume the number of actions $K$ is the same for each layer, and each layer only contains one agent. The problem setup is given in Protocol 5.5.

An algorithm similar to the idea of Algorithm 5.3 can be designed to solve the multi-stage hierarchical bandit problem near-optimally. At time $t$ and layer $d$, agent $d$ maintains the number of times it has visited an arm $\mathbf{k}^{1:d} = (k^1, \ldots, k^d) \in [K]^d$

$$n_t^d(\mathbf{k}^{1:d}) = \sum_{s=1}^{t-1} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \tag{5.7}$$

```
Protocol 5.5 Multi-stage Hierarchical Bandit
input: $K$ (#actions of each agent), $D$ (#layers), $\{\mu_{\mathbf{k}}\}_{\mathbf{k}\in[K]^D}$ (unknown to the agents).
for $t = 1, \ldots, T$ do
    Agent 1 chooses $k_t^1 \in [K]$.
    for $d = 2, \ldots, D$ do
        After receiving $(k_t^1, \ldots, k_t^{d-1})$, agent $d$ chooses $k_t^d \in [K]$.
    Based on the joint action $\mathbf{k}_t := (k_t^1, \ldots, k_t^D)$, all agents receive the reward $R_t(\mathbf{k}_t)$,
    which is sampled from $Ber(\mu_{\mathbf{k}_t})$.
```

and the empirical mean of the same arm

$$\hat{\mu}_t^d(\mathbf{k}^{1:d}) = \frac{1}{n_t^d(\mathbf{k}^{1:d})} \sum_{s=1}^{t-1} R_s(k_s^{1:D}) \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\}. \tag{5.8}$$

The bonus term of agent $d$ for the arm $\mathbf{k}^{1:d}$ is $\sqrt{\frac{K^{D-d}\log(K^D T/\delta)}{n_t^d(\mathbf{k}^{1:d})}}$, which is again the regret from its direct subordinate layer $d+1$ divided by $n_t^d(\mathbf{k}^{1:d})$. Just as Section 5.3, such a design is to ensure that agent $d$'s optimistic value is not smaller than the value of the optimal arm (with high probability). The multi-stage UCB algorithm we propose to solve the multi-stage hierarchical bandit problem is given in Algorithm 5.6.

```
Algorithm 5.6 Multi-stage UCB
input: $K$ (#actions of each agent), $D$ (#layers).
for $t = 1, \ldots, K^D$ do
    Agents play each combination of arms $\mathbf{k} = (k^1, \ldots, k^D)$ once, where $k^d \in [K]$ for
    all $d \in [D]$.
for $t = K^D, \ldots, T$ do
    for $d = 1, \ldots, D$ do
        Agent $d$ chooses $k_t^d = \arg\max_{k\in[K]} \hat{\mu}_t^d(k_t^1, \ldots, k_t^{d-1}, k) + C_d \sqrt{\frac{K^{D-d}\log(K^D T/\delta)}{n_t^d(k_t^1, \ldots, k_t^{d-1}, k)}}$, where
        $C_D \geq 2$ and $C_d \geq 6C_{d+1} + 8$.
```

In the following, we present the near-optimal regret bounds the multi-stage UCB algorithm achieves. Lemma 5.5 and Theorem 5.6 are the gap-independent results, where Lemma 5.5 is a generalization (to cases of multiple stages) of Lemma 5.1 and Theorem 5.6 is a generalization of Theorem 5.2. Lemma 5.7 and Theorem 5.8 are the gap-dependent results, where Lemma 5.7 is a generalization of Lemma 5.3 and Theorem 5.8 is a generalization of Theorem 5.4. In the bounds, the regret again grows exponentially in the depth $D$; just as Section 5.4.1, this comes from the high model complexity and is unavoidable.

**Lemma 5.5:** *For any* $t \in [T] \setminus [K^D]$, $d \in [D-1] \cup \{0\}$, $\mathbf{k}^{1:d} = (k^1, \ldots, k^d) \in [K]^d$,

$$\begin{aligned}
&\sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \left[\mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} - R_s(\mathbf{k}_s)\right] \\
\leq &\begin{cases} C_d \sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D n_t^d(\mathbf{k}^{1:d})/\delta)}, & \text{when } d > 0, \\ (6C_1 + 8)\sqrt{K^D t \log(K^D t/\delta)}, & \text{when } d = 0, \end{cases} \\
\leq &\begin{cases} C_d \sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta)}, & \text{when } d > 0, \\ (6C_1 + 8)\sqrt{K^D t \log(K^D T/\delta)}, & \text{when } d = 0. \end{cases}
\end{aligned} \tag{5.9}$$

**Proof:** See Appendix D.3. ■

**Theorem 5.6:** *For the multi-stage hierarchical bandit problem given in Protocol 5.5, with probability of at least* $1-\delta$, *Algorithm 5.6 achieves the regret bound of*

$$\sum_{t=1}^{T} [\mu_{\mathbf{1}_D} - R_t(k_t^1, \ldots, k_t^D)] \leq O\left(\sqrt{K^D D T \log(KT/\delta)}\right). \tag{5.10}$$

**Proof:** It is a special case of Lemma 5.5 with $d = 0$ and $t = T$. ■

**Lemma 5.7:** *For any* $d \in [D]$, *let* $k^d \in [K]$ *be a sub-optimal arm of agent* $d$ *given that* $\mathbf{k}^{1:d-1}$ *is chosen by the first* $d-1$ *agents, then with probability at least* $1-2\delta$,

$$\sum_{t=1}^{T} \mathbb{I}\{\mathbf{k}_t^{1:d} = \mathbf{k}^{1:d}\} \leq \frac{4(C_d+2)^2 K^{D-d} \log(K^D T/\delta)}{\left[\mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})}\right]^2}. \tag{5.11}$$

**Proof:** See Appendix D.3. ■

**Theorem 5.8:** *For the multi-stage hierarchical bandit problem given in Protocol 5.5, with probability of at least* $1-\delta$, *Algorithm 5.6 achieves the regret bound of*

$$\sum_{t=1}^{T} [\mu_{\mathbf{1}_D} - R_t(k_t^1, \ldots, k_t^D)] \leq \sum_{d=1}^{D} \sum_{\mathbf{k}^{1:d} \neq \mathbf{1}_d} O\left(\frac{K^{D-d} D \log(KDT/\delta)}{\mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})}}\right). \tag{5.12}$$

**Proof:** See Appendix D.3. ■

## 5.5 Hierarchical MDP

This setting is much more challenging than the hierarchical bandit setting. First, notice that in this setting, both agents are facing non-stationary transition and reward because of the dependence of these quantities on the policy of the other agent, which varies with time. Obtaining regret bounds in such time-varying MDPs is in general hard [1, 103, 120], except for problems with special structures or extra assumptions [103, 120, 72] like our case here. Second, notice that in the bandit case, given any choice of U1, U2 is essentially facing a stationary MAB problem, and thus we can directly apply existing theorems for standard MAB; however, in the MDP case, the world that U2 sees on a certain step is still affected by the non-stationarity of U1's policies in future steps. In this case, standard analysis for stationary MDPs cannot be directly applied.

An initial idea to deal with this setting is to let both agents run existing UCB-based algorithms (e.g., UCBVI [7], UCB-H [57]) with an increased bonus term for U1 to compensate for the regret of U2, imitating our hierarchical bandit solution. However, as we point out above, U2's world is also affected by the policies of U1 in future steps. Therefore, a natural solution is to do the following: besides letting U1 add extra bonus to compensate the regret of U2, we also let U2 add extra bonus to compensate the regret of U1 in future steps. Unfortunately, for this hypothetical algorithm, it is unclear to us how to obtain a regret bound that is polynomial in the number of steps $H$. This is because by recursively adding extra bonus in each layer, we end up with a factor of $(AB)^{H/2}$ in the regret bound, similar to the "Deep Hierarchy Case" discussed in Section 5.4.2 and Appendix D.3.

To address this issue, instead of trying to let U2 best respond to the non-stationary world created by U1, we exploit the fact that U2 has full knowledge about the joint action space, and let U2 find the best *joint policy* of U1 and U2. Then U2 will execute his part of this joint policy even though U1 may not follow it. Although this brings other issues (discussed later), it avoids the need of U2 to compensate for the regret of U1 in later steps, and prevents the exponential blowup in the regret bound.

Our algorithm for hierarchical MDPs is presented in Algorithm 5.7. To avoid cluttered notation, we drop the episode index $t$ when presenting the algorithm.

**Algorithm 5.7** UCB-H/UCBVI for Hierarchical MDP

```
define: $\alpha_\tau = \frac{H+1}{H+\tau}$, $\mathsf{bns}^1_\tau = c'\sqrt{\frac{H^3 SB\log(T/\delta)}{\tau^+}}$, $\mathsf{bns}^2_\tau = c\sqrt{\frac{H^2 S\log(T/\delta)}{\tau^+}}$ where $c, c' \geq 1$ are universal constants.
initialize: $Q^1_h(s,a), Q^2_h(s,a,b) \leftarrow H \quad \forall h,s,a,b$,
$n_h(s,a), n_h(s,a,b), n_h(s,a,b,s'), \theta_h(s,a,b) \leftarrow 0 \quad \forall h,s,a,b,s'$.
for $t = 1, \dots, T$ do
    U1 and U2 observes $s_1$.
    for $h = 1, \dots, H$ do
        U1 chooses $a_h \in \mathrm{argmax}_a\, Q^1_h(s_h, a)$.
        U2 observes $a_h$.
        U2 chooses $b_h \in \mathrm{argmax}_b\, Q^2_h(s_h, a_h, b)$.
        U1 and U2 observe $r_h$ and $s_{h+1}$.
        U1 updates counts of visits: $n_h(s_h, a_h) \overset{+}{\leftarrow} 1$.  (“$\overset{+}{\leftarrow} 1$” means to increase the number by 1.)
        U2 updates counts of visits: $n_h(s_h, a_h, b_h) \overset{+}{\leftarrow} 1$, $n_h(s_h, a_h, b_h, s_{h+1}) \overset{+}{\leftarrow} 1$.
        U2 updates cumulative reward: $\theta_h(s_h, a_h, b_h) \overset{+}{\leftarrow} r_h$.

    U1 updates $Q/V$ functions ($\approx$ UCB-H update rule):
    $V^1_{H+1}(\cdot) \leftarrow 0$.
    for $h = 1, \dots, H$ do
        $Q^1_h(s_h, a_h) \leftarrow (1-\alpha_\tau) Q^1_h(s_h, a_h) + \alpha_\tau \left(r_h + V^1_{h+1}(s_{h+1}) + \mathsf{bns}^1_\tau\right)$
        $V^1_h(s_h) \leftarrow \min\{\max_a Q^1_h(s_h, a),\ H\}$
        where $\tau = n_h(s_h, a_h)$.

    U2 updates $Q/V$ functions ($\approx$ UCBVI update rule):
    Let $\hat{P}_h(s'|s,a,b) = \frac{n_h(s,a,b,s')}{n_h(s,a,b)}$ and $\hat{R}_h(s,a,b) = \frac{\theta_h(s,a,b)}{n_h(s,a,b)} \quad \forall h,s,a,b,s'$.
    (if $n_h(s,a,b) = 0$, set $\hat{P}_h(s'|s,a,b) = \frac{1}{|\mathcal{S}|}$ and $\hat{R}_h(s,a,b) = 0$).
    $V^2_{H+1}(\cdot) \leftarrow 0$.
    for $h = H, \dots, 1$ do
        for all $s, a, b$ do
            $Q^2_h(s,a,b) \leftarrow \min\left\{\hat{R}_h(s,a,b) + \mathbb{E}_{s' \sim \hat{P}_h(\cdot|s,a,b)}\left[V^2_{h+1}(s')\right] + \mathsf{bns}^2_\tau,\ \ Q^2_h(s,a,b)\right\}$
            $V^2_h(s) \leftarrow \max_{a,b} Q^2_h(s,a,b)$
            where $\tau = n_h(s,a,b)$.
```

In Algorithm 5.7, U1 maintains optimistic value function estimators $V_h^1(s), Q_h^1(s,a)$, and U2 maintains $V_h^2(s), Q_h^2(s,a,b)$ for every $h = 1, \dots, H$. Their constructions are based on two standard UCB-based algorithms. Specifically, the constructions of $V_h^1(s)$ and $Q_h^1(s,a)$ (17-18) are similar to those of UCB Q-learning [57], with the bonus term $\mathsf{bns}_\tau^1$ enlarged by a factor of $\sqrt{SB}$. Like in the bandit setting from Section 5.3, U1 faces a non-stationary environment, and the extra $\sqrt{B}$ factor is used to compensate for the regret of U2 in future steps[4]. On the other hand, the constructions of $V_h^2(s)$ and $Q_h^2(s,a,b)$ (26-27) are similar to those of UCBVI [7]. In particular, $V_h^2(s)$ is obtained by jointly optimizing over the actions of U1 and U2 (27), conforming to our previous discussions.

Perhaps the most intriguing facet is why we use UCB-H for U1 but UCBVI for U2. From a high level, this is because UCB-H shrinks its confidence set of value functions at a slower rate, while UCBVI is faster, which fulfills our need that U1 has to explore more in the early stages, for U2 to have enough time to find his optimal policy. Recall that the value iteration performed by U2 is through $V_h^2(s) \leftarrow \max_{a,b} Q_h^2(s,a,b)$ (27). By the optimism principle, ideally we would like the agents to take actions $(a_h, b_h) = \operatorname{argmax}_{a,b} Q_h^2(s_h, a, b)$ to facilitate exploration. However, since U2 cannot control the actions taken by U1, and there is no communication between U1 and U2, it is unclear whether the optimism principle on $Q_h^2$ can be successfully carried out (the best U2 can do is to take $b_h = \operatorname{argmax}_b Q_h^2(s_h, a_h, b)$ for some $a_h$ taken by U1, as done in 9). Our key finding is that if $Q_h^1(s,a)$ always upper bounds $\max_b Q_h^2(s,a,b)$, then the agents can still perform adequate joint exploration without explicit coordination. This key property can be shown straightforwardly if we use UCB-H for U1 and UCBVI for U2 (see the proof of Lemma 5.11).

Below, we establish some lemmas to be used in the regret bound analysis. The detailed proofs are deferred to Appendix D.4. We first define new notation with the episode indices.

**Definition 5.9:** *Let $Q_{t,h}^1(\cdot,\cdot)$, $Q_{t,h}^2(\cdot,\cdot,\cdot)$ be the $Q_h^1(\cdot,\cdot)$, $Q_h^2(\cdot,\cdot,\cdot)$ at the beginning of episode $t$ in Algorithm 5.7. Let $s_{t,h}, a_{t,h}, b_{t,h}, r_{t,h}$ be the $s_h, a_h, b_h, r_h$ within episode $t$ in Algorithm 5.7.*

---

[4] The extra $\sqrt{S}$ factor arises from a technical difficulty, and we are unsure whether it is necessary.

Lemma 5.10 below shows the optimism of U2's $Q$-function estimator, and relates the cumulative sum of $Q^2_{t,h}(s_{t,h}, a_{t,h}, b_{t,h}) - Q_{*,h}(s_{t,h}, a_{t,h}, b_{t,h})$ to that of $V^2_{t,h+1}(s_{t,h+1}) - V_{*,h+1}(s_{t,h+1})$. The proof is standard and we provide it in Appendix D.4 for completeness.

**Lemma 5.10:** *With probability at least* $1 - \mathcal{O}(\delta)$, $Q^2_{t,h}(s,a,b) \geq Q_{*,h}(s,a,b)$ *for all* $t, h, s, a, b$, *and*

$$\sum_{t=1}^{T} \left(Q^2_{t,h}(s_{t,h}, a_{t,h}, b_{t,h}) - Q_{*,h}(s_{t,h}, a_{t,h}, b_{t,h})\right) \leq \sum_{t=1}^{T} \left(V^2_{t,h+1}(s_{t,h+1}) - V_{*,h+1}(s_{t,h+1})\right) + \widetilde{\mathcal{O}}\left(\sqrt{H^2S^2ABT}\right), \quad \forall h.$$

**Proof:** See Appendix D.4. ■

We remark that it is possible to improve the bound in Lemma 5.10 by a factor of $\sqrt{S}$ by using the more refined analysis in [7] and defining $\mathsf{bns}^2_\tau$ to be a $\sqrt{S}$-factor smaller. However, as we will see below, this improvement will not lead to a better final regret bound, so in Lemma 5.10 we opt to use a simpler analysis with a looser bound.

Next, we establish our key lemma, Lemma 5.11, which states that U1 has *more optimism* than U2. In this lemma we have to make $\mathsf{bns}^1_\tau$ a $\sqrt{SB}$-factor larger than that in [57]. While the $\sqrt{B}$ factor is necessary for the same reasons as in the bandit case, it is unclear whether the $\sqrt{S}$ factor is necessary. We leave the improvement of this factor as a future direction to explore.

**Lemma 5.11:** *With probability at least* $1 - \mathcal{O}(\delta)$, $Q^1_{t,h}(s,a) \geq \max_b Q^2_{t,h}(s,a,b)$ *for all* $t, h, s, a, b$.

**Proof:** See Appendix D.4. ■

Finally, in Lemma 5.12, we relate the cumulative sum of $Q^1_{t,h}(s_{t,h}, a_{t,h}) - Q^1_{*,h}(s_{t,h}, a_{t,h})$ to that of $V^1_{t,h+1}(s_{t,h+1}) - V^1_{*,h+1}(s_{t,h+1})$. The proof is similar to that of [57], but the bound is a $\sqrt{SB}$-factor larger than theirs due to the use of a larger bonus $\mathsf{bns}^1_\tau$.

**Lemma 5.12:** *With probability at least* $1-\mathcal{O}(\delta)$,

$$\sum_{t=1}^{T}\left(Q^1_{t,h}(s_{t,h},a_{t,h})-Q_{*,h}(s_{t,h},a_{t,h})\right)$$
$$\leq\left(1+\frac{1}{H}\right)\sum_{t=1}^{T}\left(V^1_{t,h+1}(s_{t,h+1})-V_{*,h+1}(s_{t,h+1})\right)+\widetilde{\mathcal{O}}\left(\sqrt{H^3S^2ABT}+HSA\right),\quad\forall h.$$

**Proof:** See Appendix D.4. ■

Thanks to the fact that $V^1_{t,h}(s_{t,h})=Q^1_{t,h}(s_{t,h},a_{t,h})$, Lemma 5.12 leads to the following simple corollary. Note that we do not have a similar corollary for Lemma 5.10 because $V^2_{t,h}(s_{t,h})\neq Q^2_{t,h}(s_{t,h},a_{t,h},b_{t,h})$ (as discussed earlier, $(a_{t,h},b_{t,h})$ is not necessarily equal to $\operatorname{argmax}_{a,b}Q^2_{t,h}(s_{t,h},a,b)$). The proof of the corollary is also in Appendix D.4.

**Corollary 5.13:** $\sum_{t=1}^{T}\left(Q^1_{t,h}(s_{t,h},a_{t,h})-Q_{*,h}(s_{t,h},a_{t,h})\right)=\widetilde{\mathcal{O}}\left(\sqrt{H^5S^2ABT}+H^2SA\right)$.

**Proof:** See Appendix D.4. ■

Finally, we are able to show our main theorem:

**Theorem 5.14:** *With probability* $1-\mathcal{O}(\delta)$, *Algorithm 5.7 guarantees*

$$\mathrm{Reg}(T)=\widetilde{\mathcal{O}}\left(H^{3.5}S\sqrt{ABT}+H^3SA\right).$$

**Proof:** We perform regret decomposition as follows:

$$\begin{aligned}
\mathrm{Reg}(T)&=\sum_{t=1}^{T}\left(V_{*,1}(s_{t,1})-V_1^{\pi^1_t,\pi^2_t}(s_{t,1})\right)\\
&=\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{s,a,b}\mathbb{P}\left[(s_{t,h},a_{t,h},b_{t,h})=(s,a,b)\;\middle|\;s_{t,1},\pi^1_t,\pi^2_t\right]\left(V_{*,h}(s)-Q_{*,h}(s,a,b)\right)\\
&\qquad\text{(by the performance difference lemma [60])}\\
&=\sum_{t=1}^{T}\sum_{h=1}^{H}\left(V_{*,h}(s_{t,h})-Q_{*,h}(s_{t,h},a_{t,h},b_{t,h})\right)+\widetilde{\mathcal{O}}\left(H\sqrt{HT}\right)\qquad\text{(by Lemma D.7)}\\
&=\sum_{t=1}^{T}\sum_{h=1}^{H}\left(V_{*,h}(s_{t,h})-Q_{*,h}(s_{t,h},a_{t,h})\right)\\
&\quad+\sum_{t=1}^{T}\sum_{h=1}^{H}\left(Q_{*,h}(s_{t,h},a_{t,h})-Q_{*,h}(s_{t,h},a_{t,h},b_{t,h})\right)+\widetilde{\mathcal{O}}\left(\sqrt{H^3T}\right).
\end{aligned}\tag{5.13}$$

Note that

$$\begin{aligned}
\mathrm{Reg}_h^1 &\triangleq \sum_{t=1}^{T}\left(V_{*,h}(s_{t,h}) - Q_{*,h}(s_{t,h},a_{t,h})\right) \le \sum_{t=1}^{T}\big(Q^1_{t,h}(s_{t,h},a_{t,h}) - Q_{*,h}(s_{t,h},a_{t,h})\big)\\
&\le \widetilde{\mathcal{O}}\left(\sqrt{H^5S^2ABT} + H^2SA\right), && \text{(by Corollary 5.13)}
\end{aligned}$$

where the first inequality is because Lemma 5.10, Lemma 5.11, and the way U1 chooses $a_{t,h}$ yield

$$V_{*,h}(s_{t,h}) = \max_{a,b} Q_{*,h}(s_{t,h},a,b) \le \max_{a,b} Q^2_{t,h}(s_{t,h},a,b) \le \max_{a} Q^1_{t,h}(s_{t,h},a) = Q^1_{t,h}(s_{t,h},a_{t,h}).$$

On the other hand,

$$\begin{aligned}
\mathrm{Reg}_h^2 &\triangleq \sum_{t=1}^{T}\left(Q_{*,h}(s_{t,h},a_{t,h}) - Q_{*,h}(s_{t,h},a_{t,h},b_{t,h})\right)\\
&\le \sum_{t=1}^{T}\big(Q^2_{t,h}(s_{t,h},a_{t,h},b_{t,h}) - Q_{*,h}(s_{t,h},a_{t,h},b_{t,h})\big)\\
&\le \sum_{t=1}^{T}\big(V^2_{t,h+1}(s_{t,h+1}) - V_{*,h+1}(s_{t,h+1})\big) + \widetilde{\mathcal{O}}\left(\sqrt{H^2S^2ABT} + H^2SA\right)\\
& && \text{(by Lemma 5.10)}\\
&\le \sum_{t=1}^{T}\big(V^1_{t,h+1}(s_{t,h+1}) - V_{*,h+1}(s_{t,h+1})\big) + \widetilde{\mathcal{O}}\left(\sqrt{H^5S^2ABT} + H^2SA\right)\\
& && \text{(by Lemma 5.11 and the definitions of } V^1_{t,h}(s) \text{ and } V^2_{t,h}(s))\\
&\le \sum_{t=1}^{T}\big(Q^1_{t,h+1}(s_{t,h+1},a_{t,h+1}) - Q_{*,h+1}(s_{t,h+1},a_{t,h+1})\big) + \widetilde{\mathcal{O}}\left(\sqrt{H^5S^2ABT} + H^2SA\right)\\
& && \text{(by the way U1 chooses } a_{t,h+1})\\
&\le \widetilde{\mathcal{O}}\left(\sqrt{H^5S^2ABT} + H^2SA\right), && \text{(by Corollary 5.13)}
\end{aligned}$$

where the first inequality follows from Lemma 5.10 and the way U2 chooses actions, resulting in $Q_{*,h}(s_{t,h},a_{t,h}) = \max_b Q_{*,h}(s_{t,h},a_{t,h},b) \le \max_b Q^2_{t,h}(s_{t,h},a_{t,h},b) = Q^2_{t,h}(s_{t,h},a_{t,h},b_{t,h})$.

Combining the bounds on $\mathrm{Reg}_h^1$ and $\mathrm{Reg}_h^2$ with (5.13) proves the theorem. ■

## 5.6 Example: Matching Problem

In this section, we study in detail the "matching problem," an example of a Dec-POMDP environment that was first introduced in Section 4.1.2.2. We present its structural results analyzed by decentralized stochastic control theory, and MARL solutions based on the structural results and the hierarchical UCB algorithm we developed in Algorithm 5.3. In the problem, two agents have imperfect measurements of the state, and are rewarded if both agents use their actions to "match" the state.

In Section 5.6.1, we derive an optimal policy when the model is known using the CI approach proposed by [88, 89]. We obtain a series of structural results using the certain features of the information structure of the matching problem, and finally conclude that there exists a time-invariant optimal policy for the problem. We give a full characterization of the optimal policy in Theorem 5.19 and Theorem 5.22. We then use this knowledge to develop a MARL methodology in Section 5.6.2 when the parameters of the model is unknown. We also discuss three generalizations of the model in Section 5.6.3; we give the optimal policy for one of them, with its MARL counterpart when the model is unknown, which involves Q-learning.

### 5.6.1 Decentralized Stochastic Control Solution: Structural Results and an Optimal Policy

We start by introducing the notation in Section 5.6.1.1. Then in Section 5.6.1.2, we apply the CI approach proposed by [88, 89] to the matching problem, determine the information state and the DP equations of the problem. The structural results rely on two important properties in our model: (i) whenever the two agents obtain the reward of 1, they know they correctly match the state in that time-step, and (ii) whenever U1 communicates the state to U2, then too both agents know the state before taking their action; in other words, there are times when the current state becomes CI. In Section 5.6.1.3, we exploit these two features to show that both the information states and the DP equations can be simplified. Finally, using the results and other properties of the matching problem, we characterize the optimal policy of the problem based on the parameters in Section 5.6.1.4.

#### 5.6.1.1 Notation

We treat the two sub-steps as two separate time steps when writing the DP equation. We denote everything regarding the first and second sub-steps with superscripts $^{\mathrm{i}}$ and $^{\mathrm{ii}}$, respectively, but both with the same subscript $t$ for the time-step index. When a sub-step index and agent index are used together, e.g. the second sub-step for U1, we write $^{\mathrm{ii},1}$, that is, the sub-step index comes first. Denote the tuple of private information at time $t$ as $M_t = (S_t, O_t, A_{t-1}) = (M_t^1, M_t^2)$, where $M_t^1 = (S_t, A_{t-1}^1)$ and $M_t^2 = (O_t, A_{t-1}^2)$ are the private information of U1 and U2, respectively; the $M_1$'s do not contain the $A$ components.

#### 5.6.1.2 The Common Information Approach

In the CI approach, the agents (or effectively the virtual coordinator) maintain the conditional probability of the private information and the state given the CI as the information state of the centralized POMDP, and solve the DP equation to find an optimal prescription function, which is an optimal strategy given the current information state. Based on this the first information state (in the first sub-step) is given by

$$\Pi_t^{\mathrm{i}}(m^t) = \mathbb{P}^{g_{1:t-1}}(M_{1:t} = m^t | Z_{1:t-1}, Y_{1:t-1}), \tag{5.14}$$

a distribution over $\Omega(M_{1:t})$ given $Z_{1:t-1}$, $Y_{1:t-1}$, and the strategies $g_{1:t-1}$. Similarly, the second information state (in the second sub-step) is

$$\Pi_t^{\mathrm{ii}}(m^t) = \mathbb{P}^{g_{1:t-1}, g_t^{\mathrm{i}}}(M_{1:t} = m^t | Z_{1:t}, Y_{1:t-1}), \tag{5.15}$$

a distribution over $\Omega(M_{1:t})$ given $Z_{1:t}$, $Y_{1:t-1}$, $g_{1:t-1}$, and the strategy $g_t^{\mathrm{i}}$. The strategy in the first sub-step maps U1's private information plus the information state to the communication decision

$$U_t = g_t^{\mathrm{i}}(M_{1:t}^1, \Pi_t^{\mathrm{i}}) := \gamma_t^{\mathrm{i}}(M_{1:t}^1), \tag{5.16}$$

where the prescription $\gamma_t^{\mathrm{i}}$ is the strategy given the information state $\Pi_t^{\mathrm{i}}$. The strategies in the second sub-step map U1 and U2's private information plus the information state to their actions

$$\begin{aligned} A_t^1 &= g_t^{\mathrm{ii},1}(M_{1:t}^1, \Pi_t^{\mathrm{ii}}) := \gamma_t^{\mathrm{ii},1}(M_{1:t}^1), \\ A_t^2 &= g_t^{\mathrm{ii},2}(M_{1:t}^2, \Pi_t^{\mathrm{ii}}) := \gamma_t^{\mathrm{ii},2}(M_{1:t}^2), \end{aligned} \tag{5.17}$$

where the prescriptions for U1 and U2, $\gamma_t^{\mathrm{ii},1}$ and $\gamma_t^{\mathrm{ii},2}$, are again their strategies given the information state $\Pi_t^{\mathrm{ii}}$. The prescriptions belong to the function spaces $\gamma_t^{\mathrm{i}} \in \Omega(\Omega(S_{1:t}) \times \Omega(A_{1:t-1}^1) \to \Omega(U_t))$, $\gamma_t^{\mathrm{ii},1} \in \Omega(\Omega(S_{1:t}) \times \Omega(A_{1:t-1}^1) \to \Omega(A_t^1))$, and $\gamma_t^{\mathrm{ii},2} \in \Omega(\Omega(O_{1:t}) \times \Omega(A_{1:t-1}^2) \to \Omega(A_t^2))$. We write $g_t := (g_t^{\mathrm{i}}, g_t^{\mathrm{ii}}) := (g_t^{\mathrm{i}}, g_t^{\mathrm{ii},1}, g_t^{\mathrm{ii},2})$ where $g_t^{\mathrm{ii}} := (g_t^{\mathrm{ii},1}, g_t^{\mathrm{ii},2})$ and also the corresponding $\gamma$'s.

The DP equations can then be written as

$$\begin{aligned} V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}) &= \sup_{\gamma_t^{\mathrm{i}}} \mathbb{E}\left[ -c\gamma_t^{\mathrm{i}}(M_{1:t}) + V_t^{\mathrm{ii}}(\eta_t^{\mathrm{i}}(\Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_t)) \middle| \, \Pi_t^{\mathrm{i}} = \pi_t^{\mathrm{i}} \right], \\ V_t^{\mathrm{ii}}(\pi_t^{\mathrm{ii}}) &= \sup_{\gamma_t^{\mathrm{ii}}} \mathbb{E}\left[ Y_t(S_t, \gamma_t^{\mathrm{ii}}(M_{1:t})) + \lambda V_{t+1}^{\mathrm{i}}(\eta_t^{\mathrm{ii}}(\Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Y_t)) \middle| \, \Pi_t^{\mathrm{ii}} = \pi_t^{\mathrm{ii}} \right], \end{aligned} \tag{5.18}$$

where $V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}})$ and $V_t^{\mathrm{ii}}(\pi_t^{\mathrm{ii}})$ are the value functions under an optimal policy for the information states in the first and the second sub-steps, and $\eta_t^{\mathrm{i}}(\Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_t)$ and $\eta_t^{\mathrm{ii}}(\Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Y_t)$ are the update rules for information states using Bayes rule namely,

$$\begin{aligned} &\Pi_t^{\mathrm{ii}}(m^t) = \eta_t^{\mathrm{i}}(\Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_t)(m^t) = \mathbb{P}(M_{1:t} = m^t | \Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_{1:t}, Y_{1:t-1}) \\ &= \frac{\mathbb{P}(M_{1:t} = m^t | \Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_{1:t-1}, Y_{1:t-1}) \mathbb{P}(Z_t | M_{1:t} = m^t, \Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_{1:t-1}, Y_{1:t-1})}{\sum_{m^{t\prime}} \mathbb{P}(M_{1:t} = m^{t\prime} | \Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_{1:t-1}, Y_{1:t-1}) \mathbb{P}(Z_t | M_{1:t} = m^{t\prime}, \Pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}}, Z_{1:t-1}, Y_{1:t-1})} \\ &= \frac{\Pi_t^{\mathrm{i}}(m^t) \mathbb{P}(Z_t | M_{1:t} = m^t, \gamma_t^{\mathrm{i}})}{\sum_{m^{t\prime}} \Pi_t^{\mathrm{i}}(m^{t\prime}) \mathbb{P}(Z_t | M_{1:t} = m^{t\prime}, \gamma_t^{\mathrm{i}})}, \end{aligned} \tag{5.19}$$

and

$$\begin{aligned} &\Pi_{t+1}^{\mathrm{i}}(m^{t+1}) = \eta_t^{\mathrm{ii}}(\Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Y_t)(m^{t+1}) = \mathbb{P}(M_{1:t+1} = m^{t+1} | \Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Z_{1:t}, Y_{1:t}) \\ &= \frac{\mathbb{P}(M_{1:t} = m^t | \Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Z_{1:t}, Y_{1:t-1}) \mathbb{P}(M_{t+1} = m_{t+1}, Y_t | M_{1:t} = m^t, \Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Z_{1:t}, Y_{1:t-1})}{\sum_{m^{t+1\prime}} \mathbb{P}(M_{1:t} = m^{t\prime} | \Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Z_{1:t}, Y_{1:t-1}) \mathbb{P}(M_{t+1} = m_{t+1}{}', Y_t | M_{1:t} = m^{t\prime}, \Pi_t^{\mathrm{ii}}, \gamma_t^{\mathrm{ii}}, Z_{1:t}, Y_{1:t-1})} \\ &= \frac{\Pi_t^{\mathrm{ii}}(m^t) \mathbb{P}(M_{t+1} = m_{t+1}, Y_t | M_{1:t} = m^t, \gamma_t^{\mathrm{ii}})}{\sum_{m^{t+1\prime}} \Pi_t^{\mathrm{ii}}(m^{t\prime}) \mathbb{P}(M_{t+1} = m_{t+1}{}', Y_t | M_{1:t} = m^{t\prime}, \gamma_t^{\mathrm{ii}})}. \end{aligned} \tag{5.20}$$

#### 5.6.1.3 Structural Results When the Information State Collapses

Notice that once some non-zero $Z$ or $Y$ appears, knowing the current value of $S$ in the CI collapses the beliefs at the time-step (i.e. becomes a deterministic value); it is critical to use only the CI to coordinate the actions of the agents. The following lemma then says that to find an optimal strategy, it is sufficient to ignore all the history of private information prior to the time-step.

**Lemma 5.15:** *At time step $t$, define*

$$\begin{aligned} \bar{t}^{\mathrm{i}}(Z_{1:t-1}, Y_{1:t-1}) &= \max\{t' : Z_{t'} \neq 0 \textit{ for } 1 \le t' < t,\ \textit{ or } Y_{t'} \neq 0 \textit{ for } 1 \le t' < t\},\\ \bar{t}^{\mathrm{ii}}(Z_{1:t}, Y_{1:t-1}) &= \max\{t' : Z_{t'} \neq 0 \textit{ for } 1 \le t' \le t,\ \textit{ or } Y_{t'} \neq 0 \textit{ for } 1 \le t' < t\}. \end{aligned} \tag{5.21}$$

*Suppose the sets over which the maxima are take are not empty, so that $\bar{t}^{\mathrm{i}}$ and $\bar{t}^{\mathrm{ii}}$ are well-defined. Then there exist $\gamma_t^{\mathrm{i}*} \in \Omega(\Omega(S_{\bar{t}^{\mathrm{i}}+1:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1}) \to \Omega(U_t))$ and $\gamma_t^{\mathrm{ii}*} \in \Omega(\Omega(M_{\bar{t}^{\mathrm{ii}}+1:t}) \to \Omega(A_t))$ that achieve the optimality of $V^{\mathrm{i}}_{t-1}(\pi^{\mathrm{i}}_t)$ and $V^{\mathrm{ii}}_t(\pi^{\mathrm{ii}}_t)$, respectively. In other words, it is without loss of generality to restrict our attention to the above two function spaces instead of $\Omega(\Omega(S_{1:t}) \times \Omega(A^1_{1:t-1}) \to \Omega(U_t))$ and $\Omega(\Omega(M_{1:t}) \to \Omega(A_t))$ for the decisions of $\gamma^{\mathrm{i}}_t$ and $\gamma^{\mathrm{ii}}_t$, respectively.*

**Proof:** We only prove the case for the first sub-step. The proof of the second sub-step is the same. Notice that whenever $Z_t \neq 0$ or $Y_t \neq 0$, $S_t$ is in the CI – if $Z_t = U_t S_t \neq 0$, it means $U_t = 1$ and $Z_t = S_t$, and since $Z_t$ is CI, so is $S_t$; on the other hand, if $Y_t = \mathbf{1}\{S_t = A^1_t = A^2_t\} = 1 \neq 0$, then both agents can deduce $S_t$ from their private information, namely past actions $A^1_t$ and $A^2_t$.

Since we consider a Markovian model, given a strategy over time, $S_{\bar{t}^{\mathrm{i}}}$ will be the sufficient statistic of all variables prior to $\bar{t}^{\mathrm{i}}$. Consider an optimal prescription $\gamma^{\mathrm{i}*}_t$ in the original space and two admissible histories $m_1, m_2 \in \Omega(S_{1:\bar{t}^{\mathrm{i}}-1}) \times \Omega(A^1_{1:\bar{t}^{\mathrm{i}}-1})$ and $m_3 \in \Omega(S_{\bar{t}^{\mathrm{i}}:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1})$ so that both $(m_1, m_3)$ and $(m_2, m_3)$ are consistent with the CI. We use proof by contradiction that under $\gamma^{\mathrm{i}*}_t$, the two histories $(m_1, m_3)$ and $(m_2, m_3)$ must have the same value. One can then construct an optimal strategy that disregards the first part of the tuple.

For more details see Appendix D.5.1. ■

**Remark 5.16:** *At time step $t$, $\bar{t}^{\mathrm{i}}$ is at most $t-1$; since the agents have observed $Y_{\bar{t}^{\mathrm{i}}}$, they can deduce each others' actions as $S_{\bar{t}^{\mathrm{i}}}$ has become CI, so that $A_{\bar{t}^{\mathrm{i}}}$ also becomes CI. This is also the case for the second sub-step when $\bar{t}^{\mathrm{ii}} \leq t-1$. When $\bar{t}^{\mathrm{ii}} = t$, U1 just sent $S_t$ to U2, and $A_t$ is yet to be decided.*

After Lemma 5.15 is established, whenever some non-zero $Z$ or $Y$ appears, we will restrict the spaces of the prescriptions according to Lemma 5.15. Then the information states can be further split. Define two new information states for the two sub-steps as

$$\begin{aligned}\Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}(m) &= \mathbb{P}^{g_{\bar{t}^{\mathrm{i}}:t-1}}(M_{\bar{t}^{\mathrm{i}}+1:t} = m | S_{\bar{t}^{\mathrm{i}}}, Z_{\bar{t}^{\mathrm{i}}:t-1}, Y_{\bar{t}^{\mathrm{i}}:t-1}),\\ \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m) &= \mathbb{P}^{g_{\bar{t}^{\mathrm{ii}}:t-1}, g^{\mathrm{i}}_t}(M_{\bar{t}^{\mathrm{ii}}+1:t} = m | S_{\bar{t}^{\mathrm{ii}}}, Z_{\bar{t}^{\mathrm{ii}}:t}, Y_{\bar{t}^{\mathrm{ii}}:t-1}).\end{aligned} \tag{5.22}$$

**Lemma 5.17:** *Consider time step $t$. Suppose we use the prescription space reduction of Lemma 5.15 throughout and that $\bar{t}^{\mathrm{i}}$ is well-defined. Then given the CI $(Z_{1:t-1}, Y_{1:t-1})$, $\Pi^{\mathrm{i}}_t$ can be split as*

$$\Pi^{\mathrm{i}}_t(m, m') = \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}}(m) \cdot \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}(m'), \tag{5.23}$$

*where $m \in \Omega(M_{1:\bar{t}^{\mathrm{i}}})$ and $m' \in \Omega(M_{\bar{t}^{\mathrm{i}}+1:t})$. Similarly, given the CI $(Z_{1:t}, Y_{1:t-1})$ and the assumption that $\bar{t}^{\mathrm{ii}}$ is well-defined, $\Pi^{\mathrm{ii}}_t$ can be split as*

$$\Pi^{\mathrm{ii}}_t(m, m') = \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}}(m) \cdot \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m'), \tag{5.24}$$

*where $m \in \Omega(M_{1:\bar{t}^{\mathrm{ii}}})$ and $m' \in \Omega(M_{\bar{t}^{\mathrm{ii}}+1:t})$.*

**Proof:** The proof follows from the definition of conditional probability, the fact that $S_{\bar{t}^{\mathrm{i}}}$ is the sufficient statistics for the previous variables given the strategies, and Lemma 5.15 so that the strategies do not depend on the first part of the history tuple. See Appendix D.5.2. ■

An intermediate proposition given in Appendix D.5.3, Proposition D.6, use the strategy space reduction in Lemma 5.15 and the split of information state in Lemma 5.17 to rewrite the DP equations and the information state updating rules in (5.19) and (5.20). The following proposition then follows from the stationarity of the model and the fact that the dynamics in Proposition D.6 do not depend on the values of $\bar{t}^{\mathrm{i}}$ and $\bar{t}^{\mathrm{ii}}$ but only on the differences between $t$ and them.

**Proposition 5.18:** *Consider time step $t$. Given the CI $(Z_{1:t-1}, Y_{1:t-1})$, define $\tau^{\mathrm{i}} = t - \bar{t}^{\mathrm{i}}$. The DP equation for the first sub-step can be written as*

$$V_t^{\mathrm{i}}(\pi^{\mathrm{i}}_{\tau^{\mathrm{i}}}) = \sup_{\gamma^{\mathrm{i}}_{\tau^{\mathrm{i}}}} \mathbb{E}\left[ -c\gamma^{\mathrm{i}}_{\tau^{\mathrm{i}}}(M_{\bar{t}^{\mathrm{i}}+1:t}) + V_t^{\mathrm{ii}}(\eta^{\mathrm{i}}_{\tau^{\mathrm{i}}}(\Pi^{\mathrm{i}}_{\tau^{\mathrm{i}}}, \gamma^{\mathrm{i}}_{\tau^{\mathrm{i}}}, Z_t)) \middle| \, \Pi^{\mathrm{i}}_{\tau^{\mathrm{i}}} = \pi^{\mathrm{i}}_{\tau^{\mathrm{i}}} \right], \tag{5.25}$$

*where $\pi^{\mathrm{i}}_{\tau^{\mathrm{i}}}$ and $\gamma^{\mathrm{i}}_{\tau^{\mathrm{i}}}$ now only depend on the gap between $\bar{t}^{\mathrm{i}}$ and $t$. Similarly, given the CI $(Z_{1:t}, Y_{1:t-1})$, define $\tau^{\mathrm{ii}} = t - \bar{t}^{\mathrm{ii}}$. The DP equation for the second sub-step can be written as*

$$V_t^{\mathrm{ii}}(\pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}) = \sup_{\gamma^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}} \mathbb{E}\left[ Y_t(S_t, \gamma^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}(M_{\bar{t}^{\mathrm{ii}}+1:t})) + \lambda V_{t+1}^{\mathrm{i}}(\eta^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}(\Pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}, \gamma^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}, Y_t)) \middle| \, \Pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}} = \pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}} \right], \tag{5.26}$$

*where $\pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}$ and $\gamma^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}$ now only depend on the gap between $\bar{t}^{\mathrm{ii}}$ and $t$. The terms $\eta^{\mathrm{i}}_{\tau^{\mathrm{i}}}$ and $\eta^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}$ are the update rules*[5]*.*

**Proof:** This follows from the stationarity of the model and the fact that the dynamics in Proposition D.6 do not depend on the values of $\bar{t}^{\mathrm{i}}$ and $\bar{t}^{\mathrm{ii}}$ but only on the differences between $t$ and them. See Appendix D.5.4. ■

#### 5.6.1.4 Characterization of the Optimal Policy

Note that Proposition 5.18 follows directly by the fact that $S_{\bar{t}}$ is deterministic in CI and the time-invariance of the model. In this Subsection, we use the details of the model to derive the optimal policy. Note that since the DP equations are time-invariant, equations (5.25) and (5.26) do not directly depend on $t$; in particular, $V_t$ is actually just $V$, and $(Z_t, Y_t)$ as well as $M_{\bar{t}+1:t}$ can be viewed as "the current $(Z, Y)$" and "the most recent $\tau$ $M$'s." Moreover, now that we have Proposition 5.18, we can review (5.22) again. First, from the definition of $\bar{t}^{\mathrm{i}}$ and $\bar{t}^{\mathrm{ii}}$, the CIs $Z_{\bar{t}^{\mathrm{i}}+1:t-1}, Y_{\bar{t}^{\mathrm{i}}+1:t-1}$ and $Z_{\bar{t}^{\mathrm{ii}}+1:t}, Y_{\bar{t}^{\mathrm{ii}}+1:t-1}$ are all 0's, so that it is unnecessary to put them in the condition. Second, from the time-invariance of Proposition 5.18, the value of $t$ does not matter.

---

[5] The update rules $\eta^{\mathrm{i}}_{\tau^{\mathrm{i}}}$ and $\eta^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}$ are omitted for brevity. They follow from (D.16) and (D.18), except that now they do not depend on the values of $\bar{t}^{\mathrm{i}}$ and $\bar{t}^{\mathrm{ii}}$, but only on the differences between $\bar{t}^{\mathrm{i}}$ and $t$ and between $\bar{t}^{\mathrm{ii}}$ and $t$, and hence, we index the differences by $\tau^{\mathrm{i}}$ and $\tau^{\mathrm{ii}}$, respectively.

We can adopt another index system, where 0 is the current time step, and $-t$ is the $t$-th most recent time step. Using this indexing and Proposition 5.18, (5.22) can be rewritten as

$$\begin{aligned} \Pi^{\mathrm{i}}_{\tau^{\mathrm{i}}}(m) &= \mathbb{P}^{g_{-\tau^{\mathrm{i}}:-1}}(M_{-\tau^{\mathrm{i}}+1:0} = m | S_{-\tau^{\mathrm{i}}}, Z_{-\tau^{\mathrm{i}}}, Y_{-\tau^{\mathrm{i}}}), \\ \Pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}(m) &= \mathbb{P}^{g_{-\tau^{\mathrm{ii}}:-1}, g^{\mathrm{i}}_0}(M_{-\tau^{\mathrm{ii}}+1:0} = m | S_{-\tau^{\mathrm{ii}}}, Z_{-\tau^{\mathrm{ii}}}, Y_{-\tau^{\mathrm{ii}}}). \end{aligned} \tag{5.27}$$

If $\tau^{\mathrm{ii}} = 0$, then $Y_0$ is yet to be observed and should not be present in $\Pi^{\mathrm{ii}}_0(m)$. The distribution $\Pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}$ is a *random* distribution whose realization only depends on the random variables $(S_{-\tau^{\mathrm{ii}}}, Z_{-\tau^{\mathrm{ii}}}, Y_{-\tau^{\mathrm{ii}}})$. We denote $\pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}},s,v}$ as a realization of $\Pi^{\mathrm{ii}}_{\tau^{\mathrm{ii}}}$, where $s \in \{+1, -1\}$ is the realization of $S_{-\tau^{\mathrm{ii}}}$, and $v = y$ if $Y_{-\tau^{\mathrm{ii}}} = 1$ and $v = z$ otherwise (in this case it must be that $Z_{-\tau^{\mathrm{ii}}} \neq 0$). As stated previously, when $\tau^{\mathrm{ii}} = 0$, it must be that $v = z$.

**Theorem 5.19:** *All optimal strategies,* $g^*_{1:\infty}$*, always assign* $A^1_t = g^{\mathrm{ii},1}_t(M^1_{1:t}, \Pi^{\mathrm{ii}}_t) = S_t$ *for all time steps* $t$ *given the reward structure of* $R^{\mathrm{ii}}_t = Y_t(S_t, A_t) = \mathbf{1}\{S_t = A^1_t = A^2_t\}$.

**Proof:** A complete policy is of the form $g_{1:\infty} = (g^{\mathrm{i}}_{1:\infty}, g^{\mathrm{ii},1}_{1:\infty}, g^{\mathrm{ii},2}_{1:\infty})$. Denote the policy that U1 always matches the state for all $t$ as $g^{\mathrm{ii},1\sharp}_{1:\infty}$. Under this policy, we have Corollary 5.20, that is, the previous state is always CI; moreover, for the remaining parts of the complete policy, we can find an optimal one, denoted as $(g^{\mathrm{i}\sharp}_{1:\infty}, g^{\mathrm{ii},2\sharp}_{1:\infty})$. We show that U1 will not deviate from $g^{\sharp}_{1:\infty} = (g^{\mathrm{i}\sharp}_{1:\infty}, g^{\mathrm{ii},1\sharp}_{1:\infty}, g^{\mathrm{ii},2\sharp}_{1:\infty})$ by not matching the state. Hence, by the Policy Improvement Theorem [115], $g^{\sharp}_{1:\infty}$ is an optimal policy. See Appendix D.6.1. ■

Since in our model the state is binary, Theorem 5.19 has important implications discussed below.

**Corollary 5.20:** *In the case where there are only two elements in the state space, under an optimal strategy over time, for any time step* $t$*,* $S_{t-1}$ *is CI, so that in the first sub-step we always have the information state being* $\Pi^{\mathrm{i}}_1$*, and in the second sub-step we always have the information state being either* $\Pi^{\mathrm{ii}}_1$ *when U1 did not communicate or* $\Pi^{\mathrm{ii}}_0$ *when it did.*

**Proof:** U2 can deduce $S_{t-1}$ by $S_{t-1} = A^2_{t-1} \cdot (-1)^{Y_{t-1}+1}$. Proposition 5.18 follows with $\vec{t}^{\mathrm{i}} = t-1$, and $\vec{t}^{\mathrm{ii}} = t-1$ if $U_t = 0$ and $\vec{t}^{\mathrm{ii}} = t$ if $U_t = 1$. See Appendix D.6.2. ■

**Remark 5.21:** *Using Lemma 5.15, Corollary 5.20 implies the optimal strategy (or strategies) is* stationary.

**Theorem 5.22:** *There is an optimal strategy over time such that for all time step t, U1 follows the communication protocol (breaking ties arbitrarily if equality holds in all the inequalities)*

$$U_t = g^{\mathrm{i}}(S_{t-1}, Y_{t-1}, S_t) = \begin{cases} 1, & \textit{if } U_t' = 1, S_t = S_{t-1}\cdot(-1)^{Y_{t-1}+1}, \textit{ and } \epsilon < 0.5, \\ 1, & \textit{if } U_t' = 1, S_t = S_{t-1}\cdot(-1)^{Y_{t-1}}, \textit{ and } \epsilon > 0.5, \\ 0, & \textit{if } U_t' = 1, S_t = S_{t-1}\cdot(-1)^{Y_{t-1}}, \textit{ and } \epsilon < 0.5, \\ 0, & \textit{if } U_t' = 1, S_t = S_{t-1}\cdot(-1)^{Y_{t-1}+1}, \textit{ and } \epsilon > 0.5, \\ 0, & \textit{if } U_t' = 0, \end{cases} \tag{5.28}$$

*where $\bar{\epsilon} = 1 - \epsilon$ and $\bar{\beta} = 1 - \beta$, and $U_t'$ is determined by the model parameters (breaking ties arbitrarily)*

$$U_t' = \begin{cases} 1, & \textit{if } \epsilon < \beta,\ \epsilon+\beta > 1,\ \textit{and } \bar{\beta} > \min\{\epsilon, \bar{\epsilon}\}c, \\ 0, & \textit{if } \epsilon < \beta,\ \epsilon+\beta > 1,\ \textit{and } \bar{\beta} < \min\{\epsilon, \bar{\epsilon}\}c, \\ 1, & \textit{if } \epsilon < \beta,\ \epsilon+\beta < 1,\ \textit{and } \epsilon > \min\{\epsilon, \bar{\epsilon}\}c, \\ 0, & \textit{if } \epsilon < \beta,\ \epsilon+\beta < 1,\ \textit{and } \epsilon < \min\{\epsilon, \bar{\epsilon}\}c, \\ 1, & \textit{if } \epsilon > \beta,\ \epsilon+\beta > 1,\ \textit{and } \bar{\epsilon} > \min\{\epsilon, \bar{\epsilon}\}c, \\ 0, & \textit{if } \epsilon > \beta,\ \epsilon+\beta > 1,\ \textit{and } \bar{\epsilon} < \min\{\epsilon, \bar{\epsilon}\}c, \\ 1, & \textit{if } \epsilon > \beta,\ \epsilon+\beta < 1,\ \textit{and } \beta > \min\{\epsilon, \bar{\epsilon}\}c, \\ 0, & \textit{if } \epsilon > \beta,\ \epsilon+\beta < 1,\ \textit{and } \beta < \min\{\epsilon, \bar{\epsilon}\}c. \end{cases} \tag{5.29}$$

*In addition, if $U_t' = 1$, U2 decodes the received communication result $U_t$ by (breaking* *ties arbitrarily but should follow U1's choice)*

$$A_t^2 = g^{\mathrm{ii},2}(S_{t-1}, Y_{t-1}, O_t, U_t) = \begin{cases} S_{t-1} \cdot (-1)^{Y_{t-1}+1}, & \textit{if } U_t = 1 \textit{ and } \epsilon < 0.5, \\ S_{t-1} \cdot (-1)^{Y_{t-1}}, & \textit{if } U_t = 1 \textit{ and } \epsilon > 0.5, \\ S_{t-1} \cdot (-1)^{Y_{t-1}}, & \textit{if } U_t = 0 \textit{ and } \epsilon < 0.5, \\ S_{t-1} \cdot (-1)^{Y_{t-1}+1}, & \textit{if } U_t = 0 \textit{ and } \epsilon > 0.5; \end{cases} \tag{5.30}$$

*otherwise, if $U'_t = 0$, then U2 chooses the best posterior guess according to (breaking ties arbitrarily)*

$$A_t^2 = g^{\mathrm{ii},2}(S_{t-1}, Y_{t-1}, O_t, U_t) = \begin{cases} O_t, & \textit{if } O_t = S_{t-1} \cdot (-1)^{Y_{t-1}+1} \textit{ and } \epsilon < \beta, \\ -O_t, & \textit{if } O_t = S_{t-1} \cdot (-1)^{Y_{t-1}+1} \textit{ and } \epsilon > \beta, \\ O_t, & \textit{if } O_t = S_{t-1} \cdot (-1)^{Y_{t-1}} \textit{ and } \epsilon + \beta > 1, \\ -O_t, & \textit{if } O_t = S_{t-1} \cdot (-1)^{Y_{t-1}} \textit{ and } \epsilon + \beta < 1. \end{cases} \tag{5.31}$$

**Proof:** Now that we have Corollary 5.20 and Remark 5.21 as well as the symmetries of transition and observation, all we have to do is to find a policy of a time step that maximizes the expected instantaneous reward. Performing the policy in every time step will maximize the expected long term reward.

Depending on the cost of communication $c$, U1 decides whether to communicate or not. If it does ($U'_t = 1$), it uses the 1-bit information of $U_t$ to convey whether $S_t = \bar{S}_t = S_{t-1} \cdot (-1)^{Y_{t-1}+1}$ or $S_t = -\bar{S}_t = S_{t-1} \cdot (-1)^{Y_{t-1}}$. Naturally, it chooses the convention (a bijection from 0/1 to $\bar{S}_t/-\bar{S}_t$) so that 1 is sent less frequently, since sending 1 is costly. In particular, U1 is choosing between three policies: $g^{\mathrm{i}}[0,0]$, $g^{\mathrm{i}}[0,1]$, and $g^{\mathrm{i}}[1,0]$, where $g^{\mathrm{i}}[j,k]$ is defined to be the policy that U1 plays $U_t = j$ deterministically when $S_t = \bar{S}_t$ and $U_t = k$ deterministically when $S_t = -\bar{S}_t$, where $j, k \in \{0,1\}$. The decision $U'_t = 1$ corresponds to the case where U1 chooses one of $g^{\mathrm{i}}[0,1]$ and $g^{\mathrm{i}}[1,0]$, and the decision $U'_t = 0$ corresponds to the case where U1 chooses $g^{\mathrm{i}}[0,0]$.

When U1 decides not to communicate at all ($U'_t = 0$), the expected instantaneous reward is $\mathbb{P}(Y_t = 1)$. There will be two cases $O_t = S_{t-1} \cdot (-1)^{Y_{t-1}+1}$ or $O_t = S_{t-1} \cdot$

$(-1)^{Y_{t-1}}$; U2 maximizes $\mathbb{P}(Y_t = 1)$ by calculating the posterior probability of $\mathbb{P}(Y_t = 1|A_t^2 = O_t)$ and $\mathbb{P}(Y_t = 1|A_t^2 = -O_t)$ to decide whether to *follow or flip* its observation for each case. See Appendix D.6.3. ■

### 5.6.2 Multi-Agent Reinforcement Learning Solution with Hierarchical Bandit

In this section we propose a reinforcement learning algorithm that achieves optimality asymptotically for the model of the matching problem when the parameters are unknown (however, $c$ is known since it can be directly observed). An initial thought would be for the agents to collect the histories and to *estimate* the parameters, while in each time step they just act based on Theorem 5.19 and Theorem 5.22 with the estimated parameters. However, the matching problem is one of the examples such that this "kernel-based method" will not work. In particular, $O_t$ is the private information of U2, and although U1 can deduce $A_t^2$ in the next time step by $Y_t$, it has no means to tell $O_t$ since it does not know whether U2 was following or flipping the observation. Without the knowledge of $O_{1:\infty}$, U1 cannot estimate $\beta$ and enact the optimal communication decision in Theorem 5.22.

Notice that Theorem 5.19 does not depend on any parameter; hence, even if the parameters are unknown, U1 still always follows the state, so that $S_{t-1}$ is still CI as in Corollary 5.20. Given $S_{t-1}$, the statistics of $(S_t, O_t)$ the agents are facing is time-invariant. Since they are encountering a set of choices with fixed but unknown statistics, this belongs to the class of multi-agent stochastic MAB problems. In particular, beside U1's following the state, in every time-step, U1 first faces the decision of communicating or not and if so the decision of the communication protocol as in Theorem 5.22, then conditioning on U1 deciding *not to* communicate, U2 faces the decision of following or flipping its observation as in Theorem 5.22. U2 is actually facing two sets of decisions, depending on whether $O_t = \bar{S}_t$ or $O_t = -\bar{S}_t$, where $\bar{S}_t = S_{t-1} \cdot (-1)^{Y_{t-1}+1}$. In addition, since U1 does not have the information of $O_t$, it does not know which set of decisions U2 faced and U2's decision of follow/flip at the end of the time step.

In the matching problem, we can argue that we have two 2HB problems described in Protocol 5.1, one for $O_t = \bar{S}_t$ and one for $O_t = -\bar{S}_t$, selected by nature. Within each problem, U1 decides $g^{\mathrm{i}} \in \{g^{\mathrm{i}}[0,0], g^{\mathrm{i}}[0,1], g^{\mathrm{i}}[1,0]\}$ followed by U2 deciding $A_t^2 \in \{O_t, -O_t\}$. Therefore, $K = 3$ and $L = 2$ in our matching problem[6]. In the case of $O_t = \bar{S}_t$, there are four possible combinations of $(S_{t-1}, Y_{t-1}, O_t)$; however, Theorem 5.22 implies that given the model parameters, the optimal policy will be either follow or flip the observations for all four combinations. Same in the case of $O_t = -\bar{S}_t$. These hold due to the symmetries of the state transition model and the observation model[7].

### 5.6.3 More Generalizations

#### 5.6.3.1 Generalization to the Time-Varying Case

Two key properties are necessary to ensure Algorithm 5.3's convergence to optimality. First, the underlying statistics must be time-invariant to allow a stochastic MAB solution. Second, U2 must receive U1's choice of $k_t$ (or exactly reproduce U1's computation); otherwise, one can still use UCB-like methods on the problem, but convergence to optimality is not guaranteed. Consider a generalization of the matching problem, where U1 observes imperfectly as well with $\mathbb{P}(O_t^1 = S_t) = \alpha$ and $\mathbb{P}(O_t^1 = -S_t) = 1 - \alpha$. Then when U1 does not receive the reward, i.e. $Y_t = 0$, it does not know whether the mistake was made by itself, or U2, or both of them; same for U2. In this case they cannot deduce $S_t$. For ease of presentation, we assume the environment will announce the current state at the end of a time step if they have failed to match two times in a row, so that at any time step $t$ either $S_{t-1}$ or $S_{t-2}$ is CI, and we enforce $U_t = 0$ (by assigning $c > \frac{1}{1-\lambda}$ for example), i.e., no communication is allowed. Let $\bar{\alpha} = 1 - \alpha$, $\bar{\beta} = 1 - \beta$, and $\bar{\epsilon} = 1 - \epsilon$.

Now consider when the model is known at time step $t$. Suppose $S_{t-1}$ is CI, so that $\mathbb{P}(S_t = \bar{S}_t) = \bar{\epsilon}$; the information state is $\Pi_0$ (we can omit the state subscript

---

[6] When $g^{\mathrm{i}} \in \{g^{\mathrm{i}}[0,1], g^{\mathrm{i}}[1,0]\}$, we can equivalently model the situation as the agents receiving $1 - \min\{\epsilon, \bar{\epsilon}\}c$ regardless of the value of $A_t^2$. Note that in our original description U2 has no other option but to align $A_t^2$ with the state indicated by U1's communication signal $U_t$.

[7] It might be confusing why in the matching problem U2 receives U1's choice at first. In fact, it does not really receive the choice – if U1 chooses $g^{\mathrm{i}}[0,1]$ but transmits a 0 following the communication protocol, then U2 only observes 0; even if U2 observes 1, it could not differentiate between $g^{\mathrm{i}}[0,1]$ and $g^{\mathrm{i}}[1,0]$. However, the fact that U2 receives $k_t$ is not crucial, since U1's perspective is the same as that of the coordinator. Every computation needed in the following Algorithm 5.3 for U1 to obtain $k_t$ can be done at U2 as well, so U2 knows the $g^{\mathrm{i}}$ U1 will be choosing in advance, but not the state $S_t$ which only U1 can see.

due to symmetry). Also denote the optimal policy in (5.31) as $g^{\mathrm{ii}*}_{\epsilon,\beta}(\bar{S}_t, O_t)$ (note that (5.31) does not use the information of $(S_{t-1}, Y_{t-1}$ apart from $\bar{S}_t)$. From the argument of Appendix D.6.3, they always want to maximize $\mathbb{P}(Y_t = 1)$, so that the optimal policies in this time step should be $g^{\mathrm{ii}*}_{\epsilon,\alpha}(\bar{S}_t, O^1_t)$ and $g^{\mathrm{ii}*}_{\epsilon,\beta}(\bar{S}_t, O^2_t)$ for U1 and U2, respectively, as $\mathbb{P}(Y_t = 1)$ will be maximized if both $\mathbb{P}(S_t = A^1_t)$ and $\mathbb{P}(S_t = A^2_t)$ are maximized. Depending on the regime $(\alpha, \beta, \epsilon)$ lies in, there are 16 possibilities; we assume the case of $\epsilon < \alpha, \beta$ and $\bar{\epsilon} < \alpha, \beta$ so that both agents always follow their observations, for ease of presentation. Now if they fail to match, the information state will grow to $\Pi_1$ at $t + 1$, where U1 (and U2 symmetrically) can calculate the prior of the state using Bayes rule according to its private observation:

$$\begin{aligned} \mathbb{P}(S_{t+1} = \bar{S}_t | O^1_t = \bar{S}_t, Y_t = 0) &= \frac{\bar{\epsilon}\alpha\bar{\beta}\epsilon + \epsilon\bar{\alpha}\bar{\epsilon}}{\bar{\epsilon}\alpha\bar{\beta} + \epsilon\bar{\alpha}} := 1 - \epsilon'_1(O^1_t = \bar{S}_t), \\ \mathbb{P}(S_{t+1} = \bar{S}_t | O^1_t = -\bar{S}_t, Y_t = 0) &= \frac{\epsilon\alpha\bar{\beta}\bar{\epsilon} + \bar{\epsilon}\bar{\alpha}\epsilon}{\epsilon\alpha\bar{\beta} + \bar{\epsilon}\bar{\alpha}} := 1 - \epsilon'_1(O^1_t = -\bar{S}_t). \end{aligned} \tag{5.32}$$

Define $\epsilon'_2(O^2_t)$ similarly (by switching $\alpha$ and $\beta$ in (5.32)). Then in $\Pi_1$ (at $t + 1$) their optimal policies will become $g^{\mathrm{ii}*}_{\epsilon'_1(O^1_t),\alpha}(\bar{S}_t, O^1_{t+1})$ and $g^{\mathrm{ii}*}_{\epsilon'_2(O^2_t),\beta}(\bar{S}_t, O^2_{t+1})$.

U1's prescription functions are as follows. In $\Pi_0$, U1's optimal prescription, $g^{\mathrm{ii}*}_{\epsilon,\alpha}(\bar{S}_t, O^1_t)$, specify whether it should follow or flip the observation for both cases of $O^1_t = \bar{S}_t$ and $O^1_t = -\bar{S}_t$. To shorten the notations, set $\bar{O} = \mathbf{1}\{O = \bar{S}\}$ and $\bar{A} = \mathbf{1}\{A = O\}$ to manifest the effective dimension of interest. Then U1's optimal prescription in state $\Pi_0$ should be a function $\gamma^{\mathrm{i}}_0 \in \Gamma^{\mathrm{i}}_0 := \Omega(\Omega(\bar{O}^1_t) \to \Omega(\bar{A}^1_t))$. Similarly, U1's optimal prescription in state $\Pi_1$, $g^{\mathrm{ii}*}_{\epsilon'_1(O^1_t),\alpha}(\bar{S}_t, O^1_{t+1})$, should be a function $\gamma^{\mathrm{i}}_1 \in \Gamma^{\mathrm{i}}_1 := \Omega(\Omega(\bar{O}^1_t) \times \Omega(\bar{O}^1_{t+1}) \to \Omega(\bar{A}^1_{t+1}))$. Note that the two $\Gamma$'s do not have subscript $t$ due to Proposition 5.18.

Consider this generalization when the model (parameters) is unknown. A naive application of bandit methods is to treat U1's policy in $\Pi_0$ as two MAB problems, each has two arms (just as U2's situation in the second layer of Algorithm 5.3), and U1's policy in $\Pi_0$ as four MAB problems, each has two arms; same for U2. Both key properties for Algorithm 5.3 to work are violated. There are two states $\Pi_0$ and $\Pi_1$, hence the environment is time-varying; one cannot treat the problem as six MABs with two arms either, since the choice of $\gamma^{\mathrm{i}}_0$ affects the probability of entering $\Pi_1$, and maximizing the reward in $\Pi_0$ may not aligned with maximizing $\mathbb{P}(\text{enter } \Pi_1)V(\Pi_1)$.

In addition, when they fail to match, they do not receive each other's actions, so that the information structure in Protocol 5.1 does not hold. However, the CI approach illuminates a path using Q-learning so that convergence to optimality is guaranteed. In particular, as the DP equation in Proposition 5.18 suggests, we treat $\{\Pi_0, \Pi_1\}$ as the state space, $\Gamma_0^{\mathrm{i}} \times \Gamma_0^{\mathrm{ii}}$ as the action space of $\Pi_0$, and $\Gamma_1^{\mathrm{i}} \times \Gamma_1^{\mathrm{ii}}$ as the action space of $\Pi_1$, then the problem can be solved with Q-learning when the model is unknown. Since the learning is done by the coordinator using only CI, the non-stationarity issue in MARL problem is alleviated. Essentially, this scheme establishes a common ground between the two agents, so that any learning done in one agent can be computationally reproduced by the other.

#### 5.6.3.2 Generalization to the Asymmetric Case

When the transition and observation probabilities do not have the symmetric structure as described in the start of Section 5.6, U1 does not necessarily match the state in the optimal policy. The following is one such example. For the transition probabilities, we have

- $\mathbb{P}(S_t = +1 | S_{t-1} = +1, Y_{t-1} = 1) = \mathbb{P}(S_t = +1 | S_{t-1} = -1, Y_{t-1} = 0) = 0.99$,
- $\mathbb{P}(S_t = -1 | S_{t-1} = +1, Y_{t-1} = 1) = \mathbb{P}(S_t = -1 | S_{t-1} = -1, Y_{t-1} = 0) = 0.01$,
- $\mathbb{P}(S_t = +1 | S_{t-1} = -1, Y_{t-1} = 1) = \mathbb{P}(S_t = +1 | S_{t-1} = +1, Y_{t-1} = 0) = 0.49$, and
- $\mathbb{P}(S_t = -1 | S_{t-1} = -1, Y_{t-1} = 1) = \mathbb{P}(S_t = -1 | S_{t-1} = +1, Y_{t-1} = 0) = 0.51$.

For the observation probabilities, we have $\mathbb{P}(O_t = +1 | S_t = +1) = \mathbb{P}(O_t = +1 | S_t = -1) = \beta$ and $\mathbb{P}(O_t = -1 | S_t = +1) = \mathbb{P}(O_t = -1 | S_t = -1) = 1 - \beta$, where $\beta \in (0, 1)$; the observation is basically useless for U2, and U2 will guess the state solely from the prior. We again forbid communication by assigning $c > \frac{1}{1-\lambda}$, where $\lambda = 0.9$.

Now suppose U1 always matches the state. Then U2 disregards observations, and always guesses $S_t = +1$ if $(Y_{t-1}, S_{t-1})$ is equal to $(1, +1)$ or $(0, -1)$, and always guesses $S_t = -1$ if $(Y_{t-1}, S_{t-1})$ is equal to $(1, -1)$ or $(0, +1)$. This means the two sets of information states (where the tuples are in the form of $(Y_{t-1}, S_{t-1})$) $\{(1, +1), (0, -1)\}$ and $\{(1, -1), (0, +1)\}$ are not commutative under the policy. For U1, at each time step there are eight possible inputs of $(Y_{t-1}, S_{t-1}, S_t)$. For the four inputs that have

$(Y_{t-1}, S_{t-1})$ equal to $(1, +1)$ or $(0, -1)$, the values are close to $1/(1 - 0.9) = 10$, while the four inputs that have $(Y_{t-1}, S_{t-1})$ equal to $(1, -1)$ or $(0, +1)$ have values close to $0.5/(1-0.9) = 5$. Then when $(Y_{t-1}, S_{t-1}, S_t) = (1, -1, -1)$, instead of getting 5 (more precisely 5.59), U1 could gain by not matching the state so that in the next time step they have $(Y_{t-1}, S_{t-1}) = (0, -1)$, obtaining roughly $\lambda \cdot 10 = 9$ (more precisely 8.91). This implies always matching is not optimal, which destroys the good properties of Corollary 5.20 and Remark 5.21. Characterizing the optimal policy in this case with the corresponding reinforcement learning method are left as future work.

#### 5.6.3.3 Generalization to More Than Two States

From Section 5.6.3.2, it is easy to imagine that with more than two states and no particular structure enforced on the probabilities, one could also find examples such that U1 always matching is not optimal. Here is a concise one with three states. Suppose $\mathcal{S} = \{I, J, K\}$ and $\mathcal{O} = \{i, j, k\}$. If $Y_{t-1} = 1$, then $\mathbb{P}(S_t = I|S_{t-1} = I) = \mathbb{P}(S_t = J|S_{t-1} = I) = \mathbb{P}(S_t = I|S_{t-1} = J) = \mathbb{P}(S_t = J|S_{t-1} = J) = 0.5$, $\mathbb{P}(S_t = I|S_{t-1} = K) = \mathbb{P}(S_t = J|S_{t-1} = K) = 0.1$, and $\mathbb{P}(S_t = K|S_{t-1} = K) = 0.8$; if $Y_{t-1} = 0$, then all states transit to $S_t = K$. For the observation probabilities, we have $\mathbb{P}(O_t = i|S_t = I) = \mathbb{P}(O_t = j|S_t = I) = \mathbb{P}(O_t = i|S_t = J) = \mathbb{P}(O_t = j|S_t = J) = 0.5$ and $\mathbb{P}(O_t = k|S_t = K) = 1$. Let $\lambda = 0.9$. The two states $I$ and $J$ are the "bad states" where U2 has unclear observation and does no better than randomly guessing between them, while in state $K$ U2 observes perfectly. In $I$ or $J$, instead of getting a 0.5 instantaneous reward, U1 would rather not matching the state and bringing them back to state $K$, so that they will earn 0.9 in the next step even with discounting.

Again, without Corollary 5.20 and Remark 5.21 it is unclear how to characterize the optimal policy when the model is known, let alone when it is unknown. Our Q-learning method works in Section 5.6.3.1 because we assume that the system will announce the state if they fail to match two times. As we learn from Proposition 5.18, this limits the length of AOH and CI required for optimal decision to two, so that the state space and the action space in the Q-learning method are finite. For general MARL problem, the length could go to infinity, and one needs to approximate the information state by discarding part of the history to construct an algorithm with reasonable complexity. This will also be left as future work.

## 5.7 Conclusion and Future Directions

In this chapter, we exploited the hierarchical information structure in hierarchical bandits, and proposed efficient and near-optimal algorithms that require no coordination or communication between the agents. A key feature of our algorithms is that the leader, i.e., the less-informed upper level agent, performs single-agent-like near-optimal subroutines with specially designed bonuses without the need of knowing or tracking the learning procedure of the lower level agent(s). We extended the analysis to the settings of multiple agents in each layer, multiple layers, and MDP. In detail we analyzed the structure of the optimal policies in the matching problem, and apply the two-stage hierarchical bandit algorithm to the problem in the RL setting. This demonstrates that if we know a certain structure about the problem model, we can first analyzed the problem to obtain structural results using control theory, then approach the problem in the RL context using much simpler algorithms based on the structural results.

One future direction is to explore other information structures that may allow simplification when more than two agents and reward information asymmetry come in. In particular, special cases of Dec-POMDP that come up frequently in applications, such as the *rich observation environment* [8], are worth to explore. Extending the results to bandit problems with structural relationships between the payoffs of the arms and to general-sum games are also interesting directions.

## CHAPTER VI

# Conclusion

In this chapter, we discuss holistic takeaways from various aspects of multi-agent systems (MASs) in Section 6.1 and general future directions for previous chapters in Section 6.2.

## 6.1 High-level Takeways

### 6.1.1 Local Memory

Agents maintain local memories to facilitate decentralization in MASs. In the distributed optimization setting studied in Chapter II and Chapter III, agents maintain copies of the decision variable and in some algorithms copies of the gradient mean. In the multi-agent reinforcement learning (MARL) setting studied in Chapter IV and Chapter V, agents maintain their private action-observation histories (AOHs) (mainly Chapter IV), and their updated local policies as well as local copies of the $Q/V$ functions (mainly Chapter V).

In this thesis, we achieved *memory efficiency* by exploiting the problem structures. In Chapter II, we used the *partial dependency structure* that often arises in application problems to reduce the local memory usage, so that the agents only have to store the variable parts their objective functions depend on. In Chapter IV, agents can keep only the compressed public and private states with the encapsulations satisfying our proposed conditions during execution without losing optimality, in contrast to keeping the entire histories.

The amount of local memory kept usually reflects a balancing of factors with other system performance factors, including the "value" (objective value or cumulative reward) attained, the amount of communication, the convergence rate, etc. In particular, in Section 2.5 we saw that there is a tradeoff between the convergence rate and the amount of local memory and communication required – we can judiciously add high-degree nodes that originally do not depend on a variable part into the associated local network to facilitate its gossip. In Chapter IV, the class of approximate encapsulations (Section 4.4) includes the class of exact encapsulations (Section 4.3). Hence, it is easy to see that with an optimality gap allowed, the approximate representations can meet the required range of value function with more lossy compressions and less local memory requirement. The same feature was observed in distributed averaging and federated learning as well [90, 105]: the quantization level is related to the size of the solution neighborhood one can converge to; therefore, the finer the quantization is, the closer one can get to the optimal solution, but the larger the local memory it requires.

### 6.1.2 Communication

As we mentioned, *information asymmetry* is the key challenge that decentralization brings in, and communication is one of the main ways to reduce information asymmetry and facilitate coordination among the agents. In the distributed optimization setting, agents communicate the local variables they keep. In the decentralized control setting, such as in the matching problem (introduced in Section 4.1.2.2 and solved in Section 5.6), communication is a type of *action* agents can actively take advantage of to produce *common information* (CI). In the networked MARL setting [135], the agents exchange policy parameters, while in the *centralized learning distributed execution* MARL setting, the agents can literally communicate anything during the learning phase.

Just as the case of local memory, in this thesis, we achieved *communication efficiency* by using the problem structures. While saving the need to memorize irrelevant variable parts in Chapter II due to the *partial dependency structure*, the agents also do not communicate irrelevant variable parts, as the decision of every part will be eventually dictated by the agents whose objectives truly depend on it. In Chapter V, we completely exempt the agents from communicating by exploiting the *hierarchi-*

*cal information structure*, as the more informed agents do more computations in the lower layers while the less informed agents in the upper layers assume optimality from lower layers.

Again, just like the case of local memory, tradeoffs exist between the amount of communication and other factors. In Section 2.5, we saw that the amount of communication per round is inversely related to the minimum iterations required for the objective function to reach $\epsilon$-optimal for some $\epsilon$. We also saw that judiciously adding highly connected nodes in the local networks can reduce the spectral radii and speed up convergence at the cost of increased communication. In the decentralized control setting such as the matching problem, communication is a type of action agents can use to achieve a higher reward. However, since communication itself is also costly, there will be an optimal communication scheme in any optimal policy, which may not be to always communicate.

### 6.1.3 Approximation

In this thesis, we used approximations to solve more general problems. In Chapter II, to accommodate cases where the gradient of the objective function is not Lipschitz continuous, we used a series of smooth functions to approximate the original objective function. In Chapter IV, we studied approximate state representations that are more scalable to general complex MARL settings. The scalability comes from the fact that the encapsulations for the representations compress the histories into quantities in *time-invariant* spaces, and are more practical as they are easier to determine. The approximation was done at the cost of a bounded loss from the optimality.

The multi-time-scale technique is another mathematical theme that appears throughout this thesis and also in optimization or learning context generally. In the technique, multiple processes are interlaced in every step so that eventually all the limiting conditions of the processes will be satisfied. The two distributed optimization algorithms we introduced in Chapter II, i.e., NEXT and DGD, are two time-scale stochastic approximation algorithms, where the consensus step (and the gradient update if there is any) is in the fast process, and the descent step is in the slow process. This was later identified in Chapter III, where we found the limiting condition of the consensus process is a projection onto the consensus plane, and thus studied a variety of subprojections that satisfy the condition for convergence speed-up. The function approximation scheme mentioned earlier also belongs to this technique, as we interleave the process of a chosen series of smooth functions converging to the original objective function into the steps. In the practical MARL framework proposed in Section 4.5.3, although not given in detail, the process of representation learning and the process of policy learning may also form a two time-scale stochastic approximation algorithm, where the representation learning will happen in the fast time-scale, and the policy learning in the slow time-scale, which is proposed in [112, 113]. In Chapter V, the hierarchical bandit algorithm and the hierarchical MDP algorithm are yet another set of multi-time-scale examples. In the algorithms, the upper layer agent explores slowly and hence is in the slow time-scale, and it assumes optimal reaction from the lower layer, which is the limiting condition of the fast learning process in the lower layer. In contrast to all the previously mentioned algorithms which use step-size (or learning rate) to control the time-scale, in these two algorithms, the multi-time-scale is achieved through an enlarged bonus term in the upper layer, which leads to increased exploration.

### 6.1.4 Coordination

The agents may coordinate to tackle the challenges posed by information asymmetry in MASs. An obvious coordination method is through communication as mentioned earlier. The goal of communication in a coordination scheme could be synchronizing the decentralization and bringing about consensus among the agents or could be giving more information for better decision making. Communicating the variables in distributed optimization settings and exchanging policy parameters in MARL settings belong to the former category, whereas the communication action in decentralized control settings is about giving information. Note that there is a major difference between the two categories: if the purpose of communication is synchronization, then when an agent sends new information, the old information it sent held by other agents is regarded as outdated and can simply be discarded. This is in general not the case when the goal of communication is to give more information about the system state, since this kind of information *accumulates* and is useful for future decision making.

Coordination among the agents does not necessarily require communication. The agents may *pre-coordinate* by setting a protocol that the agents follow in the process. The CI approach discussed in Chapter IV is an example of pre-coordination, where the agents maintain their consistency by using a sequential decomposition solely based on CI. The hierarchical RL methods proposed in Chapter V are also pre-coordination schemes that do not require communication. Instead of learning as if there were only one agent in the no coordination scheme, the lower layer agent does all the computations in the joint action space and the upper layer agent uses enlarged bonuses, which are planned before the learning process.

There are multiple perspectives one can view an MAS with an asymmetric information structure. In decentralized control/MARL settings, the *intersection* of the information held by all agents is accessible from the *coordinator's perspective*, based on which the sequential decomposition in the CI approach is carried out. In Chapter IV, we proposed the *supervisor's perspective*, from which the *union* of the information held by all agents is accessible. The supervisor's perspective proved to be extremely useful in analysis in Chapter IV. Note that depending on the considered information structure, there may be more insightful perspectives between these two extremes. For example, in [116], there are "team coordinators" who access the intersection of the information held by members of specific teams so that the scenario considered transforms into a game of the team coordinators.

## 6.2 Future Directions and Open Problems

This thesis studied cooperative operation of a set of agents. Hence, an obvious future direction would be to study the relevance and applicability of the results and methodologies to a competitive operation of a set of agents, namely a game [48], or even more complex settings such as games of teams [116]. Next we discuss other directions that are specific to each chapter.

### 6.2.1 General Analysis Framework for Distributed Optimization Algorithms

In Chapter II and Chapter III, some of the main ideas are: exploiting partial dependency structures to reduce the memory and communication overhead, using different consensus schemes for faster convergence, and using function approximations to relax the Lipschitz gradient assumption. However, all of them only apply to a small number of distributed optimization algorithms in our analysis. Specifically, for the former two ideas, although we mentioned we believe they can apply to most primal algorithms, we only showed the convergence when they are used in NEXT and DGD; for the function approximation, we only showed it applies to the NEXT algorithm. In reality, these ideas should have much wider applicability as similar notions appear ubiquitously. For example, the NEXT algorithm itself uses successive convex approximation which is a type of function approximation, and consensus step can be seen even in dual distributed optimization algorithms as well [39]. This necessitates a general analysis framework for distributed optimization algorithms, where any properties proved in the framework will apply to a large class of algorithms. Then under the framework, we can ask the following three questions: (1) How do we define partial dependency structures in this framework, and how do we manipulate the structures and characterize the tradeoffs between memory, communication, and convergence rate that probably associate with the manipulation? (2) What is the class of consensus schemes that lead to convergence to the optima, and which schemes are faster? Also, does the answer depend on objective functions or initial values, and if so, how? (3) What is the class of function approximation schemes we can interlace in one of the time-scales without losing the convergence guarantee?

### 6.2.2 Approximate Partial Dependency Structures

In Chapter II, we consider exact partial dependency, namely, we assume an objective function $f_i$ completely does not depend on some variable part $\mathbf{x}^m$. This is aligned with our numerical examples in Section 2.6, where the throughput of a BS does not depend on the resource allocation schemes of BSs farther than one-hop away. In reality, this is not true: the interference of a BS does not completely vanish beyond one-hop, it just becomes very weak. An ideal model for this is the *Dobrushin condition for weak dependence* [43], where the extent of coupling between two agents diminishes as a certain “measure” between the two agents gets large; in the above BS network example, the measure is simply spatial distance.

With the Dobrushin condition model, enforcing localization with an exact partial dependency structure as if there were no weak dependencies will result in an “localization error” in the objective value the iterates converge to. It is then of interest to study whether we can bound this error term, and how it depends on the localization scheme we use and the parameters in the Dobrushin model.

### 6.2.3 Differential Geometry for Consensus Schemes

In Chapter III, we take element-wise nonlinear transformations and average the iterates. These transformations are just special examples of *Riemannian manifolds* [70] which are invertible and preserve local geometric relations. In Chapter III, we saw that convergence was obtained by the Lipschitz continuity of the transformations. An open problem would be extending to transformations where the coordinates are transformed altogether instead of individually, and identifying the conditions of the transformations that preserve convergence.

### 6.2.4 Role of Information Structures in Decentralized Control and MARL

While we mentioned that in decentralized control, communication is a type of action that increases CI, such treatment only leads to an optimal policy with specifications of the communication scheme with very little intuition. In contrast, we can see many signaling strategies in the game of Hanabi [10], including finessing, bluffing, reverse finessing, etc. A natural question to investigate is whether there exists a communication framework specific enough so that we can obtain structural results characterizing various communication strategies, but also broad enough to include a variety of problems. Moreover, the role of communication in decentralized control only reveals information regarding the *supervisor’s state*, whereas in the MARL setting communication can additionally reveal agents’ learning status. Communicating through certain signal spaces to synchronize agents’ learning statuses requires less than sending the whole policy parameters and has not been studied. This might be a key to fully decentralized learning schemes.

For the general decentralized control/MARL representation framework developed in Chapter IV, it is also of interest to study if the method will simplify for specific information structures. On the other hand, the methods proposed in Chapter V rely on the hierarchical information structure. We are curious about whether similar near-optimal single-agent-like methods can be found for other information structures.

# APPENDICES

# APPENDIX A

# Proofs and Supplementary Material for Chapter II

## A.1 Generalizations to time-varying graphs

In this appendix we provide the less strict assumptions to accommodate the case where the underlying graph is time-varying and directed. The proof of our result is based on this more general setting. But the purpose of these assumptions remains the same – a distribution on the set of nodes will go to the uniform distribution exponentially fast with repeated application of the $W$ matrix, which is captured in Lemma A.2.

At each time slot $t$, the set of nodes $\mathcal{N}$ along with a set of time-variant directed edges $\mathcal{E}[t]$, form an directed graph $\mathcal{G}[t] = (\mathcal{N}, \mathcal{E}[t])$. Node $j$ can only send message to node $i$ in time slot $t$ if $j \to i \in \mathcal{E}[t]$.

**Assumption L$'$**

**(L3$'$)** $\mathcal{G}_m[t]$ is $B_m$-strongly connected for all $m \in [M+1]$, where $\mathcal{G}_m[t] = (\mathcal{N}_m, \mathcal{E}_m[t] = \{i \to j \in \mathcal{E}[t] : i \in \mathcal{N}_m \text{ or } j \in \mathcal{N}_m\})$, i.e. $(\mathcal{N}_m, \bigcup_{t=kB_m}^{(k+1)B_m-1} \mathcal{E}_m[t])$ is strongly connected for all $k \geq 0$;

**(L4$'$)** For all $m \in [M+1]$ there is a matrix $\mathbf{W}^m[t]$ associated with $\mathcal{N}_m$ – each entry is non-zero if and only if there is a corresponding edge in $\mathcal{E}_m[t]$, and all positive entries must be greater than or equal to some fixed $\vartheta > 0$. As before, $\mathbf{W}^m[t]$ is doubly-stochastic after deleting the zero rows and columns whose indices are not in $\mathcal{E}_m[t]$.

## A.2 Proof of Theorem 2.3

In this appendix we prove Theorem 2.3. We start with an intuitive description of the roadmap of the entire proof in the following. Our proof follows the main structure of [80]. The convergence of NEXT basically consists of two parts – at the faster time-scale the "consensus convergence" of local variable iterates $\mathbf{x}_i$ and local gradient iterates $\mathbf{y}_i$ to the average iterate $\bar{\mathbf{x}}$ and $\overline{\nabla f}(\bar{\mathbf{x}}) \triangleq \sum_{i \in \mathcal{N}} \nabla f_i(\bar{\mathbf{x}})$ (the average gradient evaluated at the average iterate), respectively, and on the slower time-scale the fixed point iterations of $\hat{\mathbf{x}}_i(\bullet)$ (defined in (A.25)) which we call "path convergence" – for details of this viewpoint and connection to two time-scale stochastic approximation see Section 3.2. It is shown in [80] that the fixed points of $\hat{\mathbf{x}}_i(\bullet)$ coincide with the stationary points. The goal is thus to prove the iterates converge to the fixed points of $\hat{\mathbf{x}}_i(\bullet)$ and the iterates of all nodes asymptotically agree. Our contributions are introducing partial dependency structure with a localization scheme and in addition successively approximating the possibly not-sipschitz gradient objective functions. The latter significantly increases the proof hardness as the gradients of the objective functions may now be unbounded.

The core idea of NEXT as in most primal algorithms, is in performing the following steps iteratively.

- The local optimization step for each node $i$ is to find the value of $\hat{\mathbf{x}}_i$ at its iterate $\mathbf{x}_i$. This $\hat{\mathbf{x}}_i$ maps the iterate to the optimum of an approximated strongly convex version of the overall objective function $U$, which involves using a strongly convex surrogate for $i$'s own objective function, and a linearized approximation of others objective functions. This step is similar to doing gradient descent in a much more efficient way by taking advantage of the surrogates, and corresponds to the "path convergence" mentioned above.

- The consensus step is where every node communicates its iterate to its neighbors and also takes an average of its neighbors' iterates. We refer to the $\mathbf{x}_i$'s asymptotically agreeing on their average $\bar{\mathbf{x}}$ as the "$x$-consensus convergence."

To make the algorithm more practical and fully decentralized, the original *Inexact NEXT* [80] and our *Localized Proximal Inexact NEXT* add multiple layers of approximations.

(1) We mentioned that the node $i$ linearize other nodes' gradients with the slope $\pi_i \triangleq \sum_{j \in \mathcal{N} \setminus \{i\}} \nabla f_j(\mathbf{x}_i)$. However, node $i$ only has the information of $\nabla f_i(\mathbf{x}_i)$. It hence keeps a variable $\mathbf{y}_i$ that tracks the average gradient $\frac{1}{n} \sum_{j \in \mathcal{N}} \nabla f_j$ so that node $i$ can approximate $\pi_i$ by $\tilde{\pi}_i$ with the knowledge of $\nabla f_i(\mathbf{x}_i)$ and $\mathbf{y}_i$ as in Algorithm 2.2. The convergence of $\mathbf{y}_i$ to the average gradient is then the "$y$-consensus convergence." The convergence is also achieved through a gossip-type consensus scheme similar to that used for $\mathbf{x}$.

(2) With the approximation scheme using the $y$ variable, the fixed point iteration we actually use in Algorithm 2.2 for the local optimization step is $\tilde{\mathbf{x}}_i$ defined in (2.5), which is similar to $\hat{\mathbf{x}}_i$ but using $\tilde{\pi}_i$ as the linearization constant. This makes the algorithm viable in practice in a fully decentralized scenario. To compare the behaviors of $\hat{\mathbf{x}}_i(\bar{\mathbf{x}})$ and $\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_i^*(\mathbf{x}_i, \tilde{\pi}_i)$, we construct a new "averaging system" assuming that the $\mathbf{x}_i$'s already converge to $\bar{\mathbf{x}}$; this includes $\mathbf{y}_i^{av}$, the average gradient evaluated at $\bar{\mathbf{x}}$, and $\tilde{\mathbf{x}}_i^{av} = \tilde{\mathbf{x}}^*(\bar{\mathbf{x}}, \tilde{\pi}_i^{av})$, the local optimization map where $\tilde{\pi}_i^{av}$ computed from $\mathbf{y}_i^{av}$ is used as the linearization constant. Note that this "averaging system" is constructed purely for analysis purposes.

(3) We use a series of functions $\{\tilde{f}_{i,t}^*\}$ to approximate $f_i$ so that the local optimization map with ideal linearization is $\hat{\mathbf{x}}_{i,t}$, and $\tilde{\mathbf{x}}_i^*$ with the $\mathbf{y}$-approximation in contrast to just $\tilde{\mathbf{x}}_i$ in *NEXT*. This is the approximation we add in addition to what was done in *Inexact NEXT* [80].

(4) The inexactness of the algorithm chooses $\mathbf{x}_i^{inx}$ within $\epsilon_i$ range of $\tilde{\mathbf{x}}_i$, which leads to another source of approximation. Because of the function approximation we use with the relaxed constraints on the schedules of $\{L_{i,t}\}$ and $\{\tau_{i,t}\}$, the difference between $\mathbf{x}_i^{inx}$ and $\mathbf{x}_i$ can potentially become increasingly larger if the error propagates, which also increases the proof hardness in contrast to the case of *inexact NEXT* where the difference $\|\mathbf{x}_i^{inx} - \tilde{\mathbf{x}}_i\|$ is bounded.

The proof of Theorem 2.3 consists of six parts. The theorem basically makes two claims: the nodes' iterates asymptotically agree, and they converge to one of the optima. The former will be the side product as we aim to prove the latter. In the first part of the proof, we first summarize the list of notations that will be used in the proof in Appendix A.2.1, and then describe a few results and one key proposition in Appendix A.2.2. Proposition A.1 as the variant of Proposition 5 in [80] shows Lipschitz properties of $\hat{\mathbf{x}}_{i,t}$. Lemma A.2 and Fact A.4 describe the main machinery we use to show the "$x$-consensus convergence" and "$y$-consensus convergence," that is, the *geometric convergence* of the product of doubly stochastic matrices to the all one matrix. The results Lemma A.5, Lemma A.6, Lemma A.7, and Technical Assumption T are technical lemmas regarding series or summations that arise in the analysis. The key proposition is Proposition A.3, which constitutes the core components of the proof.

We prove Proposition A.3 (a) in the second part Appendix A.2.3, which says the difference between $\mathbf{x}_i^{inx}$ and $\mathbf{x}_i$ cannot grow beyond a certain rate, and the tools used are the definition of minimization (2.5), the strong convexity of $\tilde{f}_{i,t}^*$, and the $y$-consensus convergence. The difference is decomposed into $\|\mathbf{x}_i^{inx} - \tilde{\mathbf{x}}_i\|$ and $\|\tilde{\mathbf{x}}_i - \mathbf{x}_i\|$, where the former is bounded by $\epsilon_i[t]$ by definition, and the latter is bounded by $O\left(\frac{L_{i,t}}{\tau_{i,t}}\right)$ using a mathematical induction argument.

Part (b) of Proposition A.3 establishes asymptotic consensus on $\mathbf{x}$ among nodes as the third part of the proof Appendix A.2.4. This part involves exploiting Lemma A.2, Lemma A.5, Fact A.4, Part (a) of Proposition A.3, and series convergence. The generalization to multiple local dependency sets is also taken care of in Part (a) and (b) of Proposition A.3 using simple inequalities regarding multi-dimensional spaces.

In the fourth part contained in Appendix A.2.5, Part (c) of Proposition A.3 is proved, which shows that the locally-optimized result using the "$y$ approximation" $\tilde{\mathbf{x}}_i^{av}$ converges to the locally-optimized result with ideal linearization $\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}})$ evaluated at $\bar{\mathbf{x}}$. In addition to applying the definitions of the maps, it is intuitive that the "$y$-consensus convergence" in the "averaging system" and hence Lemma A.2 play a crucial role in the proof; Part (a) of Proposition A.3 and Technical Assumption T which comes from Lemma A.7 and the conditions of Theorem 2.3 are also used.

We prove Part (d) of Proposition A.3 in the fifth part Appendix A.2.6. Part (d) claims the actual locally-optimized result in the algorithm $\tilde{\mathbf{x}}_i$ converges to the locally-optimized result using the "$y$ approximation" $\tilde{\mathbf{x}}_i^{av}$. The underlying reason of the convergence is the "$x$-consensus convergence." Several previous results are all used in this proof, including all of Parts (a), (b), and (c), Lemma A.5, Lemma A.7, and

Technical Assumption T.

Appendix A.2.7 then combines Parts (a), (c), and (d), convexity of $G$, and Lemma A.6 to show that $\bar{\mathbf{x}}$ converges to $\hat{\mathbf{x}}_{i,\infty}(\bar{\mathbf{x}})$. With $\tau_t$ going to 0, $\hat{\mathbf{x}}_{i,\infty}$ is no longer a function but a correspondence; variational analysis is thus introduced to deal with the minimizers of correspondences in the first case of Appendix A.2.7. Finally, we deal with unbounded gradient interior point in the second case of Appendix A.2.7, using convexity and series convergence. Generalization to study the case of an unbounded gradient boundary point is left as an open question.

Comparing to the proof of *NEXT*, the generalization to multiple local dependency set is not a technically hard one; it requires some simple inequalities as shown in part (a) and (b) of Proposition A.3. For the second generalization, we replace what was Lipschitz constant $L$ and strongly convex constant $\tau$ by series $\{L_t\}$ and $\{\tau_t\}$, which now could grow to infinity and decrease to zero, respectively. This does significantly increase the hardness of the proof. The conditions in Theorem 2.3, the Technical Assumption T stated below, and Lemma A.7 are made such that all the series now with $\{L_t\}$ and $\{\tau_t\}$ still converge. The unbounded gradient issue and the correspondence nature of $\hat{\mathbf{x}}_{i,\infty}$ also make our scheme much trickier to analyze in comparison to *NEXT*.

### A.2.1 Notations

We define a set of notations to proceed with the proof. All notation is defined for all $i$, $m$, and $t$, whenever applicable.

**Original variables**

- $\mathbf{x}^m[t] = (\mathbb{I}\{i \in \mathcal{N}_m\}\mathbf{x}_i^m[t])_{i\in\mathcal{N}} = [\mathbb{I}\{1 \in \mathcal{N}_m\}\mathbf{x}_1^m[t]^\top \ \cdots \ \mathbb{I}\{I \in \mathcal{N}_m\}\mathbf{x}_I^m[t]^\top]^\top$: the concatenation of part $m$ decision variables from all nodes in $\mathcal{N}_m$ with padded zero for nodes not in $\mathcal{N}_m$; we also use $\mathbf{x}^m[t]$ to refer to non-padded zero version $(\mathbf{x}_i^m[t])_{i\in\mathcal{N}_m}$, i.e. the vector containing only $\mathbf{x}_i^m[t]$ when $i$ is in $\mathcal{N}_m$, when the context is clear; the notation $(v_i)_{i\in\mathcal{S}}$, which denotes the vector concatenated from all the vectors of the form $v_i$ where the index $i$ is in the set $\mathcal{S}$, will be used throughout this Appendix

- $\mathbf{y}^m[t] = (\mathbb{I}\{i \in \mathcal{N}_m\}\mathbf{y}_i^m[t])_{i\in\mathcal{N}}$: the concatenation of part $m$ of $\mathbf{y}$, which tracks the average (among nodes) gradients of $\nabla_{\mathbf{x}^m} f_i$ from the nodes in $\mathcal{N}_m$ in the algorithm

- $\mathbf{r}^m[t] = (\nabla_{\mathbf{x}^m} f_{i,t}^*[t])_{i\in\mathcal{N}}$: the concatenation of ground truth gradient, with element $\nabla_{\mathbf{x}^m} f_{i,t}^*[t] = \nabla_{\mathbf{x}^m} f_{i,t}^*(\mathbf{x}_i[t])$; adding $\mathbb{I}\{i \in \mathcal{N}_m\}$ is unnecessary as the gradient would be zero for those nodes not depending on $\mathbf{x}^m$
- $\Delta\mathbf{r}^m[s,t] = (\Delta\mathbf{r}_i^m[s,t])_{i\in\mathcal{N}}$: the gradient difference, with $\Delta\mathbf{r}_i^m[s,t] = \nabla_{\mathbf{x}^m} f_{i,s}^*[s] - \nabla_{\mathbf{x}^m} f_{i,t}^*[t]$ $(s \le t)$
- $\tilde{\pi}_i[t]$: see Line 11 of Algorithm 2.2
- $\tilde{\mathbf{x}}_i[t] = \tilde{\mathbf{x}}_i^*(\mathbf{x}_i[t], \tilde{\pi}_i[t])$: see Line 5 of Algorithm 2.2 and (2.5)

**Average variables**

- $\bar{\mathbf{x}}^m[t] = \frac{1}{n_m}\sum_{i\in\mathcal{N}_m} \mathbf{x}_i^m[t]$: average of decision variable
- $\bar{\mathbf{y}}^m[t] = \frac{1}{n_m}\sum_{i\in\mathcal{N}_m} \mathbf{y}_i^1[t]$: average of gradient tracking variable
- $\bar{\mathbf{r}}^m[t] = \frac{1}{n_m}\sum_{i\in\mathcal{N}_m} \nabla_{\mathbf{x}^m} f_{i,t}^*[t]$: average of ground truth gradient
- $\Delta\bar{\mathbf{r}}^m[s,t] = \frac{1}{n_m}\sum_{i\in\mathcal{N}_m} \Delta\mathbf{r}_i^m[s,t]$: average of gradient difference

**Tracking system using average variables**

- $\nabla f_{i,t}^{*,av}[t] = \nabla f_{i,t}^*(\bar{\mathbf{x}}[t])$
- $\mathbf{r}^{m,av}[t] = (\nabla_{\mathbf{x}^m} f_{i,t}^{*,av}[t])_{i\in\mathcal{N}}$: the concatenation of ground truth gradient evaluated at average decision variable
- $\Delta\mathbf{r}^{m,av}[s,t] = (\Delta\mathbf{r}_i^{m,av}[s,t])_{i\in\mathcal{N}}$: the gradient difference evaluated at average decision variable, with $\Delta\mathbf{r}_i^{m,av}[s,t] = \nabla_{\mathbf{x}^m} f_{i,t}^{*,av}[s] - \nabla_{\mathbf{x}^m} f_{i,t}^{*,av}[t]$
- $\mathbf{y}_i^{m,av}[t+1] = \sum_j w_{ij}^m[t]\mathbf{y}_j^{m,av}[t] + \Delta\mathbf{r}_i^{m,av}[t+1,n]$: tracking of average gradient evaluated at average decision variable, with $\mathbf{y}_i^{m,av}[0] = \nabla_{\mathbf{x}^m} f_{i,t}^{*,av}[0]$; concatenating $\mathbf{y}_i^{m,av}[t+1]$ for $i \in \mathcal{N}$ makes $\mathbf{y}^{m,av}[t] = (\mathbb{I}\{i \in \mathcal{N}_m\}\mathbf{y}_i^{m,av}[t])_{i\in\mathcal{N}}$
- $\tilde{\pi}_i^{m,av}[t] = n_m\mathbf{y}_i^{m,av}[t] - \nabla_{\mathbf{x}^m} f_{i,t}^{*,av}[t]$

- $\tilde{\mathbf{x}}_i^{av}[t] = \tilde{\mathbf{x}}_i^*(\bar{\mathbf{x}}_i[t], \tilde{\pi}_i^{av}[t])$: optimized result evaluated at average decision variable and average tracking system

- $\bar{\mathbf{r}}^{m,av}[t] = \frac{1}{n_m} \sum_{i \in \mathcal{N}_m} \nabla_{\mathbf{x}^m} f_{i,t}^{*,av}[t]$: average of ground truth gradient evaluated at average decision variable

**Doubly stochastic matrices**

- $\mathbf{P}^m[t,s] = \mathbf{W}^m[t]\mathbf{W}^m[t-1] \cdots \mathbf{W}^m[s] \quad t \geq s$

- $\hat{\mathbf{W}}^m[t] = \mathbf{W}^m[t] \otimes I_{d_m}$

- $\hat{\mathbf{P}}^m[t,s] = \hat{\mathbf{W}}^m[t]\hat{\mathbf{W}}^m[t-1] \cdots \hat{\mathbf{W}}^m[s] = \mathbf{P}^m[t,s] \otimes \mathbf{I}_{d_m} \quad t \geq s$

- $J^m = \frac{1}{n_m}\mathbf{1}_{\mathcal{N}_m}\mathbf{1}_{\mathcal{N}_m}^\top \otimes \mathbf{I}_{d_m}$, where $\mathbf{1}_{\mathcal{N}_m} = \{\mathbb{I}\{i \in \mathcal{N}_m\} : i \in \mathcal{N}\}$, and $\mathbf{I}$ is the identity matrix

- $J_\perp^m = \mathbf{I}_{d_m n_m} - J^m$, where $\mathbf{I}_{d_m n_m}$ is the identity matrix with dimension $d_m \times n_m$

### A.2.2 Key Propositions

The next proposition is a variant of Proposition 5 in [80].

**Proposition A.1:** *Let* $\pi_i^{\mathcal{S}_i}(\tilde{\mathbf{x}}) = \sum_{j \neq i} \nabla_{\mathbf{x}^{\mathcal{S}_i}} f_{j,t}^*(\tilde{\mathbf{x}}) = \left(\sum_{j \in \mathcal{N}_m, j \neq i} \nabla_{\mathbf{x}^m} f_{j,t}^*(\tilde{\mathbf{x}}^{\mathcal{N}_m})\right)_{m \in \mathcal{S}_i}$ *be the concatenation of* $\sum_{j \in \mathcal{N}_m, j \neq i} \nabla_{\mathbf{x}^m} f_{j,t}^*(\tilde{\mathbf{x}}^{\mathcal{N}_m})$ *for all* $m$ *in* $\mathcal{S}_i$*. Define the mapping* $\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\cdot) : \mathcal{K} \to \mathcal{K}_{\mathcal{S}_i} = \Pi_{m \in \mathcal{S}_i}\mathcal{K}_m$ *by*

$$\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}^{\mathcal{S}_i}} \ \tilde{f}_{i,t}^*(\mathbf{x}^{\mathcal{S}_i}; \tilde{\mathbf{x}}^{\mathcal{S}_i}) + \pi_i^{\mathcal{S}_i}(\tilde{\mathbf{x}})^\top(\mathbf{x}^{\mathcal{S}_i} - \tilde{\mathbf{x}}^{\mathcal{S}_i}) + G(\mathbf{x}^c) \quad \forall \ i \in \mathcal{N}, \tag{A.1}$$

*and the mapping* $\hat{\mathbf{x}}_{i,t}(\cdot) : \mathcal{K} \to \mathcal{K}$ *by* $\hat{\mathbf{x}}_{i,t}(\tilde{\mathbf{x}}) = \left(\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\tilde{\mathbf{x}}), \tilde{\mathbf{x}}^{\mathcal{N} \backslash \mathcal{S}_i}\right)$*, that is, preserving everything in the* $\mathcal{K}_{\mathcal{N} \backslash \mathcal{S}_i}$ *subspace unchanged while mapping with* $\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\cdot)$ *in the* $\mathcal{S}_i$ *subspace. Then, under Assumptions A, F, and N, the mapping* $\hat{\mathbf{x}}_{i,t}(\cdot)$ *has the following properties:*

*(a)* $\forall \ \mathbf{z} \in \mathcal{K}$ *and* $i \in \mathcal{N}$,

$$(\hat{\mathbf{x}}_{i,t}(\mathbf{z}) - \mathbf{z})^\top \nabla F(\mathbf{z}) + G(\hat{\mathbf{x}}_{i,t}(\mathbf{z})) - G(\mathbf{z}) \leq -\tau_t^{\min}\|\hat{\mathbf{x}}_{i,t}(\mathbf{z}) - \mathbf{z}\|^2,$$

*where* $F(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$*. Here we use* $G(\mathbf{x})$ *and* $G(\mathbf{x}^c)$ *interchangeably as they are the same thing.*

(b) $\hat{\mathbf{x}}_{i,t}(\cdot)$ *is Lipschitz continuous, i.e.* $\|\hat{\mathbf{x}}_{i,t}(\mathbf{w}) - \hat{\mathbf{x}}_{i,t}(\mathbf{z})\| \leq L_{i,t}\|\mathbf{w}-\mathbf{z}\| \quad \forall\ \mathbf{w}, \mathbf{z} \in \mathcal{K}$ *for* $i \in \mathcal{N}$[1]*.*

The only thing we do is to substitute $\tilde{f}_i$ in [80] as $\tilde{f}^*_{i,t}$. Although the equations are written in localization form, it does not really change anything here.

**Lemma A.2:** *Define* $\mathbf{P}[t,s] \triangleq \mathbf{W}[t]\mathbf{W}[t-1]\cdots\mathbf{W}[s]$*. Then under Assumption L′(doubly stochasticity and lower bounded entries for edges),*

$$\left\|\mathbf{P}[t,s] - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right\| \leq c_0\rho^{t-1+1},\ \ \forall\ t \geq s$$

*for some* $c_0 > 0$ *and* $\rho \in (0,1)$*.*

Strictly speaking the above is for the $\mathbf{W}^c = \mathbf{W}^{M+1}$, which is the matrix used for the averaging of the entire network, in Assumption L′. For $\mathbf{W}^m$ where $m \in [M]$, the Lemma also holds after we delete the zero rows/columns. In this case the product converges to $\frac{1}{n_m}\mathbf{1}\mathbf{1}^\top$ where the $\mathbf{1}$ is of the proper dimension. From now on we will take $\rho$ as the largest geometric convergence factor among all $\mathbf{W}^m$, $m \in [M+1]$.

Before proving Theorem 2.3, we will first prove the following proposition.

**Proposition A.3:** *Let* $\{\mathbf{x}^m[t]\}_n \triangleq \{(\mathbf{x}_i^m[t])_{i\in\mathcal{N}_m}\}$ *and* $\{\bar{\mathbf{x}}^m[t]\}_n \triangleq \left\{\frac{1}{n_m}\sum_{i\in\mathcal{N}_m}\mathbf{x}_i^m[t]\right\}_n$*,* $m \in [M+1]$ *be the sequences generated by Algorithm 2.2, in the settings of the Theorem 2.3. Then the following holds:*

(a) *For all* $n$*,* $\|\mathbf{x}_i^{m,inx}[t] - \mathbf{x}_i^m[t]\| \leq \frac{c^m L_{i,t}}{\tau_{i,t}} \quad \forall\ i \in \mathcal{N}_m, m \in [M+1]$*.*

(b) $\lim_{t\to\infty}\|\mathbf{x}_i^m[t] - \bar{\mathbf{x}}^m[t]\| = 0$*,* $\sum_{t=1}^{\infty}\alpha[t]\|\mathbf{x}_i^m[t] - \bar{\mathbf{x}}^m[t]\| < \infty$*,* $\sum_{t=1}^{\infty}\|\mathbf{x}_i^m[t] - \bar{\mathbf{x}}^m[t]\|^2 < \infty \quad \forall\ i \in \mathcal{N}_m, m \in [M+1]$*.*

(c) $\lim_{t\to\infty}\|\tilde{\mathbf{x}}_i^{av}[t] - \hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}[t])\| = 0$*,* $\sum_{t=1}^{\infty}\alpha[t]L_t^{\max}\|\tilde{\mathbf{x}}_i^{av}[t] - \hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}[t])\| < \infty \quad \forall\ i \in \mathcal{N}$*.*

---

[1] Note that this holds for $\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}$ as well because the elements in the $\mathcal{K}_{\mathcal{N}\setminus\mathcal{S}_i}$ subspace just cancel each other out.

*(d)* $\lim_{t\to\infty}\|\tilde{\mathbf{x}}_i^m[t]-\tilde{\mathbf{x}}_i^{m,av}[t]\|=0$, $\sum_{t=1}^{\infty}\alpha[t]L_t^{\max}\|\tilde{\mathbf{x}}_i^m[t]-\tilde{\mathbf{x}}_i^{m,av}[t]\|<\infty \quad \forall\ i\in\mathcal{N}_m, m\in[M+1]$.

We will use the following assumption frequently when proving Proposition A.3. These are the exact technical inequalities we use in the proof, while all of them are implicitly implied by the conditions of Theorem 2.3 as we will show below.

**Technical Assumption T**

**(T1)** $\lim_{t\to\infty}\rho^t\frac{L_t^{\max}}{\tau_t^{\min}}=0$;

**(T2)** $\lim_{t\to\infty}\frac{L_t^{\max}}{\tau_t^{\min}}\sum_{s=0}^{t-1}\rho^{t-s}\frac{\alpha[s](L_s^{\max})^2}{\tau_s^{\min}}=0$;

**(T3)** $\lim_{t\to\infty}\frac{1}{\tau_t^{\min}}\sum_{s=0}^{t-1}\rho^{t-s}\|\nabla f_{i,s}^*(\mathbf{x})-\nabla f_{i,s-1}^*(\mathbf{x})\|=0$ for all $\mathbf{x}$ and $i$;

**(T4)** $\sum_{t=1}^{\infty}\rho^t\frac{L_t^{\max}}{\tau_t^{\min}}<\infty$;

**(T5)** $\sum_{t=1}^{\infty}\frac{\alpha[t]L_t^{\max}}{\tau_t^{\min}}\sum_{s=0}^{t-1}\rho^{t-s}\frac{\alpha[s](L_s^{\max})^2}{\tau_s^{\min}}<\infty$;

**(T6)** $\sum_{t=1}^{\infty}\frac{\alpha[t]L_t^{\max}}{\tau_t^{\min}}\sum_{s=0}^{t-1}\rho^{t-s}\|\nabla f_{i,s}^*(\mathbf{x})-\nabla f_{i,s-1}^*(\mathbf{x})\|<\infty$ for all $\mathbf{x}$ and $i$.

**Proof:**

**(T1)**: it should be evident from the conditions of Theorem 2.3 that none of the parameters could be growing or decaying at an exponential rate. We are considering the setting where $\alpha[t]$ and $\tau_t^{\min}$ are going to zero while $L_t^{\max}$ is going to infinity. The condition $\sum_{t=0}^{\infty}(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2<\infty$ implies that if either $L_t^{\max}$ is growing exponentially or $\tau_t^{\min}$ is decaying exponentially, then $\alpha[t]$ must also be decaying exponentially. But then $\sum_{t=0}^{\infty}\tau_t^{\min}\alpha[t]=\infty$ would never be possible.

**(T2)**: recall the conditions of Theorem 2.3 imply $\lim_{t\to\infty}\alpha[t]\frac{(L_t^{\max})^3}{(\tau_t^{\min})^3}=0$, then apply the first part of Lemma A.7.

**(T3)**: from $\lim_{t\to\infty}\frac{\eta_n^{\max}}{\tau_t^{\min}}=0$ and the first part of Lemma A.7.

**(T4)**: again the parameters are not growing at an exponential rate.

**(T5)**: from $\sum_{t=0}^{\infty}(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2<\infty$ and the second part of Lemma A.7.

**(T6)**: from $\sum_{t=0}^{\infty}\frac{\alpha[t]L_t^{\max}\eta_n^{\max}}{\tau_t^{\min}}<\infty$ and the second part of Lemma A.7. ■

### A.2.3 Proof of Proposition A.3 (a)

Consider a local dependency set $\mathcal{N}_m$ and any node $i \in \mathcal{N}_m$. By the definition of $\mathbf{x}_i^{\mathcal{S}_i}$ defined in the minimization of (2.5), we have

$$\sum_{m\in\mathcal{S}_i} (\mathbf{x}_i^m[t]-\tilde{\mathbf{x}}_i^m[t])^\top \left[\nabla_{\mathbf{x}_i^m} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{\mathcal{S}_i}[t];\mathbf{x}_i^{\mathcal{S}_i}[t]) + \tilde{\pi}_i^m[t]\right] + (\mathbf{x}_i^c[t]-\tilde{\mathbf{x}}_i^c[t])^\top \partial G(\tilde{\mathbf{x}}_i^c[t]) \geq 0. \tag{A.2}$$

From the Line 11 of Algorithm 2.2 and (F2′), we have

$$\tilde{\pi}_i^m[t] = n_m \cdot \mathbf{y}_i^m[t] - \nabla_{\mathbf{x}_i^m} \tilde{f}_{i,t}^*(\mathbf{x}_i^{\mathcal{S}_i}[t];\mathbf{x}_i^{\mathcal{S}_i}[t]).$$

Substitute this result into A.2 and rearrange the terms to get

$$\begin{aligned} &\tau_{i,t}\|\mathbf{x}_i^{\mathcal{S}_i}-\tilde{\mathbf{x}}_i^{\mathcal{S}_i}\|^2 = \tau_{i,t}\sum_{m\in\mathcal{S}_i}\|\mathbf{x}_i^m-\tilde{\mathbf{x}}_i^m\|^2 \\ &\leq (\mathbf{x}_i^{\mathcal{S}_i}-\tilde{\mathbf{x}}_i^{\mathcal{S}_i})^\top \cdot [\nabla_{\mathbf{x}_i^{\mathcal{S}_i}}\tilde{f}_{i,t}^*(\mathbf{x}_i^{\mathcal{S}_i};\mathbf{x}_i^{\mathcal{S}_i}) - (\nabla_{\mathbf{x}_i^{\mathcal{S}_i}}\tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{\mathcal{S}_i};\mathbf{x}_i^{\mathcal{S}_i})] \quad \text{(strong convexity of } \tilde{f}_{i,t}^*\text{)} \\ &\leq \sum_{m\in\mathcal{S}_i}(\mathbf{x}_i^m-\tilde{\mathbf{x}}_i^m)^\top(n_m\mathbf{y}_i^m) + (\mathbf{x}_i^c-\tilde{\mathbf{x}}_i^c)^\top\partial G(\tilde{\mathbf{x}}_i^c)] \qquad \text{(from A.2)} \\ &\leq \sum_{m\in\mathcal{S}_i}(n_m\|\mathbf{y}_i^m\|)\cdot\|\mathbf{x}_i^m-\tilde{\mathbf{x}}_i^m\| + L_G\|\mathbf{x}_i^c-\tilde{\mathbf{x}}_i^c\| \qquad \text{(C-S inequality+(A2)).} \end{aligned} \tag{A.3}$$

We have omitted all the time indices in A.3 since all the variables have the same time index $[t]$.

Suppose that $\|\mathbf{y}_i^m\|$ is bounded by $l_m L_{i,t}$ for all $m \in \mathcal{S}_i$. Then A.3 is of the form

$$\tau_{i,t}\sum_{m\in\mathcal{S}_i}\|\mathbf{x}_i^m-\tilde{\mathbf{x}}_i^m\|^2 \leq \sum_{m\in\mathcal{S}_i} l_m L_{i,t}\|\mathbf{x}_i^m-\tilde{\mathbf{x}}_i^m\|, \tag{A.4}$$

which implies that all $\|\mathbf{x}_i^m-\tilde{\mathbf{x}}_i^m\|$'s are bounded by $\frac{\sum_{m\in\mathcal{S}_i} l_m L_{i,t}}{\tau_{i,t}}$[2]. This is due to the following argument: if $\{x_i\}, \{l_i\}$ are non-negative and $\sum_i x_i^2 \leq \sum_i l_i x_i$, then $\max\{x_i\} \leq \sum_i l_i$; otherwise, W.O.L.G. we can assume $x_1 = \max\{x_i\}$ and hence

---

[2] Note that $\mathbf{x}^c$ refers to $\mathbf{x}^{M+1}$ and $M+1$ is in $\mathcal{S}_i$ as well if the part exists. Therefore, the second term $L_G\|\mathbf{x}_i^c-\tilde{\mathbf{x}}_i^c\|$ can be put into the summation in the first term.

$\sum_i l_i < x_1$, then the following holds

$$\sum_i x_i^2 > x_1^2 > x_1 \sum_i l_i > \sum_i l_i x_i,$$

which is a contradiction. Thus, with A.4, we get

$$\begin{aligned} \|\mathbf{x}_i^{m,inx}[t] - \mathbf{x}_i^m[t]\| &\leq \|\mathbf{x}_i^{m,inx}[t] - \tilde{\mathbf{x}}_i^m[t]\| + \|\tilde{\mathbf{x}}_i^m[t] - \mathbf{x}_i^m[t]\| \\ &\leq \epsilon_i^m[t] + \frac{\sum_{m\in\mathcal{S}_i} l_m L_{i,t}}{\tau_{i,t}} \leq \frac{c^m L_{i,t}}{\tau_{i,t}}, \end{aligned}$$

where $c^m$ is some constant independent of $n$ and $i$. This proves the claim. It only remains to show that $\|\mathbf{y}_i^m\|$ is actually bounded by $l_m L_{i,t}$.

We use mathematical induction to finish the proof. The statement is that

$$\|\Delta\mathbf{x}_i^{m,inx}[t]\| = \|\mathbf{x}_i^{m,inx}[t] - \mathbf{x}_i^m[t]\| \leq \frac{c^m L_{i,t}}{\tau_{i,t}}, \;\; \|\mathbf{y}_i^m[t]\| \leq l_m L_{i,t}$$

holds for all $n$. We have already shown that the latter implies the former. The base case is obvious as we initialize $\mathbf{y}_i^m[0]$ to be $\nabla_{\mathbf{x}^m} f_{i,0}^*[0]$, which is assumed to be Lipschitz continuous. For the induction step, we assume the statement is true for $t-1$ and proved the latter part $\|\mathbf{y}_i^m[t]\| \leq l_m L_{i,t}$ holds for $n$.

By the definition of $\mathbf{y}$, we have

$$\mathbf{y}_i^m[t] = \hat{\mathbf{W}}^m[t-1]\mathbf{y}_i^m[t-1] + \Delta\mathbf{r}_i^m[t,t-1]$$

where

$$\begin{aligned} &\|\Delta\mathbf{r}_i^m[t,t-1]\| \\ =& \|\nabla_{\mathbf{x}^m} f_{i,t}^*[t] - \nabla_{\mathbf{x}^m} f_{i,t-1}^*[t-1]\| = \|\nabla f_{i,t}^*(\mathbf{x}_i[t]) - \nabla_{\mathbf{x}^m} f_{i,t-1}^*(\mathbf{x}_i[t-1])\| \\ \leq& \|\nabla_{\mathbf{x}^m} f_{i,t}^*(\mathbf{x}_i[t]) - \nabla_{\mathbf{x}^m} f_{i,t}^*(\mathbf{x}_i[t-1]) + \nabla_{\mathbf{x}^m} f_{i,t}^*(\mathbf{x}_i[t-1]) - \nabla_{\mathbf{x}^m} f_{i,t-1}^*(\mathbf{x}_i[t-1])\| \\ \leq& L_{i,t}\|\mathbf{x}_i[t] - \mathbf{x}_i[t-1]\| + \|\nabla_{\mathbf{x}^m} f_{i,t}^*(\mathbf{x}_i[t-1]) - \nabla_{\mathbf{x}^m} f_{i,t-1}^*(\mathbf{x}_i[t-1])\|. \end{aligned} \tag{A.5}$$

To reach the last line we utilize the triangle inequality and the Lipschitz continuity of $\nabla f_{i,t}^*$. For the first term,

$$\begin{aligned}
\|\mathbf{x}_i^m[t] - \mathbf{x}_i^m[t-1]\| \leq \|\mathbf{x}^m[t] - \mathbf{x}^m[t-1]\| &\leq \left\| \frac{\mathbf{1}_{\mathcal{N}_m}^\top \otimes \mathbf{I}_{d_m}}{n_m} (\mathbf{x}^m[t] - \mathbf{x}^m[t-1]) \right\| \\
&= \|\bar{\mathbf{x}}^m[t] - \bar{\mathbf{x}}^m[t-1]\| \qquad \text{(Fact A.4 (c))} \\
&= \alpha[t-1] \left\| \frac{\mathbf{1}_{\mathcal{N}_m}^\top \otimes \mathbf{I}_{d_m}}{n_m} \Delta \mathbf{x}^{m,inx}[t-1] \right\| \qquad \text{(Fact A.4 (e))} \\
&\leq c_1 \alpha[t-1] \|\Delta \mathbf{x}^{m,inx}[t-1]\|.
\end{aligned} \tag{A.6}$$

Using the induction hypothesis of $\Delta \mathbf{x}$, we obtain

$$\begin{aligned}
&\|\Delta \mathbf{r}_i^m[t, t-1]\| \leq L_{i,t} \|\mathbf{x}_i[t] - \mathbf{x}_i[t-1]\| + c_2 \eta_{i,t} \\
&\leq c_1 \alpha[t-1] L_{i,t} \|\Delta \mathbf{x}^{m,inx}[t-1]\| + c_2 \eta_{i,t} \leq \frac{c_3 \alpha[t-1] L_{i,t} L_{i,t-1}}{\tau_{i,t-1}} + c_2 \eta_{i,t}.
\end{aligned}$$

Therefore, we finally obtain

$$\begin{aligned}
\|\mathbf{y}_i^m[t]\| &\leq \left\| \hat{\mathbf{W}}^m[t-1] \mathbf{y}_i^m[t-1] \right\| + \|\Delta \mathbf{r}_i^m[t, t-1]\| \\
&\leq c_4 l_m L_{i,t-1} + \frac{c_3 \alpha[t-1] L_{i,t} L_{i,t-1}}{\tau_{i,t-1}} + c_2 \eta_{i,t} \\
&\leq L_{i,t} \left( c_4 l_m + \frac{c_3 \alpha[t-1] L_{i,t-1}}{\tau_{i,t-1}} + \frac{c_2 \eta_{i,t}}{L_{i,t}} \right) \leq c_5 L_{i,t}
\end{aligned}$$

using the induction hypothesis of $\Delta \mathbf{y}$ and the fact that $\frac{\alpha[t-1] L_{i,t-1}}{\tau_{i,t-1}}$ also goes to zero when $n \to \infty$ implied by the condition of Theorem 2.3.

### A.2.4 Proof of Proposition A.3 (b)

We only prove the case for $\mathbf{x}^c = \mathbf{x}^{M+1}$ to save the ubiquitous subscript of $m$. The proof of the claims for general $\mathbf{x}^m$ is exactly the same with appropriate substitutions of $\mathbf{x}^c, \hat{\mathbf{W}}, \mathbf{P}, \mathbf{r}^c, J, J_\perp, \mathbf{1}_n, I$ by $\mathbf{x}^m, \hat{\mathbf{W}}^m, \mathbf{P}^m, \mathbf{r}^m, J^m, J_\perp^m, \mathbf{1}_{\mathcal{N}_m}, n_m$.

(i)

$$\mathbf{x}^c[t] - \mathbf{1}_n \otimes \bar{\mathbf{x}}^c[t] = \mathbf{x}^c[t] - J\mathbf{x}^c[t] = J_\perp \mathbf{x}^c[t] \triangleq \mathbf{x}_\perp^c[t].$$

Notice that with Fact A.4 (d) and (e), the difference of $\mathbf{x}^c[t]$ and $\mathbf{1}_n \otimes \bar{\mathbf{x}}^c[t]$ which is $\mathbf{x}_\perp^c[t]$ can be expressed as a linear combination of $\mathbf{x}_\perp^c[t-1]$ and $\Delta \mathbf{x}^{c,inx}[t-1]$. We can thus expand $\mathbf{x}_{\perp}^{c}[t-1]$ iteratively as follows:

$$
\begin{aligned}
&\mathbf{x}_{\perp}^{c}[t] \\
&= J_{\perp}\hat{\mathbf{W}}[t-1]\mathbf{x}_{\perp}^{c}[t-1] + \alpha[t-1]J_{\perp}\hat{\mathbf{W}}[t-1]\Delta\mathbf{x}^{c,inx}[t-1] \\
&= J_{\perp}\hat{\mathbf{W}}[t-1](J_{\perp}\hat{\mathbf{W}}[t-2]\mathbf{x}_{\perp}^{c}[t-2] + \alpha[t-2]J_{\perp}\hat{\mathbf{W}}[t-2]\Delta\mathbf{x}^{c,inx}[t-2]) \\
&\qquad\qquad + \alpha[t-1]J_{\perp}\hat{\mathbf{W}}[t-1]\Delta\mathbf{x}^{c,inx}[t-1] \\
&\vdots \\
&= J_{\perp}\hat{\mathbf{W}}[t-1]J_{\perp}\hat{\mathbf{W}}[t-2]\cdots J_{\perp}\hat{\mathbf{W}}[0]\mathbf{x}_{\perp}^{c}[0] + \sum_{s=0}^{t-1} J_{\perp}\hat{\mathbf{W}}[t-1]\cdots J_{\perp}\hat{\mathbf{W}}[s]\alpha[s]\Delta\mathbf{x}^{c,inx}[s] \\
&= \left[\left(\mathbf{P}[t-1,0] - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\top}\right)\otimes n_m\right]\mathbf{x}_{\perp}^{c}[0] \\
&\qquad\qquad + \sum_{s=0}^{t-1}\left[\left(\mathbf{P}[t-1,s] - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\top}\right)\otimes n_m\right]\alpha[s]\Delta\mathbf{x}^{c,inx}[s]
\end{aligned}
\tag{A.7}
$$

where the last equation resulted from Fact A.4 (b). From Proposition A.3 (a) we know

$$
\|\Delta\mathbf{x}^{c,inx}[t]\| \leq \|\Delta\mathbf{x}^{inx}[t]\| = \sqrt{\sum_{i=1}^{n}\|\Delta\mathbf{x}_i^{inx}[t]\|^2} \leq c_1\max_{n}\|\Delta\mathbf{x}_i^{inx}[t]\| \leq \frac{c_2 L_t^{\max}}{\tau_t^{\min}} \tag{A.8}
$$

for some constants $c_1$ and $c_2$. Consequently, we get

$$
\|\mathbf{x}_{\perp}^{c}[t]\| \leq c_3\rho^t + c_4\sum_{s=0}^{t-1}\rho^{t-s}\frac{\alpha[s]L_s^{\max}}{\tau_s^{\min}} \xrightarrow{n\to\infty} 0 \tag{A.9}
$$

by first utilizing triangle inequality, and then using (A.8), Lemma A.2, and finally Lemma A.5 (a). Remark that $\lim_{t\to\infty}\alpha[t]\left(\frac{L_t^{\max}}{\tau_t^{\min}}\right)^3 = 0$ implies $\lim_{t\to\infty}\frac{\alpha[t]L_t^{\max}}{\tau_t^{\min}} = 0$, which we use in (A.9) as the condition of Lemma A.5 (a).

(ii)

$$\begin{aligned}
&\lim_{t\to\infty}\sum_{k=1}^{t}\alpha[k]\|\mathbf{x}_i^c[k]-\bar{\mathbf{x}}^c[k]\| \le \lim_{t\to\infty}\sum_{k=1}^{t}\alpha[k]\|\mathbf{x}_\perp^c[k]\| \\
&\le \lim_{t\to\infty}\sum_{k=1}^{t}\alpha[k]\left(c_3\rho^k + c_4\sum_{s=0}^{k-1}\rho^{k-s}\frac{\alpha[s]L_s^{\max}}{\tau_s^{\min}}\right) \qquad \text{(from (A.9))} \\
&\le \lim_{t\to\infty}\left(c_3\sum_{k=1}^{t}\rho^k\alpha[k] + c_4\rho\sum_{k=1}^{t}\sum_{s=1}^{k}\rho^{k-s}\alpha[k]\frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}}\right) < \infty.
\end{aligned}$$

The bound for the last term comes from Lemma A.5 (b).

(iii)

$$\begin{aligned}
\lim_{t\to\infty}\sum_{k=1}^{t}\|\mathbf{x}_\perp^c[k]\|^2 \le \lim_{t\to\infty}&\left(c_3^2\sum_{k=1}^{t}\rho^{2k} + 2c_3c_4\rho\sum_{k=1}^{t}\sum_{s=1}^{k}\rho^{2k-l}\frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}}\right. \\
&\left. + c_4^2\rho^2\sum_{k=1}^{t}\sum_{s=1}^{k}\sum_{t=1}^{k}\rho^{2k-l-t}\frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}}\frac{\alpha[t-1]L_{t-1}^{\max}}{\tau_{t-1}^{\min}}\right) < \infty.
\end{aligned}$$

The bound for the first term is natural. The double summation is bounded due to the second equality Lemma A.5 (b) with $(\lambda, \beta[k], \nu[s])$ being $(\rho, \rho^k, \frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}})$. The condition of Theorem 2.3 $\sum_{t=1}^{\infty}(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2 < \infty$ guarantees that $\sum_{t=1}^{\infty}\left(\frac{\alpha[t]L_t^{\max}}{\tau_t^{\min}}\right)^2 < \infty$. The inequality of the triple summation follows from

$$\begin{aligned}
&\lim_{t\to\infty}\sum_{k=1}^{t}\sum_{s=1}^{k}\sum_{t=1}^{k}\rho^{2k-l-t}\frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}}\frac{\alpha[t-1]L_{t-1}^{\max}}{\tau_{t-1}^{\min}} \\
&\le \lim_{t\to\infty}\sum_{k=1}^{t}\sum_{s=1}^{k}\sum_{t=1}^{k}\rho^{k-s}\cdot\rho^{k-t}\cdot\frac{\left(\frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}}\right)^2+\left(\frac{\alpha[t-1]L_{t-1}^{\max}}{\tau_{t-1}^{\min}}\right)^2}{2} \\
&\le \lim_{t\to\infty}\frac{1}{1-\rho}\sum_{k=1}^{t}\sum_{s=1}^{k}\rho^{k-s}\left(\frac{\alpha[s-1]L_{s-1}^{\max}}{\tau_{s-1}^{\min}}\right)^2 < \infty,
\end{aligned}$$

where the last inequality is due to the first equality of Lemma A.5 (b). Again, the convergence of $\sum_{t=1}^{\infty}\left(\frac{\alpha[t]L_t^{\max}}{\tau_t^{\min}}\right)^2$ is implied by the convergence of $\sum_{t=1}^{\infty}(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2$.

### A.2.5 Proof of Proposition A.3 (c)

We exploit the optimality of $\tilde{\mathbf{x}}_i^{av}$ and (F1′) and (A2) to get

$$\left[\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}) - \tilde{\mathbf{x}}_i^{av}\right]^\top \left[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{av};\bar{\mathbf{x}}^{\mathcal{S}_i}) + \tilde{\pi}_i^{av} + (\mathbf{0}^{\mathcal{S}_i\setminus\{c\}}, \partial G(\tilde{\mathbf{x}}_i^{c,av}))\right] \geq 0; \tag{A.10}$$

and the optimality of $\bar{\mathbf{x}}$ (for the mapping of $\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}})$) leads to

$$\left[\tilde{\mathbf{x}}_i^{av} - \hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}})\right]^\top \left[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}});\bar{\mathbf{x}}^{\mathcal{S}_i}) + \pi_i^{\mathcal{S}_i}(\bar{\mathbf{x}}) + (\mathbf{0}^{\mathcal{S}_i\setminus\{c\}}, \partial G(\hat{\mathbf{x}}_{i,t}^{c}(\bar{\mathbf{x}}))\right] \geq 0. \tag{A.11}$$

$\mathbf{0}^{\mathcal{S}_i\setminus\{c\}}$ is an all zero vector in the subspace $\mathcal{K}_{\mathcal{S}_i\setminus\{c\}}$. It should be clear that $\hat{\mathbf{x}}_{i,t}^{c}(\bar{\mathbf{x}})$ refers to the component of $\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}})$ in the subspace $\mathcal{K}_c$. Then

$$\begin{aligned}
&\tau_{i,t}\left\|\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}) - \tilde{\mathbf{x}}_i^{av}\right\|^2 \\
\leq{}& \left[\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}) - \tilde{\mathbf{x}}_i^{av}\right]^\top \cdot \Big[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{av};\bar{\mathbf{x}}^{\mathcal{S}_i}) - \nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}});\bar{\mathbf{x}}^{\mathcal{S}_i}) \\
&\qquad + \left(\mathbf{0}^{\mathcal{S}_i\setminus\{c\}}, \partial G(\hat{\mathbf{x}}_{i,t}^{c}(\bar{\mathbf{x}})) - \partial G(\tilde{\mathbf{x}}_i^{c,av}))\right)\Big] && \text{((F1′) and (A2))} \\
\leq{}& \left[\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}) - \tilde{\mathbf{x}}_i^{av}\right]^\top \cdot \left[\tilde{\pi}_i^{av} - \pi_i^{\mathcal{S}_i}(\bar{\mathbf{x}})\right] && \text{((A.10) and (A.11))} \\
\leq{}& \left\|\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}) - \tilde{\mathbf{x}}_i^{av}\right\| \cdot \left\|\tilde{\pi}_i^{av} - \pi_i^{\mathcal{S}_i}(\bar{\mathbf{x}})\right\|. && \text{(C-S inequality)}
\end{aligned} \tag{A.12}$$

From (A.12),

$$\begin{aligned}
\left\|\hat{\mathbf{x}}_{i,t}^{\mathcal{S}_i}(\bar{\mathbf{x}}) - \tilde{\mathbf{x}}_i^{av}\right\| &\leq \frac{1}{\tau_{i,t}}\left\|\tilde{\pi}_i^{av} - \pi_i^{\mathcal{S}_i}(\bar{\mathbf{x}})\right\| \\
= \frac{1}{\tau_{i,t}}&\left\|(n_m \mathbf{y}_i^{m,av})_{m\in\mathcal{S}_i} - \nabla_{\mathbf{x}^{\mathcal{S}_i}} f_{i,t}^*(\bar{\mathbf{x}}) - \sum_{j\neq i}\nabla_{\mathbf{x}^{\mathcal{S}_i}} f_{j,t}^*(\bar{\mathbf{x}})\right\| \\
\leq \frac{1}{\tau_{i,t}}&\sum_{m\in\mathcal{S}_i}\|n_m\mathbf{y}_i^{m,av} - n_m\bar{\mathbf{r}}^{m,av}\| \\
\leq \frac{1}{\tau_{i,t}}&\sum_{m\in\mathcal{S}_i} n_m\|\mathbf{y}^{m,av} - \mathbf{1}_{\mathcal{N}_m}\otimes\bar{\mathbf{r}}^{m,av}\|.
\end{aligned}$$

Up until now, the context is clear enough to allow us to drop all $[t]$ time index. Again, we only focus on the case of $m = M+1$; that is, proving $\frac{1}{\tau_{i,t}}\|\mathbf{y}^{c,av} - \mathbf{1}_n \otimes \bar{\mathbf{r}}^{c,av}\|$ goes to zero. We calculate

$$\begin{aligned}
\mathbf{y}^{c,av}[t] &= \hat{\mathbf{W}}[t-1]\mathbf{y}^{c,av}[t-1] + \Delta\mathbf{r}^{c,av}[t,t-1] \\
&= \hat{\mathbf{W}}[t-1](\hat{\mathbf{W}}[t-2]\mathbf{y}^{c,av}[t-2] + \Delta\mathbf{r}^{c,av}[t-1,t-2]) + \Delta\mathbf{r}^{c,av}[t,t-1] \\
&= \hat{\mathbf{P}}[t-1,t-2](\hat{\mathbf{W}}[t-3]\mathbf{y}^{c,av}[t-3] + \Delta\mathbf{r}^{c,av}[t-2,t-3]) \\
&\qquad + \hat{\mathbf{P}}[t-1,t-1]\Delta\mathbf{r}^{c,av}[t-1,t-2] + \Delta\mathbf{r}^{c,av}[t,t-1] \\
&\vdots \\
&= \hat{\mathbf{P}}[t-1,0]\mathbf{r}^{c,av}[0] + \sum_{s=1}^{t-1}\hat{\mathbf{P}}[t-1,s]\Delta\mathbf{r}^{c,av}[s,s-1] + \Delta\mathbf{r}^{c,av}[t,t-1],
\end{aligned} \tag{A.13}$$

and

$$\begin{aligned}
\mathbf{1}_n \otimes \bar{\mathbf{r}}^{c,av}[t] &= \mathbf{1}_n \otimes \left(\bar{\mathbf{r}}^{c,av}[0] + \sum_{s=1}^{n}\Delta\bar{\mathbf{r}}^{c,av}[s,s-1]\right) \\
&= \mathbf{1}_n \cdot \frac{\mathbf{1}_n^\top \otimes I_d}{n}\left(\mathbf{r}^{c,av}[0] + \sum_{s=1}^{n}\Delta\mathbf{r}^{c,av}[s,s-1]\right) \\
&= J\mathbf{r}^{c,av}[0] + \sum_{s=1}^{t-1}J\Delta\mathbf{r}^{c,av}[s,s-1] + J\Delta\mathbf{r}^{c,av}[t,t-1].
\end{aligned} \tag{A.14}$$

Similar to (A.5) we have

$$\begin{aligned}
&\|\Delta\mathbf{r}^{c,av}[s,s-1]\| \\
={}& \sum_{i\in\mathcal{N}}\|\nabla_{\mathbf{x}^c}f^*_{i,s}(\bar{\mathbf{x}}[s]) - \nabla_{\mathbf{x}^c}f^*_{i,s-1}(\bar{\mathbf{x}}[s-1])\| \\
={}& \sum_{i\in\mathcal{N}}\|\nabla_{\mathbf{x}^c}f^*_{i,s}(\bar{\mathbf{x}}[s]) - \nabla_{\mathbf{x}^c}f^*_{i,s-1}(\bar{\mathbf{x}}[s]) + \nabla_{\mathbf{x}^c}f^*_{i,s-1}(\bar{\mathbf{x}}[s]) - \nabla_{\mathbf{x}^c}f^*_{i,s-1}(\bar{\mathbf{x}}[s-1])\| \\
\le{}& nL^{\max}_{s-1}\|\bar{\mathbf{x}}[s] - \bar{\mathbf{x}}[s-1]\| + \sum_{i\in\mathcal{N}}\|\nabla_{\mathbf{x}^c}f^*_{i,s}(\bar{\mathbf{x}}[s]) - \nabla_{\mathbf{x}^c}f^*_{i,s-1}(\bar{\mathbf{x}}[s])\|.
\end{aligned} \tag{A.15}$$

Similar to the technique as in (A.7) to (A.9), by combining (A.13), (A.14), plus Lemma A.2, and then (A.15), we have

$$
\begin{aligned}
&\frac{1}{\tau_{i,t}}\|\mathbf{y}^{c,av}[t]-\mathbf{1}_n\otimes\bar{\mathbf{r}}^{c,av}[t]\| \\
&\le c_1\frac{\rho^t}{\tau_t^{\min}}+c_2\sum_{s=1}^{t-1}\frac{\rho^{t-s}}{\tau_t^{\min}}\|\Delta\mathbf{r}^{c,av}[s,s-1]\|+c_3\frac{1}{\tau_t^{\min}}\|\Delta\mathbf{r}^{c,av}[t,t-1]\| \\
&\le c_1\frac{\rho^t}{\tau_t^{\min}}+c_2\sum_{s=1}^{t-1}\frac{\rho^{t-s}}{\tau_t^{\min}}\left(nL_{s-1}^{\max}\|\bar{\mathbf{x}}[s]-\bar{\mathbf{x}}[s-1]\|+\sum_i\|\nabla_{\mathbf{x}^c}f_{i,s}^*(\bar{\mathbf{x}}[s])-\nabla_{\mathbf{x}^c}f_{i,s-1}^*(\bar{\mathbf{x}}[s])\|\right) \\
&\qquad+c_3\frac{1}{\tau_t^{\min}}\left(nL_{t-1}^{\max}\|\bar{\mathbf{x}}[t]-\bar{\mathbf{x}}[t-1]\|+\sum_i\|\nabla_{\mathbf{x}^c}f_{i,t}^*(\bar{\mathbf{x}}[t])-\nabla_{\mathbf{x}^c}f_{i,t-1}^*(\bar{\mathbf{x}}[t])\|\right) \\
&=c_1\frac{\rho^t}{\tau_t^{\min}}+c_2\sum_{s=1}^{t-1}\frac{\rho^{t-s}\alpha[s-1]L_{s-1}^{\max}}{\tau_t^{\min}}\left\|\left(\frac{1}{n_m}\mathbf{1}_{n_m}^\top\otimes I_{d_m}\Delta\mathbf{x}^{m,inx}[s-1]\right)_{m\in[M+1]}\right\| \\
&\qquad+c_3\frac{\alpha[t-1]L_{t-1}^{\max}}{\tau_t^{\min}}\left\|\left(\frac{1}{n_m}\mathbf{1}_{n_m}^\top\otimes I_{d_m}\Delta\mathbf{x}^{m,inx}[t-1]\right)_{m\in[M+1]}\right\| \\
&\qquad+c_2\sum_{s=1}^{t-1}\frac{\rho^{t-s}}{\tau_t^{\min}}\sum_i\|\nabla_{\mathbf{x}^c}f_{i,s}^*(\bar{\mathbf{x}}[s])-\nabla_{\mathbf{x}^c}f_{i,s-1}^*(\bar{\mathbf{x}}[s])\| \\
&\qquad+c_3\frac{1}{\tau_t^{\min}}\sum_i\|\nabla_{\mathbf{x}^c}f_{i,t}^*(\bar{\mathbf{x}}[t])-\nabla_{\mathbf{x}^c}f_{i,t-1}^*(\bar{\mathbf{x}}[t])\| \\
&\le c_1\frac{\rho^t}{\tau_t^{\min}}+c_4\sum_{s=1}^{t-1}\rho^{t-s}\frac{\alpha[s-1](L_{s-1}^{\max})^2}{\tau_t^{\min}\tau_{s-1}^{\min}}+c_5\frac{\alpha[t-1](L_{t-1}^{\max})^2}{\tau_t^{\min}\tau_{t-1}^{\min}} \\
&\qquad+c_2\sum_{s=1}^{t-1}\frac{\rho^{t-s}}{\tau_t^{\min}}\sum_i\|\nabla_{\mathbf{x}^c}f_{i,s}^*(\bar{\mathbf{x}}[s])-\nabla_{\mathbf{x}^c}f_{i,s-1}^*(\bar{\mathbf{x}}[s])\| \\
&\qquad+c_3\frac{1}{\tau_t^{\min}}\sum_i\|\nabla_{\mathbf{x}^c}f_{i,t}^*(\bar{\mathbf{x}}[t])-\nabla_{\mathbf{x}^c}f_{i,t-1}^*(\bar{\mathbf{x}}[t])\| \qquad \text{(Proposition A.3 (a))} \\
&\xrightarrow{n\to\infty} 0. \qquad \text{((T1), (T2), and (T3))}
\end{aligned}
\tag{A.16}
$$

In the last equation, we also have $\lim_{t\to\infty}\frac{\alpha[t](L_t^{\max})^2}{(\tau_t^{\min})^2}=0$ and $\lim_{t\to\infty}\frac{1}{\tau_t^{\min}}\|\nabla f_{i,t}^*(\bar{\mathbf{x}}[t])-\nabla f_{i,t-1}^*(\bar{\mathbf{x}}[t])\|=0$ implied by the conditions of Theorem 2.3. For the second part of the claim, we can equivalently prove

$$\sum_{t=1}^{\infty}\frac{\alpha[t]L_t^{\max}}{\tau_t^{\min}}\|\mathbf{y}^{c,av}[t]-\mathbf{1}_n\otimes\bar{\mathbf{r}}^{c,av}[t]\|<\infty.$$

This is true because

$$
\begin{aligned}
&\sum_{t=1}^{\infty} \frac{\alpha[t] L_t^{\max}}{\tau_t^{\min}} \|\mathbf{y}^{c,av}[t] - \mathbf{1}_n \otimes \bar{\mathbf{r}}^{c,av}[t]\| \\
&\le c_1 \sum_{t=1}^{\infty} \rho^t \frac{\alpha[t] L_t^{\max}}{\tau_t^{\min}} + c_4 \sum_{t=1}^{\infty} \alpha[t] L_t^{\max} \sum_{s=1}^{t-1} \rho^{t-s} \frac{\alpha[s-1](L_{s-1}^{\max})^2}{\tau_t^{\min}\tau_{s-1}^{\min}} \\
&\quad + c_5 \sum_{t=1}^{\infty} \frac{\alpha[t]\alpha[t-1] L_t^{\max} (L_{t-1}^{\max})^2}{\tau_t^{\min}\tau_{t-1}^{\min}} \\
&\quad + c_2 \sum_{t=1}^{\infty} \alpha[t] L_t^{\max} \sum_{s=1}^{t-1} \frac{\rho^{t-s}}{\tau_t^{\min}} \sum_i \|\nabla_{\mathbf{x}^c} f_{i,s}^*(\bar{\mathbf{x}}[s]) - \nabla_{\mathbf{x}^c} f_{i,s-1}^*(\bar{\mathbf{x}}[s])\| \\
&\quad + c_3 \sum_{t=1}^{\infty} \frac{\alpha[t] L_t^{\max}}{\tau_t^{\min}} \sum_i \|\nabla_{\mathbf{x}^c} f_{i,t}^*(\bar{\mathbf{x}}[t]) - \nabla_{\mathbf{x}^c} f_{i,t-1}^*(\bar{\mathbf{x}}[t])\| < \infty.
\end{aligned}
$$

All the terms are finite because of the following. The first term is due to (T4) – after multiplying a going-to-zero $\alpha[t]$, the term remains to be bounded. The second term is due to (T5). The third term is in the condition of Theorem 2.3. The fourth term is due to (T6). The last term is also in the condition of Theorem 2.3.

### A.2.6 Proof of Proposition A.3 (d)

Recall we have

$$\tilde{\mathbf{x}}_i[t] = \operatorname*{arg\,min}_{\mathbf{x}_i \in \mathcal{K}_{\mathcal{S}_i}} \ \tilde{U}_{i,t}(\mathbf{x}_i; \mathbf{x}_i[t], \tilde{\pi}_i[t])$$

and

$$\tilde{\mathbf{x}}_i^{av}[t] = \operatorname*{arg\,min}_{\mathbf{x}_i \in \mathcal{K}_{\mathcal{S}_i}} \ \tilde{U}_{i,t}(\mathbf{x}_i; \bar{\mathbf{x}}_i[t], \tilde{\pi}_i^{av}[t]),$$

where

$$\tilde{U}_{i,t}(\mathbf{x}_i; \mathbf{x}_i[t], \tilde{\pi}_i[t]) = \tilde{f}_{i,t}^*(\mathbf{x}_i; \mathbf{x}_i[t]) + \sum_{k \in \mathcal{S}_i} \tilde{\pi}_i^k[t]^\top (\mathbf{x}_i^k - \mathbf{x}_i^k[t]) + G(\mathbf{x}^c).$$

These along with (F1′) and (A2) lead to the following:

$$(\tilde{\mathbf{x}}_i^{av} - \tilde{\mathbf{x}}_i)^\top \cdot \left[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i; \mathbf{x}_i) + \tilde{\pi}_i + (\mathbf{0}^{\mathcal{S}_i \setminus \{c\}}, \partial G(\tilde{\mathbf{x}}_i^c))\right] \ge 0 \qquad \text{(A.17)}$$

and

$$(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{av})^\top \cdot \left[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{av}; \bar{\mathbf{x}}_i) + \tilde{\pi}_i^{av} + (\mathbf{0}^{\mathcal{S}_i \setminus \{c\}}, \partial G(\tilde{\mathbf{x}}_i^{c,av}))\right] \geq 0. \tag{A.18}$$

As we did in (A.12),

$$\begin{aligned}
&\tau_{i,t} \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{av}\|^2 \\
&\leq (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{av})^\top \cdot \left[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i; \mathbf{x}_i) - \nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{av}; \mathbf{x}_i) + (\mathbf{0}^{\mathcal{S}_i \setminus \{c\}}, \partial G(\tilde{\mathbf{x}}_i^{c}) - \partial G(\tilde{\mathbf{x}}_i^{c,av})\right] \\
&\hfill \text{((A2) and (F1}'\text{))} \\
&\leq (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{av})^\top \cdot \left[\nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{av}; \bar{\mathbf{x}}_i) - \nabla_{\mathbf{x}^{\mathcal{S}_i}} \tilde{f}_{i,t}^*(\tilde{\mathbf{x}}_i^{av}; \mathbf{x}_i) + \tilde{\pi}_i^{av} - \tilde{\pi}_i\right] \\
&\hfill \text{((A.17) and (A.18))} \\
&\leq \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{av}\| \cdot \left[L_{i,t} \|\bar{\mathbf{x}}_i - \mathbf{x}_i\| + \left\|(n_m(\mathbf{y}_i^{m,av} - \mathbf{y}_i^m))_{m \in \mathcal{S}_i} - \nabla_{\mathbf{x}^{\mathcal{S}_i}} f_{i,t}^*(\bar{\mathbf{x}}_i) - \nabla_{\mathbf{x}^{\mathcal{S}_i}} f_{i,t}^*(\mathbf{x}_i)\right\|\right] \\
&\hfill \text{((N1))} \\
&\leq \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{av}\| \cdot \left[2L_{i,t} \|\bar{\mathbf{x}}_i - \mathbf{x}_i\| + \sum_{m \in \mathcal{S}_i} \|n_m(\mathbf{y}_i^m - \mathbf{y}_i^{m,av})\|\right].
\end{aligned}$$

Hence,

$$\begin{aligned}
&\left[\sum_{m \in \mathcal{S}_i} \|\tilde{\mathbf{x}}_i^m - \tilde{\mathbf{x}}_i^{m,av}\|^2\right]^{1/2} \\
&\leq \frac{2L_{i,t}}{\tau_{i,t}} \left(\sum_{m \in \mathcal{S}_i} \|\bar{\mathbf{x}}^m - \mathbf{x}_i^m\|\right) + \sum_{m \in \mathcal{S}_i} \frac{n_m}{\tau_{i,t}} \|\mathbf{y}^m - \mathbf{y}^{m,av}\| \\
&\leq \frac{2L_{i,t}}{\tau_{i,t}} \left(\sum_{m \in \mathcal{S}_i} \|\bar{\mathbf{x}}^m - \mathbf{x}_i^m\|\right) + \sum_{m \in \mathcal{S}_i} \frac{n_m}{\tau_{i,t}} \big(\|\mathbf{1}_{\mathcal{N}_m} \otimes (\bar{\mathbf{r}}^m - \bar{\mathbf{r}}^{m,av})\| \\
&\qquad + \|\mathbf{y}^m - \mathbf{y}^{m,av} - \mathbf{1}_{\mathcal{N}_m} \otimes (\bar{\mathbf{r}}^m - \bar{\mathbf{r}}^{m,av})\|\big).
\end{aligned} \tag{A.19}$$

Since $\|\tilde{\mathbf{x}}_i^m - \tilde{\mathbf{x}}_i^{m,av}\|$ is not larger than $\left[\sum_{m \in \mathcal{S}_i} \|\tilde{\mathbf{x}}_i^m - \tilde{\mathbf{x}}_i^{m,av}\|^2\right]^{1/2}$, (A.19) implies the former goes to zero as $n$ goes to infinity if we can show all terms in the RHS do so. The first term does go to zero as we showed in part (b) (combining (A.9), Lemma A.5 (a), and the fact that $\lim_{t \to \infty} \alpha[t] \left(\frac{L_t^{\max}}{\tau_t^{\min}}\right)^2$). The following shows this property holds for the remaining two terms as well. As always we omit all time index $[t]$ from above as the context is clear enough.

We have

$$
\begin{aligned}
&\frac{1}{\tau_{i,t}}\|\mathbf{1}_{\mathcal{N}_m}\otimes(\bar{\mathbf{r}}^m[t]-\bar{\mathbf{r}}^{m,av}[t])\| \\
&\le \frac{1}{\tau_{i,t}}\sum_{i\in\mathcal{N}_m}\left\|\nabla_{\mathbf{x}^m} f^*_{i,t}(\mathbf{x}_i[t])-\nabla_{\mathbf{x}^m} f^*_{i,t}(\bar{\mathbf{x}}^{\mathcal{S}_i}[t])\right\| \\
&\le \sum_{i\in\mathcal{N}_m}\frac{L_{i,t}}{\tau_{i,t}}\|\mathbf{x}_i[t]-\bar{\mathbf{x}}^{\mathcal{S}_i}[t]\| && \text{((N1))} \\
&\xrightarrow{n\to\infty} 0, && \text{(Proposition A.3 (b))}
\end{aligned}
$$

and

$$
\begin{aligned}
&\frac{1}{\tau_{i,t}}\|\mathbf{y}^m-\mathbf{y}^{m,av}-\mathbf{1}_{\mathcal{N}_m}\otimes(\bar{\mathbf{r}}^m-\bar{\mathbf{r}}^{m,av})\| \\
&\le c_1\frac{\rho^t}{\tau_t^{\min}}+c_2\sum_{s=1}^{t-1}\frac{\rho^{t-s}}{\tau_t^{\min}}\|\Delta\mathbf{r}^m[s,s-1]-\Delta\mathbf{r}^{m,av}[s,s-1]\| \\
&\qquad +c_3\frac{1}{\tau_t^{\min}}\|\Delta\mathbf{r}^m[t,t-1]-\Delta\mathbf{r}^{m,av}[t,t-1]\| && \text{((A.13), (A.14), and (A.16))} \\
&\le c_1\frac{\rho^t}{\tau_t^{\min}}+c_4\sum_{s=1}^{t-1}\frac{\rho^{t-s}}{\tau_t^{\min}}\sum_{i\in\mathcal{N}_m}\big(L_s^{\max}\|\mathbf{x}_i^m[s]-\bar{\mathbf{x}}^m[s]\|+L_{s-1}^{\max}\|\mathbf{x}_i^m[s-1]-\bar{\mathbf{x}}^m[s-1]\| \\
&\qquad +\|\nabla_{\mathbf{x}^m} f^*_{i,s}(\mathbf{x}_i[s-1])-\nabla_{\mathbf{x}^m} f^*_{i,s-1}(\mathbf{x}_i[s-1])\| \\
&\qquad +\|\nabla_{\mathbf{x}^m} f^*_{i,s}(\bar{\mathbf{x}}[s-1])-\nabla_{\mathbf{x}^m} f^*_{i,s-1}(\bar{\mathbf{x}}[s-1])\|\big) \\
&\qquad +c_5\frac{1}{\tau_t^{\min}}\sum_{i\in\mathcal{N}_m}\big(L_t^{\max}\|\mathbf{x}_i^m[t]-\bar{\mathbf{x}}^m[t]\|+L_{t-1}^{\max}\|\mathbf{x}_i^m[t-1]-\bar{\mathbf{x}}^m[t-1]\| \\
&\qquad +\|\nabla_{\mathbf{x}^m} f^*_{i,s}(\mathbf{x}_i[t-1])-\nabla_{\mathbf{x}^m} f^*_{i,s-1}(\mathbf{x}_i[t-1])\| \\
&\qquad +\|\nabla_{\mathbf{x}^m} f^*_{i,s}(\bar{\mathbf{x}}[t-1])-\nabla_{\mathbf{x}^m} f^*_{i,s-1}(\bar{\mathbf{x}}[t-1])\|\big) && \text{((N1))} \\
&\xrightarrow{n\to\infty} 0. && \text{((T1), (T2), Proposition A.3 (b), Lemma A.5 (a), and (T3))}
\end{aligned}
$$

For the terms of the form $\frac{L_t}{\tau_t}\|\mathbf{x}_\perp[t]\|$ to converge to zero, refer to (A.9) and Assumption T1 and T2. In the second line, from (A.13), (A.14), and (A.16) we know that $\|\mathbf{y}^{m,av}-\mathbf{1}_{\mathcal{N}_m}\otimes\bar{\mathbf{r}}^{m,av}\|$ can be represented as a sum of $\Delta\mathbf{r}^{m,av}[s,s-1]$'s; using the same method $\|\mathbf{y}^m-\mathbf{1}_{\mathcal{N}_m}\otimes\bar{\mathbf{r}}^m\|$ can also be represented as a sum of $\Delta\mathbf{r}^m[s,s-1]$'s, which we omit here. In the last inequality one can alternatively use (A.6) to bound $\Delta\mathbf{r}^m[s,s-1]$ and $\Delta\mathbf{r}^{m,av}[s,s-1]$, which is simpler and sufficient for our purposes.

For the second part of the claim,

$$
\begin{aligned}
&\sum_{t=1}^{\infty} \alpha[t] L_t^{\max} \left\| \tilde{\mathbf{x}}_i^m[t] - \tilde{\mathbf{x}}_i^{m,av}[t] \right\| \\
&\le c_6 \sum_{t=1}^{\infty} \frac{\alpha[t](L_t^{\max})^2}{\tau_t^{\min}} \|\mathbf{x}_\perp^{\mathcal{S}_i}[t]\| + c_1 \sum_{t=1}^{\infty} \frac{\rho^t \alpha[t] L_t^{\max}}{\tau_t^{\min}} \\
&\quad + c_4 \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \frac{\rho^{t-s}\alpha[t] L_t^{\max}}{\tau_t^{\min}} \sum_{i \in \mathcal{N}_m} \Big( L_s^{\max} \|\mathbf{x}_i^m[s] - \bar{\mathbf{x}}^m[s]\| + L_{s-1}^{\max} \|\mathbf{x}_i^m[s-1] - \bar{\mathbf{x}}^m[s-1]\| \\
&\qquad\qquad + \|\nabla_{\mathbf{x}^m} f_{i,s}^*(\mathbf{x}_i[s-1]) - \nabla_{\mathbf{x}^m} f_{i,s-1}^*(\mathbf{x}_i[s-1])\| \\
&\qquad\qquad + \|\nabla_{\mathbf{x}^m} f_{i,s}^*(\bar{\mathbf{x}}[s-1]) - \nabla_{\mathbf{x}^m} f_{i,s-1}^*(\bar{\mathbf{x}}[s-1])\| \Big) \\
&\quad + c_5 \sum_{t=1}^{\infty} \frac{\alpha[t] L_t^{\max}}{\tau_t^{\min}} \sum_{i \in \mathcal{N}_m} \Big( L_t^{\max} \|\mathbf{x}_i^m[t] - \bar{\mathbf{x}}^m[t]\| + L_{t-1}^{\max} \|\mathbf{x}_i^m[t-1] - \bar{\mathbf{x}}^m[t-1]\| \\
&\qquad\qquad + \|\nabla_{\mathbf{x}^m} f_{i,s}^*(\mathbf{x}_i[t-1]) - \nabla_{\mathbf{x}^m} f_{i,s-1}^*(\mathbf{x}_i[t-1])\| \\
&\qquad\qquad + \|\nabla_{\mathbf{x}^m} f_{i,s}^*(\bar{\mathbf{x}}[t-1]) - \nabla_{\mathbf{x}^m} f_{i,s-1}^*(\bar{\mathbf{x}}[t-1])\| \Big) \\
&< \infty.
\end{aligned}
$$

For the first term, use (A.9), (T4), and (T5). Second term is finite due to (T4) with additional $\alpha[t]$. The terms in the forth line are just like the first term. The terms in the fifth line converge by the condition of Theorem 2.3. The terms in the second line are of the type $\sum_t \frac{\alpha[t]L_t}{\tau_t} \sum_s \rho^{t-s} L_s \|\mathbf{x}_\perp[s]\|$, from (A.9) and (T4) one can show that $\sum_t \frac{\alpha[t]L_t^2}{\tau_t} \|\mathbf{x}_\perp[t]\|$ converges, hence the convergence of the terms by applying second part of Lemma A.7. The terms in the third line converge because of (T6).

### A.2.7 Proof of Theorem 2.3

Denote $F_t^* = \sum_{i\in\mathcal{N}} f_{i,t}^*$. By descent Lemma,

$$
\begin{aligned}
&F_t^*(\bar{\mathbf{x}}[t+1]) \\
&\le F_t^*(\bar{\mathbf{x}}[t]) + \nabla F_t^*(\bar{\mathbf{x}}[t])^\top(\bar{\mathbf{x}}[t+1] - \bar{\mathbf{x}}[t]) + \frac{L_t^{\max}}{2}\|\bar{\mathbf{x}}[t+1] - \bar{\mathbf{x}}[t]\|^2 \\
&= F_t^*(\bar{\mathbf{x}}[t]) + \sum_m \left[\nabla_{\mathbf{x}^m} F_t^*(\bar{\mathbf{x}}[t])^\top(\bar{\mathbf{x}}^m[t+1] - \bar{\mathbf{x}}^m[t]) + \frac{L_t^{\max}}{2}\|\bar{\mathbf{x}}^m[t+1] - \bar{\mathbf{x}}^m[t]\|^2\right] \\
&= F_t^*(\bar{\mathbf{x}}[t]) + \sum_m \left[\frac{\alpha[t]}{n_m}\nabla_{\mathbf{x}^m} F_t^*(\bar{\mathbf{x}}[t])^\top \sum_{i\in\mathcal{N}_m}(\mathbf{x}_i^{m,inx}[t] - \bar{\mathbf{x}}_i^m[t]) + \frac{L_t^{\max}}{2}\|\bar{\mathbf{x}}^m[t+1] - \bar{\mathbf{x}}^m[t]\|^2\right] \\
&\le F_t^*(\bar{\mathbf{x}}[t]) + \sum_m \left[\frac{\alpha[t]}{n_m}\nabla_{\mathbf{x}^m} F_t^*(\bar{\mathbf{x}}[t])^\top \sum_{i\in\mathcal{N}_m}\left[\left(\hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t]) - \bar{\mathbf{x}}_i^m[t]\right) + \left(\tilde{\mathbf{x}}_i^{m,av}[t] - \hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t])\right)\right.\right. \\
&\qquad\left.\left. + (\tilde{\mathbf{x}}_i^m[t] - \tilde{\mathbf{x}}_i^{m,av}[t]) + \left(\mathbf{x}_i^{m,inx}[t] - \tilde{\mathbf{x}}_i^m[t]\right)\right] + \frac{L_t^{\max}}{2}\|\bar{\mathbf{x}}^m[t+1] - \bar{\mathbf{x}}^m[t]\|^2\right]
\end{aligned}
\tag{A.20}
$$

By the convexity of $G$ (A2),

$$
\begin{aligned}
G(\bar{\mathbf{x}}^c[t+1]) &\le (1-\alpha[t])G(\bar{\mathbf{x}}^c[t]) + \alpha[t]G\left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^{c,inx}[t]\right) \\
&\le (1-\alpha[t])G(\bar{\mathbf{x}}^c[t]) + \frac{\alpha[t]}{n}\sum_{i=1}^n G(\mathbf{x}_i^{c,inx}[t]).
\end{aligned}
\tag{A.21}
$$

Then using Proposition A.1 (a) and the fact that $G$ has bounded subgradients,

$$
\begin{aligned}
&\sum_m \frac{\alpha[t]}{n_m}\nabla_{\mathbf{x}^m} F_t^*(\bar{\mathbf{x}}[t])^\top \sum_{i\in\mathcal{N}_m}\left(\hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t]) - \bar{\mathbf{x}}_i^m[t]\right) \\
&\le -\tau_t^{\min}\alpha[t]\diamondsuit[t] + \alpha[t]\left[G(\bar{\mathbf{x}}^c[t]) - \frac{1}{n}\sum_{i=1}^n G\left(\hat{\mathbf{x}}_{i,t}^c(\bar{\mathbf{x}}[t])\right)\right] \qquad \text{(Proposition A.1 (a))} \\
&\le -\tau_t^{\min}\alpha[t]\diamondsuit[t] + G(\bar{\mathbf{x}}^c[t]) - G(\bar{\mathbf{x}}^c[t+1]) + \frac{\alpha[t]}{n}\sum_{i=1}^n \left\|G(\mathbf{x}_i^{c,inx}[t]) - G\left(\hat{\mathbf{x}}_{i,t}^c(\bar{\mathbf{x}}[t])\right)\right\| \\
&\qquad \text{(by (A.21))} \\
&\le -\tau_t^{\min}\alpha[t]\diamondsuit[t] + G(\bar{\mathbf{x}}^c[t]) - G(\bar{\mathbf{x}}^c[t+1]) + \frac{L_G\alpha[t]}{n}\sum_{i=1}^n \left\|\mathbf{x}_i^{c,inx}[t] - \hat{\mathbf{x}}_{i,t}^c(\bar{\mathbf{x}}[t])\right\| \\
&\qquad \text{((A2))}
\end{aligned}
\tag{A.22}
$$

where $\diamondsuit[t]$ stands for the expression $\sum_m \sum_{i\in\mathcal{N}_m} \|\hat{\mathbf{x}}_i^m(\bar{\mathbf{x}}[t]) - \bar{\mathbf{x}}_i^m[t]\|^2$. Combining (A.20), (A.22) and (N1) with Cauchy-Schwarz inequality as well as triangle inequality, we get

$$\begin{aligned}
&F_t^*(\bar{\mathbf{x}}[t+1]) \\
&\leq F_t^*(\bar{\mathbf{x}}[t]) + G(\bar{\mathbf{x}}^c[t]) - G(\bar{\mathbf{x}}^c[t+1]) + \frac{\alpha[t]L_G}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i^{c,inx}[t] - \hat{\mathbf{x}}_{i,t}^c(\bar{\mathbf{x}}^c[t])\right\| \\
&\quad + \sum_m \left[\frac{\alpha[t]L_t^{\max}}{n_m}\sum_{i\in\mathcal{N}_m}\left(\left\|\tilde{\mathbf{x}}_i^{m,av}[t] - \hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t])\right\| + \|\tilde{\mathbf{x}}_i^m[t] - \tilde{\mathbf{x}}_i^{m,av}[t]\| + \epsilon_i^m[t]\right)\right] \\
&\quad + \sum_m \frac{L_t^{\max}}{2}\|\bar{\mathbf{x}}^m[t+1] - \bar{\mathbf{x}}^m[t]\|^2 - \tau_t^{\min}\alpha[t]\diamondsuit[t].
\end{aligned} \tag{A.23}$$

From the triangle inequality, Proposition A.3 (a) and Fact A.4 (e),

$$\begin{aligned}
\left\|\mathbf{x}_i^{c,inx}[t] - \hat{\mathbf{x}}_{i,t}^c(\bar{\mathbf{x}}^c[t])\right\| &\leq \left\|\mathbf{x}_i^{c,inx}[t] - \tilde{\mathbf{x}}_i^c[t]\right\| + \|\tilde{\mathbf{x}}_i^c[t] - \tilde{\mathbf{x}}_i^{c,av}[t]\| \\
&\quad + \left\|\tilde{\mathbf{x}}_i^{c,av}[t] - \hat{\mathbf{x}}_{i,t}^c(\bar{\mathbf{x}}^c[t])\right\|, \\
\|\bar{\mathbf{x}}^m[t+1] - \bar{\mathbf{x}}^m[t]\|^2 &\leq \left(\frac{c^m\alpha[t]L_t^{\max}}{n_m\tau_t^{\min}}\right)^2 \quad \forall\ m.
\end{aligned}$$

Substitute these expression back into (A.23) and rearrange the terms to get

$$\begin{aligned}
&U_{n+1}^*(\bar{\mathbf{x}}[t+1]) \leq U_n^*(\bar{\mathbf{x}}[t]) - \tau_t^{\min}\alpha[t]\diamondsuit[t] + c_1(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2 \\
&\quad + F_{n+1}^*(\bar{\mathbf{x}}[t+1]) - F_t^*(\bar{\mathbf{x}}[t+1]) \\
&\quad + c_2\sum_m\left[\alpha[t]L_t^{\max}\sum_{i\in\mathcal{N}_m}\left(\left\|\tilde{\mathbf{x}}_i^{m,av}[t] - \hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t])\right\| + \|\tilde{\mathbf{x}}_i^m[t] - \tilde{\mathbf{x}}_i^{m,av}[t]\| + \epsilon_i^m[t]\right)\right].
\end{aligned} \tag{A.24}$$

We now exploit Lemma A.6 with $Y[t] = U_n^*(\bar{\mathbf{x}}[t])$, $X[t] = \tau_t^{\min}\alpha[t]\diamondsuit[t]$ and

$$\begin{aligned}
Z[t] &= c_1(L_t^{\max})^3\left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2 + F_{n+1}^*(\bar{\mathbf{x}}[t+1]) - F_t^*(\bar{\mathbf{x}}[t+1]) \\
&\quad + c_2\sum_m\left[\alpha[t]L_t^{\max}\sum_{i\in\mathcal{N}_m}\left(\left\|\tilde{\mathbf{x}}_i^{m,av}[t] - \hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t])\right\| + \|\tilde{\mathbf{x}}_i^m[t] - \tilde{\mathbf{x}}_i^{m,av}[t]\| + \epsilon_i^m[t]\right)\right].
\end{aligned}$$

Since $U(\bar{\mathbf{x}}[t])$ is coercive ((A3)), $Y[t] \not\to -\infty$; on the other hand, from Proposition A.3

(c), (d), and the assumption of the Theorem, $\sum_{t=1}^{\infty} Z[t] < \infty$. Thus, by Lemma A.6 $\{U_n^*(\bar{\mathbf{x}}[t])\}$ converges to a finite value and $\sum_{t=1}^{\infty} \tau_t^{\min}\alpha[t]\diamondsuit[t]$ converges as well, which means

$$\sum_{t=1}^{\infty} \tau_t^{\min}\alpha[t] \left\|\hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t]) - \bar{\mathbf{x}}^m[t]\right\|^2 < \infty \quad \forall\ i \in \mathcal{N}_m,\ \forall\ m.$$

This in turn implies

$$\lim_{t\to\infty} \left\|\hat{\mathbf{x}}_{i,t}^m(\bar{\mathbf{x}}[t]) - \bar{\mathbf{x}}^m[t]\right\| = 0 \quad \forall\ i \in \mathcal{N}_m,\ \forall\ m.$$

At this point the localization is no longer an issue, and we will use the generalized definition of $\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}[t]) \in \mathcal{K}$ so that we have $\lim_{t\to\infty} \|\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}[t]) - \bar{\mathbf{x}}[t]\| = 0$ for all $i \in \mathcal{N}$.

Since $\{\bar{\mathbf{x}}[t]\}$ is bounded following from the convergence of $\{U_n^*(\bar{\mathbf{x}}[t])\}$, there exists a limit point $\bar{\mathbf{x}}^\infty \in \mathcal{K}$ of the set. We assume $\bar{\mathbf{x}}[t] \to \bar{\mathbf{x}}^\infty$. If this is not the case, then one can find a subsequence $\bar{\mathbf{x}}[t_k]$ indexed by $k$ such that $\bar{\mathbf{x}}[t_k] \to \bar{\mathbf{x}}^\infty$ as $k \to \infty$. We consider a partition of three cases: (1) bounded gradient ($\exists\ B$ s.t. $\|\nabla f_i(\mathbf{x})\| < B\ \forall\ i, \mathbf{x}$), (2) unbounded gradient and interior point ($\bar{\mathbf{x}}^\infty \in int(\mathcal{K})$), and (3) unbounded gradient and boundary point ($\bar{\mathbf{x}}^\infty \in bd(\mathcal{K})$).

**(1) bounded gradient**: Recall the map defined in Proposition A.1

$$\hat{\mathbf{x}}_{i,t}(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}}\ \tilde{f}_{i,t}^*(\mathbf{x};\tilde{\mathbf{x}}) + \pi_i(\tilde{\mathbf{x}})^\top(\mathbf{x}-\tilde{\mathbf{x}}) + G(\mathbf{x}) \triangleq \arg\min_{\mathbf{x}}\ \tilde{U}_{i,t}(\mathbf{x};\tilde{\mathbf{x}}).$$

This map is converging to the following map

$$\hat{\mathbf{x}}_t(\tilde{\mathbf{x}}) = \arg\min_{\mathbf{x}}\ \tilde{f}_i(\mathbf{x};\tilde{\mathbf{x}}) + \pi_i(\tilde{\mathbf{x}})^\top(\mathbf{x}-\tilde{\mathbf{x}}) + G(\mathbf{x}) \triangleq \arg\min_{\mathbf{x}}\ \tilde{U}_t(\mathbf{x};\tilde{\mathbf{x}}), \tag{A.25}$$

which might be multi-valued since we do not require $\tilde{f}_i$ to be strongly convex. The latter map is well-defined everywhere only with bounded gradient. Otherwise $\pi_i$ could be infinite; moreover, when $\nabla f_i(\mathbf{x}) = \infty$ and $\mathbf{x} \in int(\mathcal{K})$, it is not possible to achieve $\nabla\tilde{f}_i(\mathbf{x};\mathbf{x}) = \nabla f_i(\mathbf{x})$, $\tilde{f}_i$ being defined everywhere and being convex simultaneously. Thus, the analysis for this case does not work for the other two cases.

Now consider the two maps evaluated at $\bar{\mathbf{x}}[t]$ and $\bar{\mathbf{x}}^\infty$ respectively, $\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}[t])$, the minimizer of $\tilde{U}_{i,t}(\bullet;\bar{\mathbf{x}}[t]) \triangleq \psi_t$, and $\hat{\mathbf{x}}_t(\bar{\mathbf{x}}^\infty)$, the set of minimizers of $\tilde{U}_t(\bullet;\bar{\mathbf{x}}^\infty) \triangleq \psi$. We have the following two properties.

- $\{\psi_t\}$ is eventually level-bounded, i.e. $\forall\ \alpha \in \mathbb{R}$, $\bigcup_{t\in N, N\in\mathcal{N}_\infty} lev_{\leq\alpha}\psi_t$ is bounded. Refer to [107], p. 8, p. 109, and p. 123 for the definitions of the notations. This is ensured by Assumption F3, i.e. either $\tilde{f}_i(\bullet;\mathbf{x})$ is coercive $\forall\ \mathbf{x}, i$ or $G(\bullet)$ is coercive.

- $\psi_t \overset{e}{\to} \psi$, i.e. $\psi_t$ epi-converges to $\psi$. See [107], p. 241 for the definition. This is due to $\{\tilde{U}_{i,t}\}$ and $\tilde{U}_i$ being continuous and $\lim_{t\to\infty} \tilde{U}_{i,t} = \tilde{U}_i$, then by [107] Theorem 7.2, p. 241 we have $\psi_t \overset{e}{\to} \psi$.

By [107] Theorem 7.33, p. 266, with these two properties, we then have

$$\bar{\mathbf{x}}^\infty = \lim_{t\to\infty} \hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}[t]) = \limsup_{t\to\infty}(\arg\min \psi_t) \subset \arg\min \psi = \hat{\mathbf{x}}_t(\bar{\mathbf{x}}^\infty).$$

In [80] Proposition 5(b) says that the fixed point of $\hat{\mathbf{x}}_t$ is also the stationary point of the original optimization problem, which is proved in [40] Proposition 8(b). Things change slightly here as the minimizer of $\hat{\mathbf{x}}_t$ may not be unique. However, in the proof they did not exploit any strong convexity property. Hence, we still have $\bar{\mathbf{x}}^\infty$ being a stationary point.

**(2) unbounded gradient and interior point**: Effectively we want to show

$$\nabla F(\bar{\mathbf{x}}^\infty)^\top(\mathbf{z} - \bar{\mathbf{x}}^\infty) + G(\mathbf{z}) - G(\bar{\mathbf{x}}^\infty) \geq 0 \quad \forall\ \mathbf{z} \in \mathcal{K}, \tag{A.26}$$

but we can no longer argue anything with $\hat{\mathbf{x}}_i$. Only for the following we will write $\bar{\mathbf{x}}_t$ instead of $\bar{\mathbf{x}}[t]$ for simplicity. From the optimality condition of $\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)$, we have that for all $\mathbf{z} \in \mathcal{K}$,

$$\begin{aligned} 0 &\leq \left[\nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\bar{\mathbf{x}}_t) + \sum_{j\neq i} \nabla f^*_{i,t}(\bar{\mathbf{x}}_t)\right]^\top (\mathbf{z} - \hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)) + G(z) - G\left(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)\right) \\ &= \left[\nabla \tilde{f}^*_{i,t}(\bar{\mathbf{x}}_t;\bar{\mathbf{x}}_t) + \sum_{j\neq i} \nabla f^*_{i,t}(\bar{\mathbf{x}}_t)\right]^\top (\mathbf{z} - \hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)) + G(z) - G\left(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)\right) \\ &\quad + \left[\nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\bar{\mathbf{x}}_t) - \nabla \tilde{f}^*_{i,t}(\bar{\mathbf{x}}_t;\bar{\mathbf{x}}_t)\right]^\top (\mathbf{z} - \hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)), \end{aligned} \tag{A.27}$$

where $\nabla \tilde{f}^*_{i,t}(\bar{\mathbf{x}}_t;\bar{\mathbf{x}}_t) + \sum_{j\neq i} \nabla f^*_{i,t}(\bar{\mathbf{x}}_t)$ is just $\nabla F^*_t(\bar{\mathbf{x}}_t)$. The terms in the second bracket are bounded as follows

$$\begin{aligned}
&\left\|\nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\bar{\mathbf{x}}_t) - \nabla \tilde{f}^*_{i,t}(\bar{\mathbf{x}}_t;\bar{\mathbf{x}}_t)\right\| \\
\leq &\left\|\nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\bar{\mathbf{x}}_t) - \nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t))\right\| \\
&\qquad\qquad + \left\|\nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)) - \nabla \tilde{f}^*_{i,t}(\bar{\mathbf{x}}_t;\bar{\mathbf{x}}_t)\right\| \\
= &\left\|\nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\bar{\mathbf{x}}_t) - \nabla \tilde{f}^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t);\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t))\right\| + \left\|\nabla f^*_{i,t}(\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t)) - \nabla f^*_{i,t}(\bar{\mathbf{x}}_t)\right\| \\
\leq &L_{i,t}\left\|\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t) - \bar{\mathbf{x}}_t\right\| + L_{i,t}\left\|\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t) - \bar{\mathbf{x}}_t\right\| \leq 2L_t^{\max}\left\|\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t) - \bar{\mathbf{x}}_t\right\|,
\end{aligned}$$

where the second inequality is due to the Lipschitz continuities of $\nabla \tilde{f}^*_{i,t}(\mathbf{x};\bullet)$ and $\nabla f^*_{i,t}(\bullet)$. Since we assume $\sum_n (L_t^{\max})^3 \left(\frac{\alpha[t]}{\tau_t^{\min}}\right)^2 < \infty$ in the condition and get $\sum_n \alpha[t]\tau_t^{\min}\|\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t) - \bar{\mathbf{x}}_t\|^2 < \infty$, it must be that $\|\hat{\mathbf{x}}_{i,t}(\bar{\mathbf{x}}_t) - \bar{\mathbf{x}}_t\|^2 = O\left(\alpha[t]\frac{(L_t^{\max})^3}{(\tau_t^{\min})^3}\right)$. Hence, with the conditions of $\lim_{t\to\infty}\alpha[t]\frac{(L_t^{\max})^5}{(\tau_t^{\min})^3} = 0$ and $\lim_{t\to\infty}\nabla F_t^* = \nabla F$, taking $n \to \infty$ in (A.27) yields exactly (A.26). It is evident that $\bar{\mathbf{x}}^\infty$ must be a point such that $\nabla F(\bar{\mathbf{x}}^\infty) < \infty$, because if not so $\bar{\mathbf{x}}^\infty$ is an interior point and there must exist one descent direction.

**(3) unbounded gradient and boundary point**: We can consider two subcases.

- $\nabla F(\bar{\mathbf{x}}^\infty) < \infty$: we can use the same argument in case (2) to show that $\bar{\mathbf{x}}^\infty$ is a stationary point. If we have $\|\nabla f_i(\bar{\mathbf{x}}^\infty)\| < B \ \forall\ i$, we can also use the same argument in case (1) confined to a small neighborhood of $\bar{\mathbf{x}}^\infty$.

- $\nabla F(\bar{\mathbf{x}}^\infty) = \infty$: the definition of stationary point fails here and we can only turn to the definition of local minimum. However, both NEXT and our algorithm can numerically converge to a point which is not a local minimum.

## A.3 Technical Lemmas

We put some technical lemmas used in the proof in this appendix. Some of them are from [80].

**Fact A.4:** *For all $m$ we have the following.*

*(a)* $J_{\perp}^{m}\hat{\mathbf{W}}^{m}[t] = J_{\perp}^{m}\hat{\mathbf{W}}^{m}[t]J_{\perp}^{m} = \hat{\mathbf{W}}^{m}[t] - \frac{1_m}{n}\mathbf{1}_{\mathcal{N}_m}\mathbf{1}_{\mathcal{N}_m}^{\top} \otimes I_{d_m}$, *where we also have*

$$\begin{aligned} J_{\perp}^{m}\hat{\mathbf{W}}^{m}[t] &= \hat{\mathbf{W}}^{m}[t] - \left(\hat{\mathbf{W}}^{m}[t] \cdot \frac{1}{n_m}\mathbf{1}_{\mathcal{N}_m}\mathbf{1}_{\mathcal{N}_m}^{\top}\right) \otimes \mathbf{I}_{d_m} \\ &= \left(\mathbf{I}_{d_m n_m} - \mathbf{1}_{\mathcal{N}_m}\mathbf{1}_{\mathcal{N}_m}^{\top} \otimes \mathbf{I}_{d_m}\right)\hat{\mathbf{W}}^{m}[t]\left(\mathbf{I}_{d_m n_m} - \mathbf{1}_{\mathcal{N}_m}\mathbf{1}_{\mathcal{N}_m}^{\top} \otimes \mathbf{I}_{d_m}\right) \\ &= J_{\perp}^{m}\hat{\mathbf{W}}^{m}[t]J_{\perp}^{m}. \end{aligned}$$

*We use the equality* $(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D)$ *in showing the above equation.*

*(b)* $J_{\perp}^{m}\hat{\mathbf{W}}^{m}[t]J_{\perp}^{m}\hat{\mathbf{W}}^{m}[t-1] \cdots J_{\perp}^{m}\hat{\mathbf{W}}^{m}[s] = J_{\perp}^{m}\hat{\mathbf{P}}^{m}[t,s] = \left(\mathbf{P}^{m}[t,s] - \frac{1}{n_m}\mathbf{1}_{\mathcal{N}_m}\mathbf{1}_{\mathcal{N}_m}^{\top}\right) \otimes \mathbf{I}_{d_m}$.

*(c)* $\bar{\mathbf{q}}^{m} \triangleq \frac{1}{n_m}\sum_{i\in\mathcal{N}_m} q_i = \frac{\mathbf{1}_{\mathcal{N}_m}^{\top} \otimes \mathbf{I}_{d_m}}{n_m}\mathbf{q}$ *where* $\mathbf{q} = [q_1^{\top} \ \cdots \ q_I^{\top}]^{\top}$ *and* $q_1, \ldots, q_I$ *are all arbitrary in* $\mathbb{R}^{d_m}$.

*(d)* $\mathbf{x}^{m}[t] = \hat{\mathbf{W}}^{m}[t-1]\mathbf{x}^{m}[t-1] + \alpha[t-1]\hat{\mathbf{W}}^{m}[t-1]\Delta\mathbf{x}^{m,inx}[t-1]$ *where* $\Delta\mathbf{x}^{m,inx}[t] = \left(\mathbb{I}\{i \in \mathcal{N}_m\}(\mathbf{x}_i^{m,inx}[t] - \mathbf{x}_i^{m}[t])\right)_{i\in\mathcal{N}}$. *This simply follows from Lines 7 and 9 of Algorithm 2.2.*

*(e)* $\bar{\mathbf{x}}^{m}[t] = \bar{\mathbf{x}}^{m}[t-1] + \frac{\alpha[t-1]}{n_m}\left(\mathbf{1}_{\mathcal{N}_m}^{\top} \otimes \mathbf{I}_{d_m}\right)\Delta\mathbf{x}^{m,inx}[t-1]$. *This follows from applying (c) to (d).*

**Lemma A.5:** *Let* $0 < \lambda < 1$, *and let* $\{\beta[t]\}$ *and* $\{\nu[t]\}$ *be two positive scalar sequences. Then*

*(a)* *If* $\lim_{t\to\infty}\beta[t] = 0$, *then* $\lim_{t\to\infty}\sum_{s=1}^{n}\lambda^{t-s}\beta[s] = 0$.

*(b)* *If further we have* $\sum_{t=1}^{\infty}\beta^2[t] < \infty$ *and* $\sum_{t=1}^{\infty}\nu^2[t] < \infty$, *then* $\lim_{t\to\infty}\sum_{k=1}^{t}\sum_{s=1}^{k}\lambda^{k-s}\beta^2[s] < \infty$ *and* $\lim_{t\to\infty}\sum_{k=1}^{t}\sum_{s=1}^{k}\lambda^{k-s}\beta[k]\nu[s] < \infty$.

**Lemma A.6:** *Let* $\{Y[t]\}$, $\{X[t]\}$, *and* $\{Z[t]\}$ *be three sequences of numbers such that* $X[t] \geq 0$ *for all* $n$. *If* $Y[t+1] \leq Y[t] - X[t] + Z[t]$ *for all* $n$ *and* $\sum_{t=1}^{\infty} Z[t] < \infty$, *then either* $Y[t] \to -\infty$ *or* $\{Y[t]\}$ *converges to a finite value and* $\sum_{t=1}^{\infty} X[t] < \infty$.

**Lemma A.7:** *Let* $0 < \lambda < 1$, *and let* $\{\beta[t]\}$ *and* $\{\nu[t]\}$ *be two positive scalar sequences such that* $\beta[t] \to 0$, $\nu[t] \to \infty$, *and* $\beta[t]\nu[t] \to 0$. *If further there exist* $1 > \tilde{\lambda} > \lambda$ *and* $N$ *such that* $\frac{\beta[t]}{\beta[s]} \geq \tilde{\lambda}^{t-s}$ *for all* $n \geq l \geq N$, *then* $\lim_{t\to\infty}\nu[t]\sum_{s=1}^{n}\lambda^{t-s}\beta[s] = 0$. *Moreover, if* $\beta[t]\nu[t]$ *is summable, then so is* $\nu[t]\sum_{s=1}^{n}\lambda^{t-s}\beta[s]$.

**Proof:** The proof of the first part of the claim is straightforward.

$$\begin{aligned}\nu[t]\sum_{s=1}^{n}\lambda^{t-s}\beta[s] &= \nu[t]\sum_{s=1}^{t-1}\lambda^{t-s}\beta[s] + \beta[t]\nu[t]\sum_{l=N}^{n}\lambda^{t-s}\frac{\beta[s]}{\beta[t]}\\ &\leq \nu[t]\sum_{s=1}^{t-1}\lambda^{t-s}\beta[s] + \beta[t]\nu[t]\sum_{l=N}^{n}\frac{\lambda^{t-s}}{\tilde{\lambda}^{t-s}} \qquad \text{(A.28)}\\ &\leq \nu[t]\sum_{s=1}^{t-1}\lambda^{t-s}\beta[s] + \beta[t]\nu[t]\cdot\frac{1}{1-\lambda/\tilde{\lambda}}.\end{aligned}$$

The second term goes to zero by the condition. Note that the meaning of $\frac{\beta[t]}{\beta[s]} \geq \tilde{\lambda}^{t-s}$ basically says $\beta[t]$ cannot decay to zero faster than at an exponential rate. Thus, $\beta[t]\nu[t] \to 0$ would imply that $\lambda^n\nu[t] \to 0$ as well. For the second part of the claim, just sum (A.28) over $n$. ■

## A.4 Proof of Theorem 2.6

From [55], with the notation of $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1^\top & \dots & \mathbf{x}_n^\top\end{bmatrix}^\top$, (2.6) can be written in matrix form as

$$\mathbf{X}[t+1] = (\mathbf{W}\otimes\mathbf{I}_d)\mathbf{X}[t] - \alpha[t]\nabla F(\mathbf{X}[t]).$$

Multiplying both sides by $(\mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\otimes\mathbf{I}_d := \mathbf{E}_n\otimes\mathbf{I}_d$ yields

$$\tilde{\mathbf{X}}[t+1] = (\tilde{\mathbf{W}}\otimes\mathbf{I}_d)\tilde{\mathbf{X}}[t] - \alpha[t]\mathbf{E}_n\otimes\mathbf{I}_d\nabla F(\mathbf{X}[t]), \qquad \text{(A.29)}$$

where $\tilde{\mathbf{X}} = \mathbf{E}_n\otimes\mathbf{I}_d\mathbf{X}$ and $\tilde{\mathbf{W}} = \mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. It is established that with the optimal choice of $\alpha[t]$, we have

$$\|\tilde{\mathbf{X}}[t]\| \leq \frac{\sqrt{nd}GB(\mu)}{2Lt^{1/3}}. \qquad \text{(A.30)}$$

Note that $\mu := \|\tilde{\mathbf{W}}\otimes\mathbf{I}_d\|_2$ does not depend on the dimension $d$. This inequality is the basis of where the terms in (2.7) come from.

Now, after localization, with the similar concatenating vector notation, for all $m$

$$\mathbf{X}^m[t+1] = (\mathbf{W}^m\otimes\mathbf{I}_d)\mathbf{X}^m[t] - \alpha[t]\nabla_{\mathbf{x}^m} f_i(\mathbf{x}_i^{\mathcal{S}_i}[t]).$$

Similar to (A.29), we multiply both sides of (A.30) by $\mathbf{E}_{n_m}\otimes\mathbf{I}_{d_m}$ where $d_m$ is the dimension of $\mathbf{x}^m$, getting

$$\tilde{\mathbf{X}}^m[t+1] = (\tilde{\mathbf{W}}^m \otimes \mathbf{I}_{d_m})\tilde{\mathbf{X}}^m[t] - \alpha[t]\mathbf{E}_{n_m} \otimes \mathbf{I}_{d_m} \nabla_{\mathbf{x}^m} F(\mathbf{X}[t]). \tag{A.31}$$

Then collect all the parts to make an ensemble vector $\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{X}}^{1\top} & \dots & \tilde{\mathbf{X}}^{n\top} \end{bmatrix}^\top$ so that (A.31) becomes

$$\tilde{\mathbf{X}}[t+1] = \tilde{\mathbf{W}}\tilde{\mathbf{X}}[t] - \text{terms}(\nabla F),$$

where

$$\tilde{\mathbf{W}} = \begin{bmatrix} \tilde{\mathbf{W}}^1 \otimes \mathbf{I}_{d_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \tilde{\mathbf{W}}^M \otimes \mathbf{I}_{d_M} \end{bmatrix}.$$

Note that

$$\|\tilde{\mathbf{W}}\|_2 = \max\{\|\tilde{\mathbf{W}}^1 \otimes \mathbf{I}_{d_1}\|_2, \dots, \|\tilde{\mathbf{W}}^M \otimes \mathbf{I}_{d_M}\|_2\} = \max\{\mu_1, \dots, \mu_M\} \triangleq \mu'.$$

These lead to

$$\|\tilde{\mathbf{X}}[t]\| \leq \frac{\sqrt{\sum_{m=1}^M n_m d_m} GB(\mu')}{2Lt^{1/3}}.$$

## A.5 Proof of Proposition 2.20

We will prove a stronger result. Given any weights $\mathbf{c} \in \mathbb{R}^{|\mathcal{E}|}$ for the original set of edges $\mathcal{E}$. Let $\epsilon$ be the weight for the additional edge $vv_n$, with the constraint that $0 \leq \epsilon \leq c_n$ where $c_n = 1 - \sum_{j \in N(v_n)} c_{v_j v_n}$ is the weight for the self-loop edge of $v_n$ and $N(v_n)$ denote the set of neighboring nodes of $v_n$. Then

$$\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon)) = \left[\begin{array}{c|c} \mathbf{W}_{\mathcal{G}}(\mathbf{c}) & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right] + \left[\begin{array}{c|cc} \mathbf{0} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & -\epsilon & \epsilon \\ & \epsilon & -\epsilon \end{array}\right]. \tag{A.32}$$

Viewing (A.32) in the form of $\mathbf{M} = \mathbf{H} + \mathbf{R}$ (note that they are all $n \times n$ matrices) and invoking Weyl's inequality with $i = 2$, $j = 3$, and $k = t - 1$, so that $j + k - n =$

$3 + (t-1) - n \geq 2 = i$ is satisfied, we get $\mu_2 \geq \nu_3 + \rho_{t-1}$. This means

$$\begin{aligned}
&\lambda_2\left(\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon))\right) \\
&\geq \lambda_3\left(\left[\begin{array}{c|c} \mathbf{W}_{\mathcal{G}}(\mathbf{c}) & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right]\right) + \lambda_{-2}\left(\left[\begin{array}{c|cc} \mathbf{0} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & -\epsilon & \epsilon \\ & \epsilon & -\epsilon \end{array}\right]\right) \\
&= \lambda_2\left(\mathbf{W}_{\mathcal{G}}(\mathbf{c})\right) + 0 = \lambda_2\left(\mathbf{W}_{\mathcal{G}}(\mathbf{c})\right).
\end{aligned} \tag{A.33}$$

This equality is due to the following argument. For the first matrix, the value 1 is one of its largest eigenvalue; thus, its third largest eigenvalue will be the second largest eigenvalue of its submatrix $\mathbf{W}_{\mathcal{G}}(\mathbf{c})$. The eigenvalues of the second matrix are $\{0, \ldots, 0, -2\epsilon\}$; hence, the second smallest eigenvalue of this matrix is 0.

Similarly, invoking Weyl's inequality again with $i = n$, $r = n$, and $s = 1$, so that $i = n \geq n + t - 1 = r + s - 1$ is satisfied, we get $\mu_n \leq \nu_n + \rho_1$. This means

$$\begin{aligned}
&\lambda_{-1}\left(\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon))\right) \\
&\leq \lambda_{-1}\left(\left[\begin{array}{c|c} \mathbf{W}_{\mathcal{G}}(\mathbf{c}) & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right]\right) + \lambda_{1}\left(\left[\begin{array}{c|cc} \mathbf{0} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & -\epsilon & \epsilon \\ & \epsilon & -\epsilon \end{array}\right]\right) \\
&= \lambda_{-1}\left(\mathbf{W}_{\mathcal{G}}(\mathbf{c})\right) + 0 = \lambda_{-1}\left(\mathbf{W}_{\mathcal{G}}(\mathbf{c})\right),
\end{aligned} \tag{A.34}$$

or equivalently $-\lambda_{-1}\left(\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon))\right) \geq -\lambda_{-1}\left(\mathbf{W}_{\mathcal{G}}(\mathbf{c})\right)$. Combining (A.33) and (A.34), given any $\mathbf{c} \in \mathcal{C}_{\mathcal{G}}$ defined in (2.21) and any $0 \leq \epsilon \leq c_n$, we have the stronger relation

$$\begin{aligned}
{\mu'}_{\mathcal{G}}(\mathbf{c}) &= \mu_{\mathcal{G}}(\mathbf{W}_{\mathcal{G}}(\mathbf{c})) = \max\{\lambda_2(\mathbf{W}_{\mathcal{G}}(\mathbf{c})), -\lambda_{-1}(\mathbf{W}_{\mathcal{G}}(\mathbf{c}))\} \\
&\leq \max\{\lambda_2(\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon))), -\lambda_{-1}(\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon)))\} \\
&= \mu_{\mathcal{G}}(\mathbf{W}_{\mathcal{G}'}((\mathbf{c}, \epsilon))) = {\mu'}_{\mathcal{G}'}((\mathbf{c}, \epsilon)).
\end{aligned}$$

From this we obtain

$$\rho(\mathcal{G}) = \min_{\mathbf{c} \in \mathcal{C}_{\mathcal{G}}} \mu'_{\mathcal{G}}(\mathbf{c}) \leq \min_{\mathbf{c} \in \mathcal{C}_{\mathcal{G}}, 0 \leq \epsilon \leq c_n} \mu'_{\mathcal{G}'}((\mathbf{c}, \epsilon)) = \rho(\mathcal{G}').$$

## A.6 Pseudo Code of the Algorithms in Section 2.6

In this appendix we give the complete pseudo code of algorithms in Section 2.6 for reader's reference. All $\bar{t}$'s represent $t+1$ for compression.

### A.6.1 LXGP-RM

The LXGP-RM algorithm is given as follows:

**Algorithm A.1** LXGP-RM

```
Initialization: $\forall\ b$, $p^b_{BK}[0] = \epsilon$, $x_{bI(b)K}[0] = 0$, $r^b_{BK}[0] = 0$, $\tilde{\pi}^b_{BK}[0] = 0$, $t = 0$
while $p^b_{BK}[t]$ and $x_{bI(b)K}[t]$ do not satisfy the termination criterion do
    $t \leftarrow t + 1$
    $\alpha[t] = \frac{\alpha_0}{(t+1)^\beta}$
    $(\tilde{p}^b_{BK}[t], \tilde{x}_{bI(b)K}[t]) = \underset{p^b_{BK}, x_{bI(b)K}}{\arg\min}\ \tilde{f}_b(p^b_{BK}, x_{bI(b)K}; p^b_{BK}[t], x_{bI(b)K}[t])$
            $+\tilde{\pi}^b_{BK}[t] \cdot (p^b_{BK} - p^b_{BK}[t])$
    $q^b_{BK}[t] = p^b_{BK}[t] + \alpha[t](\tilde{p}^b_{BK}[t] - p^b_{BK}[t])$
    $x_{bI(b)K}[\bar{t}] = x_{bI(b)K}[t] + \alpha[t](\tilde{x}_{bI(b)K}[t] - x_{bI(b)K}[t])$
    $p^b_{BK}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'} q^{b'}_{BK}[t]$
    $r^b_{BK}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'} r^{b'}_{BK}[t]$
            $+ \left[\nabla_{p^b_{BK}} f_b(p^b_{BK}[\bar{t}], x_{bI(b)K}[\bar{t}]) - \nabla_{p^b_{BK}} f_b(p^b_{BK}[t], x_{bI(b)K}[t])\right]$
    $\tilde{\pi}^b_{BK}[\bar{t}] = |B| \mathbf{1}_{BK} \circ r^b_{BK}[\bar{t}] - \nabla_{p^b_{BK}} f_b(p^b_{BK}[\bar{t}], x_{bI(b)K}[\bar{t}])$
Output: $p^b_{bK}[t]$ and $x_{bI(b)K}[t]$
```

In the algorithm, $p^b_{BK}$ is the collection of the variables $p^b_{b'k}\ \forall\ b' \in B, k \in K$, and so are the other quantities $x_{bI(b)K}$, $q^b_{BK}$, etc. The collection of variables $p^b_{BK}$ can be viewed as a $B \times K$ matrix, or a vector of $BK$ dimensions. $\mathbf{1}_{BK}$ is a $B \times K$ matrix (or vector) consisting of all 1's. $\circ$ denotes the element-wise product, also known as Hadamard product or Schur product.

### A.6.2 LXLP-RM

Using the same notations as in LXGP-RM, the LXLP-RM algorithm is given as follows:

**Algorithm A.2** LXLP-RM

```
1  Initialization: $\forall\ b$, $p^b_{Nb(b)K}[0] = \epsilon$, $x_{bI(b)K}[0] = 0$, $r^b_{Nb(b)K}[0] = 0$, $\tilde{\pi}^b_{Nb(b)K}[0] = 0$, $t = 0$
2  while $p^b_{Nb(b)K}[t]$ and $x_{bI(b)K}[t]$ do not satisfy the termination criterion do
3      $t \leftarrow t + 1$
4      $\alpha[t] = \frac{\alpha_0}{(t+1)^\beta}$
5      $(\tilde{p}^b_{Nb(b)K}[t], \tilde{x}_{bI(b)K}[t]) = \underset{p^b_{Nb(b)K}, x_{bI(b)K}}{\arg\min}\ \tilde{f}_b(p^b_{Nb(b)K}, x_{bI(b)K}; p^b_{Nb(b)K}[t], x_{bI(b)K}[t])$
           $+\tilde{\pi}^b_{Nb(b)K}[t] \cdot (p^b_{Nb(b)K} - p^b_{Nb(b)K}[t])$
6      $q^b_{Nb(b)K}[t] = p^b_{Nb(b)K}[t] + \alpha[t](\tilde{p}^b_{Nb(b)K}[t] - p^b_{Nb(b)K}[t])$
7      $x_{bI(b)K}[\bar{t}] = x_{bI(b)K}[t] + \alpha[t](\tilde{x}_{bI(b)K}[t] - x_{bI(b)K}[t])$
8      $p^b_{Nb(b)K}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'}(Nb(b)) q^{b'}_{Nb(b)K}[t]$
9      $r^b_{Nb(b)K}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'}(Nb(b)) r^{b'}_{Nb(b)K}[t]$
           $+ \left[\nabla_{p^b_{Nb(b)K}} f_b(p^b_{Nb(b)K}[\bar{t}], x_{bI(b)K}[\bar{t}]) - \nabla_{p^b_{Nb(b)K}} f_b(p^b_{Nb(b)K}[t], x_{bI(b)K}[t])\right]$
10     $\tilde{\pi}^b_{Nb(b)K}[\bar{t}] = [(d_{Nb(b)} + \mathbf{1}_{Nb(b)})\mathbf{1}_K^\top] \circ r^b_{Nb(b)K}[\bar{t}] - \nabla_{p^b_{Nb(b)K}} f_b(p^b_{Nb(b)K}[\bar{t}], x_{bI(b)K}[\bar{t}])$
Output: $p^b_{bK}[t]$ and $x_{bI(b)K}[t]$
```

In line 10, for all $b'' \in Nb(b)$, the notation stands for $p^b_{b''K}[t+1]$ being updated as $\sum_{b' \in Nb(b)} W_{bb'}(b'') q^{b'}_{b''K}[t]$, and so do line 11 and 12. In line 12, $d_{b''}$ means the degree of $b''$, i.e. $d_{b''} = |N(b'')|$. If considering $\tilde{\pi}^b_{Nb(b)K}$ to be a $Nb(b) \times K$ matrix, then $(d_{Nb(b)} + \mathbf{1}_{Nb(b)})\mathbf{1}_K^\top$ consists of $Nb(b)$ rows; each row has $|K|$ elements, and every element in the $b''$-th row is $d_{b''} + 1 = |Nb(b'')|$.

### A.6.3 GXGP-CM

the GXGP-CM algorithm is given as follows:

**Algorithm A.3** GXGP-CM

```
Initialization: $\forall\ b$, $p^b_{BK}[0] = \epsilon$, $x^b_{BI(B)K}[0] = 0$, $r^b_{BK}[0] = 0$, $\tilde{\pi}^b_{BK}[0] = 0$, $y^b_{BI(B)K}[0] = 0$, $\tilde{\tau}^b_{BI(B)K}[0] = 0$, $t = 0$
while $p^b_{BK}[t]$ and $x^b_{BI(B)K}[t]$ do not satisfy the termination criterion do
    $t \leftarrow t + 1$
    $\alpha[t] = \frac{\alpha_0}{(t+1)^\beta}$
    $(\tilde{p}^b_{BK}[t], \tilde{x}^b_{BI(B)K}[t]) = \underset{p^b_{BK}, x^b_{BI(B)K}}{\arg\min}\ \tilde{f}_b(p^b_{BK}, x^b_{BI(B)K}; p^b_{BK}[t], x^b_{BI(B)K}[t])$
            $+\tilde{\pi}^b_{BK}[t] \cdot (p^b_{BK} - p^b_{BK}[t]) + \tilde{\tau}^b_{BI(B)K}[t] \cdot (x^b_{BI(B)K} - x^b_{BI(B)K}[t]) + G(p^b_{BK}, x^b_{BI(B)K})$
    $q^b_{BK}[t] = p^b_{BK}[t] + \alpha[t](\tilde{p}^b_{BK}[t] - p^b_{BK}[t])$
    $z^b_{BI(B)K}[\bar{t}] = x^b_{BI(B)K}[t] + \alpha[t](\tilde{x}^b_{BI(B)K}[t] - x^b_{BI(B)K}[t])$
    $p^b_{BK}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'} q^{b'}_{BK}[t]$
    $x^b_{BI(B)K}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'} z^{b'}_{BI(B)K}[t]$
    $r^b_{BK}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'} r^{b'}_{BK}[t]$
            $+ \left[\nabla_{p^b_{BK}} f_b(p^b_{BK}[t+1], x^b_{BI(B)K}[\bar{t}]) \nabla_{p^b_{BK}} f_b(p^b_{BK}[t], x^b_{BI(B)K}[t])\right]$
    $y^b_{BI(B)K}[\bar{t}] = \sum_{b' \in Nb(b)} W_{bb'} y^{b'}_{BI(B)K}[t]$
            $+ \left[\nabla_{x^b_{BI(B)K}} f_b(p^b_{BK}[\bar{t}], x^b_{BI(B)K}[\bar{t}]) - \nabla_{x^b_{BI(B)K}} f_b(p^b_{BK}[t], x^b_{BI(B)K}[t])\right]$
    $\tilde{\pi}^b_{BK}[\bar{t}] = |B| \mathbf{1}_{BK} \circ r^b_{BK}[\bar{t}] - \nabla_{p^b_{BK}} f_b(p^b_{BK}[\bar{t}], x_{bI(b)K}[\bar{t}])$
    $\tilde{\tau}^b_{BI(B)K}[\bar{t}] = |B| \mathbf{1}_{BI(B)K} \circ y^b_{BI(B)K}[\bar{t}] - \nabla_{x^b_{BI(B)K}} f_b(p^b_{BK}[\bar{t}], x^b_{BI(B)K}[\bar{t}])$
Output: $p^b_{bK}[t]$ and $x^b_{bI(b)K}[t]$
```

# APPENDIX B

# Proofs for Chapter III

**Definition B.1:** *For an ensemble vector* $\mathbf{X} \triangleq [\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_n^\top]^\top$ *where* $\mathbf{x}_i = [x_{i,1} \ \cdots \ x_{i,d}]^\top \in \mathbb{R}^d$ *for* $i \in [n]$*, we define its range* $sp(\mathbf{X})$ *to be*

$$sp(\mathbf{X}) \triangleq d \cdot \max_{l\in[d]} \left( \max_i x_{i,l} - \min_i x_{i,l} \right).$$

*Note that if we let* $\mathbf{X}_l \triangleq [x_{1,l} \ \cdots \ x_{n,l}]^\top$ *to be the dimension* $l \in [d]$ *of the ensemble, then the range can be expressed by* $sp(\mathbf{X}) = d \cdot \max_{l\in[d]} sp(\mathbf{X}_l)$*. In the context of this chapter, the* $i \in [n]$ *is the agent index while the* $l \in [d]$ *is the dimension index of the decision variable.*

**Lemma B.2:** *Let* $\mathbf{W} \in \mathbb{R}^{n\times n}$ *be a row stochastic matrix with all elements* $w_{ij} \in [\vartheta, 1-\vartheta]$ *for some* $\vartheta \in (0, 0.5)$*, and* $\mathbf{v} \in \mathbb{R}^n$ *be a vector. Then* $sp(\mathbf{W}\mathbf{v}) \leq (1-2\vartheta) sp(\mathbf{v})$*.*

**Proof:** Let $a \in \operatorname{argmax}_i v_i$, $b \in \operatorname{argmin}_i v_i$, $a' \in \operatorname{argmax}_i (Wv)_i$, and $b' \in \operatorname{argmin}_i (Wv)_i$. Then $sp(\mathbf{v}) = v_a - v_b$, and

$$\begin{aligned} sp(\mathbf{W}\mathbf{v}) &= \left( \sum_{j=1}^n W_{a'j} v_j \right) - \left( \sum_{j=1}^n W_{b'j} v_j \right) \\ &= W_{a'b} v_b + \left( \sum_{j\neq b} W_{a'j} v_j \right) - W_{b'a} v_a - \left( \sum_{j\neq a} W_{b'j} v_j \right) \\ &\leq W_{a'b} v_b + \left( \sum_{j\neq b} W_{a'j} \right) v_a - W_{b'a} v_a - \left( \sum_{j\neq a} W_{b'j} \right) v_b \end{aligned}$$

$$
\begin{aligned}
&= W_{a'b}v_b + (1 - W_{a'b})\, v_a - W_{b'a}v_a - (1 - W_{b'a})\, v_b \\
&= (1 - W_{a'b} - W_{b'a}) \cdot (v_a - v_b) \leq (1 - 2\vartheta) sp(\mathbf{v}).
\end{aligned}
$$

■

**Proof of Theorem 3.7:** The main intuition behind the proof is that even with time-varying and agent-dependent transformations, the consensus scheme is still a subprojection converging exponentially fast onto the consensus plane $\mathcal{C}$. Lemma B.2 is the prototype of the convergence, which shows that the range of a vector shrinks geometrically when being multiplied by a doubly stochastic matrix with positive elements. With a fixed transformation, it is easy to see that the convergence property remains – performing a change of variable $z = \varphi(x)$, the nonlinear consensus scheme is a linear consensus scheme in the $z$-domain, so the range of $z$ decreases exponentially, which also implies the range of $x$ decreases exponentially because of the Lipschitz assumption. However, when the communication graph is not fully connected, the transformation is time-varying and agent-dependent, and the process is coupled with the distributed descent process, more careful treatment is required.

We start by showing that the nonlinear consensus scheme with time-varying and agent-dependent transformations and a fixed (possibly not fully-) connected communication graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a subprojection onto $\mathcal{C}$. Let $D = diam(\mathcal{G})$ be the diameter of the graph $\mathcal{G}$, then it is easy to show that for any doubly stochastic matrix $\mathbf{W}$ satisfying the constraint given in Section 2.2.1 (that is, $W_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$ for $i \neq j$ and $W_{ii} > 0$), the matrix $\mathbf{W}^D$ is a doubly stochastic matrix with positive elements. Let $\vartheta > 0$ be the smallest entry of $\mathbf{W}$. Without the transformations, the range of the vector will shrink by $(1 - 2\vartheta^D)$ from Lemma B.2 every $D$ steps as the smallest entry in $\mathbf{W}^D$ will be $\vartheta^D$. We show that this is still the case with the transformations. We consider the averaging process

$$
\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_{i,t}(\mathbf{x}_j[t])\right);
$$

note that without the descent process, different dimensions do not couple together, so we only focus on dimension $l \in [d]$: $x_{i,l}[t+1] = \varphi_{i,t,l}^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_{i,t,l}(x_{j,l}[t])\right)$

and from now on we will omit the dimension index $l$. Consider at time $t$, we have $x_{\max} \triangleq \max_i x_i[t]$ and $x_{\min} \triangleq \min_i x_i[t]$, so that $sp(\mathbf{X}[t]) = x_{\max} - x_{\min}$ where $\mathbf{X} = [x_1 \ \cdots \ x_n]^\top$ is the ensemble vector of the variables from all agents (only for the considered dimension $l$). Let $i^* \in \operatorname{argmax}_i x_i[t]$ be one of the nodes with the maximum value at $t$; the value will spread through the network within $D$ steps. For any node $j$ that is a direct neighbor of $i^*$, assuming without loss of generality that $\varphi_{j,t}$ is increasing instead of decreasing, we have

$$\begin{aligned}
\varphi_{j,t}(x_j[t+1]) = \sum_k W_{jk}\varphi_{j,t}(x_k[t]) &= W_{ji^*}\varphi_{j,t}(x_{i^*}[t]) + \sum_{k\neq i^*} W_{jk}\varphi_{j,t}(x_k[t]) \\
&\geq W_{ji^*}\varphi_{j,t}(x_{\max}) + \sum_{k\neq i^*} W_{jk}\varphi_{j,t}(x_{\min}) \qquad (*) \\
&\geq \vartheta\varphi_{j,t}(x_{\max}) + (1-\vartheta)\varphi_{j,t}(x_{\min}).
\end{aligned}$$

Hence, after one step, $x_j$ can be lower bounded by

$$\begin{aligned}
x_j[t+1] &\geq x_{\min} + \frac{1}{L_+}\left[\varphi_{j,t}(x_j[t+1]) - \varphi_{j,t}(x_{\min})\right] && \text{(Lipschitz continuity of } \varphi_{j,t}) \\
&\geq x_{\min} + \frac{1}{L_+}\cdot\vartheta\left[\varphi_{j,t}(x_{\max}) - \varphi_{j,t}(x_{\min})\right] && \text{(by (*))} \\
&\geq x_{\min} + \frac{\vartheta}{L_+L_-}(x_{\max} - x_{\min}). && \text{(Lipschitz continuity of } \varphi_{j,t}^{-1})
\end{aligned}$$

Similarly, after two steps, for any node $j'$ whose shortest path to $i^*$ is within two hops, we have

$$x_{j'}[t+2] - x_{\min} \geq \frac{\vartheta}{L_+L_-}\left(x_j[t+1] - x_{\min}\right) \geq \left(\frac{\vartheta}{L_+L_-}\right)^2 (x_{\max} - x_{\min})$$

as it will be averaged with one of $j$ which is $i^*$'s direct neighbor. We can conclude that after $D$ steps, for any node $j$ in the network, we will have

$$x_j[t+D] \geq x_{\min} + \left(\frac{\vartheta}{L_+L_-}\right)^D (x_{\max} - x_{\min})$$

and by a symmetric argument

$$x_j[t+D] \leq x_{\max} - \left(\frac{\vartheta}{L_+L_-}\right)^D (x_{\max} - x_{\min});$$

thus, the range of $\mathbf{X}[t+D]$ also shrinks with a factor smaller than 1 in comparison to the range of $\mathbf{X}[t]$:

$$sp(\mathbf{X}[t+D]) \leq \left[1 - 2\left(\frac{\vartheta}{L_+L_-}\right)^D\right](x_{\max} - x_{\min}) = \left[1 - 2\left(\frac{\vartheta}{L_+L_-}\right)^D\right] sp(\mathbf{X}[t]).$$

Note that the factor $\rho \triangleq \left[1 - 2\left(\frac{\vartheta}{L_+L_-}\right)^D\right]$ will be between 0 and 1, i.e. $\rho \in [0,1)$. This is because of the following two reasons. First, $\vartheta$ is defined as the smallest non-zero entry in $\mathbf{W}$; we cannot have 1 in $\mathbf{W}$ since that will imply the graph is not connected, so $\vartheta$ will be the largest when all rows in $\mathbf{W}$ have support equal to 2 and all entries are either 0 or $\frac{1}{2}$; hence, $0 < \vartheta \leq \frac{1}{2}$. On the other hand, we have $L_+L_- \geq 1$, where the equality only attains when all the transformations are linear. The factor $\rho$ only equals to 0 in the trivial case of $\vartheta = \frac{1}{2}$, $D = 1$, and $L_+L_- = 1$; otherwise, it is positive but strictly less than 1. The consensus scheme is thus a subprojection as the range of $\mathbf{X}$ converges to 0 as its components asymptotically agree.

Now we take the descent process into account. Note that the proof of Proposition 9 Part (a) in [80] still follows as it only involves the consensus scheme of the $\mathbf{y}$ variable, which remains unchanged in our algorithm; the descent vector $\|\Delta\mathbf{x}_i^{inx}[t]\| = \|\mathbf{x}_i^{indx}[t] - \mathbf{x}_i[t]\| \leq c_1$ is thus bounded by some constant. To prove Theorem 3.7, we only have to show Proposition 9 Part (b) in [80], i.e. the asymptotic agreement of $\mathbf{x}_i$'s, with the nonlinear consensus scheme of $\mathbf{x}$. Recall that in Algorithm 3.1, we have $\mathbf{z}_i[t] = \mathbf{x}_i[t] + \alpha[t]\Delta\mathbf{x}_i^{inx}[t]$ and $\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^n W_{ij}\varphi_{i,t}(\mathbf{z}_j[t])\right)$; expanding the result gives

$$\begin{aligned} &\left|\varphi_{i,t,l}^{-1}\left(\sum_{j=1}^n W_{ij}\varphi_{i,t,l}(x_{j,l}[t])\right) - x_{i,l}[t+1]\right| \\ \leq\ & L_-\left|\sum_{j=1}^n W_{ij}\varphi_{i,t,l}(x_{j,l}[t]) - \varphi_{i,t,l}(x_{i,l}[t+1])\right| \qquad \text{(Lipschitz continuity of } \varphi_{i,t,l}^{-1}) \end{aligned}$$

$$
\begin{aligned}
&= L_- \left| \sum_{j=1}^n W_{ij} \varphi_{i,t,l}(z_{j,l}[t] - \alpha[t] \Delta x_{j,l}^{inx}[t]) - \varphi_{i,t,l}(x_{i,l}[t+1]) \right| \\
&\leq L_- \left| \sum_{j=1}^n W_{ij} \varphi_{i,t,l}(z_{j,l}[t]) - \varphi_{i,t,l}(x_{i,l}[t+1]) \right| + L_+ L_- \sum_{j=1}^n W_{ij} \alpha[t] \left| \Delta x_{j,l}^{inx}[t] \right| \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Lipschitz continuity of } \varphi_{i,t,l}) \\
&= 0 + L_+ L_- \sum_{j=1}^n W_{ij} \alpha[t] \left| \Delta x_{j,l}^{inx}[t] \right| \leq c_2 \alpha[t]
\end{aligned}
$$

for some fixed constant $c_2$ for dimension $l$. In the worst scenario, the added $\alpha$'s always go against the consensus direction, so that in the previous projection argument we have

$$x_j[t+1] \geq x_{\min} + \frac{\vartheta}{L_+ L_-}(x_{\max} - x_{\min}) - c_2 \alpha[t]$$

for $i^*$'s direct neighbors and

$$x_{j'}[t+2] \geq x_{\min} + \left( \frac{\vartheta}{L_+ L_-} \right)^2 (x_{\max} - x_{\min}) - \frac{c_2 \vartheta}{L_+ L_-} \alpha[t] - c_2 \alpha[t+1]$$

for $i^*$'s 2-hop neighbors, etc. After $D$ steps, we have the range of $\mathbf{X}$ shrunk by

$$sp(\mathbf{X}[t+D]) \leq \rho \cdot sp(\mathbf{X}[t]) + 2c_2 \sum_{s=0}^{D-1} \hat{\rho}^{D-1-s} \alpha[t+s]$$

where $\hat{\rho} = \frac{\vartheta}{L_+ L_-} \in (0, \frac{1}{2})$. Further inducting $t$ backwards to $t = 0$ gives

$$sp(\mathbf{X}[t]) \leq \rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0]) + c_3 \sum_{s=1}^{t} \hat{\rho}^{t-s} \alpha[s-1] \xrightarrow{t \to \infty} 0. \tag{B.1}$$

With the range converging to 0, the variables from all the nodes asymptotically agree. Note that (B.1) holds for any dimension $l \in [d]$ and $\mathbf{X}_l[t] = [x_{1,l}[t] \ \cdots \ x_{n,l}[t]]^\top$; therefore, it holds for the entire ensemble $\mathbf{X}[t] \triangleq [\mathbf{x}_1^\top[t] \ \cdots \ \mathbf{x}_n^\top[t]]^\top$, with $sp(\mathbf{X}[t]) = d \cdot \max_{l \in [d]} sp(\mathbf{X}_l[t])$ defined in Definition B.1. Moreover, since the deviation from the mean $\|\mathbf{x}_\perp[t]\| = \|\mathbf{X}[t] - \mathbf{1}_n \otimes \bar{\mathbf{x}}[t]\| \leq \|sp(\mathbf{X}[t])\|$ will be bounded by the range, we established Eq. (62) in [80]. Also, since the form of (B.1) nearly coincides with that of Eq. (77) from [80], Eq. (63) and Eq. (64) can be obtained in exactly the same ways as in [80], and we have reestablished Part (b) of Proposition 9 from [80]. ■

**Proof of Theorem 3.8:** We follow similar framework as [132] and [13], and classic analysis of gradient descent [25]. We begin with one lemma and its corollary to show that the gradients are bounded from the gradient of mean.

**Lemma B.3** (Bounded deviation from mean)**:** *For any* $i \in [n]$ *and* $t$*, with a constant learning rate* $\alpha[t] = \alpha$*, we have*

$$\|\mathbf{x}_i[t] - \bar{\mathbf{x}}[t]\| \leq sp(\mathbf{X}[t]) \leq \rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0]) + \frac{2\alpha B L_+ L_-}{1-\hat{\rho}}, \tag{B.2}$$

*where* $\hat{\rho} = \frac{\vartheta}{L_+ L_-} \in (0, \frac{1}{2})$, $\rho = \left[1 - 2\left(\frac{\vartheta}{L_+ L_-}\right)^D\right] \in [0, 1)$, *and* $D = diam(\mathcal{G})$.

**Proof:** To show this, notice that the process of DGD with nonlinear transformation (Algorithm 3.2) is actually very similar to the process of inexact NEXT with nonlinear transformation (Algorithm 3.1); the derivation of Theorem 3.7, particularly Lemma B.2, can therefore be reused here. Recall that in Algorithm 3.1 at time $t$ for node $i$, the variable update is

$$\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^n W_{ij}\varphi_{i,t}\big(\mathbf{x}_j[t] + \alpha[t]\Delta\mathbf{x}_j^{inx}[t]\big)\right),$$

while in Algorithm 3.2 the variable update is

$$\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^n W_{ij}\varphi_{i,t}\big(\mathbf{x}_j[t] - \alpha[t]\nabla f_j(\mathbf{x}_j[t])\big)\right).$$

In Algorithm 3.1, the boundedness of $\Delta\mathbf{x}_i^{inx}[t]$ is already established in Proposition 9 Part (a) in [80] with any divergent but square summable schedule of $\alpha[t]$. Similarly, for DGD, for a fixed learning constant $\alpha[t] = \alpha$, it is shown that with strongly convex smooth objective functions of the nodes, the gradients $\nabla f_i(\mathbf{x}_i[t])$'s will remain bounded. This result has to be reestablished now after adding transformations; however, for simplicity, we will just apply the assumption that the gradient $\nabla f_i$ is uniformly bounded for all $i \in [n]$, i.e. $\|\nabla f_i\| \leq B$.

In the proof of Theorem 3.7, we show that every local average will move variables to each other at least by $\hat{\rho} = \frac{\vartheta}{L_+ L_-} \in (0, \frac{1}{2})$ while being dragged apart by $c_2\alpha[t]$ in the worst case; the net effect is in every $D = diam(\mathcal{G})$ steps, $sp(\mathbf{X}[t])$ will be shrunk by $\rho =$

$\left[1-2\left(\frac{\vartheta}{L_+L_-}\right)^D\right] \in [0,1)$ and will have an additional term $2c_2\sum_{s=0}^{D-1}\hat{\rho}^{D-1-s}\alpha[t+s]$. Here, the boundedness of $\Delta\mathbf{x}_j^{inx}[t]$ is replaced by the boundedness of $\nabla f_i$, so that similar to (B.1) we have

$$sp(\mathbf{X}[t]) \leq \rho^{\lfloor t/D\rfloor}\cdot sp(\mathbf{X}[0]) + 2BL_+L_-\sum_{s=1}^{t}\hat{\rho}^{t-s}\alpha[s-1]. \tag{B.3}$$

With $\alpha[t]=\alpha$ being constant and individual node's deviation from the mean bounded by the range

$$\|\mathbf{x}_i[t]-\bar{\mathbf{x}}[t]\| \leq \|\mathbf{x}_\perp[t]\| = \|\mathbf{X}[t]-\mathbf{1}_n\otimes\bar{\mathbf{x}}[t]\| \leq sp(\mathbf{X}[t]),$$

we have (B.2) for all $t$ and $i$. ■

**Corollary B.4** (Bounded gradient deviation from gradient of mean)**:** *For any $i\in[n]$ and $t$, with a constant learning rate $\alpha[t]=\alpha$ and all objective functions $f_i$'s being smooth, i.e. $\|\nabla f_i(\mathbf{x}_1)-\nabla f_i(\mathbf{x}_2)\| \leq L_f\|\mathbf{x}_1-\mathbf{x}_2\|$ for any $i\in[n]$ and $\mathbf{x}_1,\mathbf{x}_2\in\mathbb{R}^d$ and some constant $L_f>0$, we have*

$$\|\mathbf{g}[t]-\bar{\mathbf{g}}[t]\| \leq L_f\left[\rho^{\lfloor t/D\rfloor}\cdot sp(\mathbf{X}[0]) + \frac{2\alpha BL_+L_-}{1-\hat{\rho}}\right], \tag{B.4}$$

*where $\hat{\rho}=\frac{\vartheta}{L_+L_-}\in(0,\frac{1}{2})$, $\rho=\left[1-2\left(\frac{\vartheta}{L_+L_-}\right)^D\right]\in[0,1)$, $D=diam(\mathcal{G})$, $\mathbf{g}[t]\triangleq\frac{1}{n}\sum_{i=1}^n\nabla f_i(\mathbf{x}_i[t])$ and $\bar{\mathbf{g}}[t]\triangleq\frac{1}{n}\sum_{i=1}^n\nabla f_i(\bar{\mathbf{x}}[t])$.*

**Proof:** The result simply follows from the Lipschitz continuity of $\nabla f_i$'s and Lemma B.3, as

$$\|\mathbf{g}[t]-\bar{\mathbf{g}}[t]\| \leq \frac{1}{n}\sum_{i=1}^n\|\nabla f_i(\mathbf{x}_i[t])-\nabla f_i(\bar{\mathbf{x}}[t])\| \leq \frac{L_f}{n}\sum_{i=1}^n\|\mathbf{x}_i[t]-\bar{\mathbf{x}}[t]\|.$$

■

Since all objective functions $f_i$'s have Lipschitz continuous gradient with constant $L_f$, their sum $F=\sum_i f_i$ has Lipschitz continuous gradient with constant $nL_f$. Denote

$\Delta\bar{\mathbf{x}}[t] \triangleq \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i[t+1]-\mathbf{x}_i[t])$. Applying the descent lemma we get

$$\begin{aligned}
&F(\bar{\mathbf{x}}[t+1]) \\
&\leq F(\bar{\mathbf{x}}[t]) + \nabla F(\bar{\mathbf{x}}[t])^\top(\bar{\mathbf{x}}[t+1]-\bar{\mathbf{x}}[t]) + \frac{nL_f}{2}\|\bar{\mathbf{x}}[t+1]-\bar{\mathbf{x}}[t]\|^2 \\
&= F(\bar{\mathbf{x}}[t]) + n\bar{\mathbf{g}}[t]^\top\Delta\bar{\mathbf{x}}[t] + \frac{nL_f}{2}\|\Delta\bar{\mathbf{x}}[t]\|^2 \\
&= F(\bar{\mathbf{x}}[t]) - n\alpha\|\bar{\mathbf{g}}[t]\|^2 + n\bar{\mathbf{g}}[t]^\top(\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]) + \frac{nL_f}{2}\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]-\alpha\bar{\mathbf{g}}[t]\|^2 \\
&= F(\bar{\mathbf{x}}[t]) - n\alpha\left(1-\frac{\alpha L_f}{2}\right)\|\bar{\mathbf{g}}[t]\|^2 \\
&\qquad + n(1-\alpha L_f)\bar{\mathbf{g}}[t]^\top(\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]) + \frac{nL_f}{2}\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]\|^2 \\
&\leq F(\bar{\mathbf{x}}[t]) - n\left(\alpha - \frac{\alpha^2 L_f}{2} - \frac{c_4(1-\alpha L_f)}{2}\right)\|\bar{\mathbf{g}}[t]\|^2 \\
&\qquad + \frac{n}{2}\left(L_f + \frac{(1-\alpha L_f)}{c_4}\right)\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]\|^2
\end{aligned} \tag{B.5}$$

for any constant $c_4 > 0$, where the last inequality follows from the perfect square formula

$$n(1-\alpha L_f)\bar{\mathbf{g}}[t]^\top(\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]) \leq \frac{c_4 n(1-\alpha L_f)}{2}\|\bar{\mathbf{g}}[t]\|^2 + \frac{n(1-\alpha L_f)}{2c_4}\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]\|^2.$$

Taking $c_4 = \alpha$, (B.5) reduces to

$$F(\bar{\mathbf{x}}[t+1]) \leq F(\bar{\mathbf{x}}[t]) - \frac{n\alpha}{2}\|\bar{\mathbf{g}}[t]\|^2 + \frac{n}{2\alpha}\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]\|^2. \tag{B.6}$$

In DGD (with nonlinear transformation averaging) we let $\mathbf{z}_i[t] = \mathbf{x}_i[t]+\alpha\nabla f_i(\mathbf{x}_i[t])$ be the result of local optimization at node $i$, so that the new iterate is $\mathbf{x}_i[t+1] = \varphi_{i,t}^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_{i,t}(\mathbf{z}_j[t])\right)$. The term $\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]\|$ in (B.6) can then be bounded by

$$\begin{aligned}
&\|\Delta\bar{\mathbf{x}}[t]+\alpha\bar{\mathbf{g}}[t]\| \\
&\leq \|\alpha\bar{\mathbf{g}}[t]-\alpha\mathbf{g}[t]\| + \|\Delta\bar{\mathbf{x}}[t]+\alpha\mathbf{g}[t]\| \\
&= \alpha\|\mathbf{g}[t]-\bar{\mathbf{g}}[t]\| + \left\|\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i[t+1] - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i[t] + \frac{\alpha}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i[t])\right\|
\end{aligned}$$

$$= \alpha\|\mathbf{g}[t] - \bar{\mathbf{g}}[t]\| + \frac{1}{n}\left\|\sum_{i=1}^{n}\left[\varphi_{i,t}^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_{i,t}(\mathbf{z}_j[t])\right) - \mathbf{z}_i[t]\right]\right\|$$

$$\leq \alpha\|\mathbf{g}[t] - \bar{\mathbf{g}}[t]\| + \frac{1}{n}\sum_{l=1}^{d}\left|\sum_{i=1}^{n}\left[\varphi_{i,t,l}^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_{i,t,l}(z_{j,l}[t])\right) - z_{i,l}[t]\right]\right|.$$

(since $\|\cdot\|_2 \leq \|\cdot\|_1$)

(B.7)

We now use Taylor's theorem to expand $\varphi$ and $\varphi^{-1}$ with linear approximation (first order Taylor polynomial) and a quadratic remainder. For shorthand of notation, we will omit the dimension index $l$ and the time index $t$; it is understood that the derivation works for all $l$ and $t$. We first expand $\varphi_i(z_j)$ centered at $\bar{z}_i^W \triangleq \sum_{k=1}^{n} W_{ik}z_k$:

$$\varphi_i(z_j) = \varphi_i(\bar{z}_i^W) + \varphi_i'(\bar{z}_i^W)(z_j - \bar{z}_i^W) + \frac{\varphi_i''(\xi_{ij}^W)}{2}(z_j - \bar{z}_i^W)^2$$

for some number $\xi_{ij}^W$ between $\bar{z}_i^W$ and $z_j$. After being weighted-summed over $W_{ij}$, the linear term cleverly cancels out

$$\begin{aligned}
&\sum_{j=1}^{n} W_{ij}\varphi_i(z_j) \\
&= \varphi_i(\bar{z}_i^W)\cdot\sum_{j=1}^{n} W_{ij} + \varphi_i'(\bar{z}_i^W)\left(\sum_{j=1}^{n} W_{ij}z_j - \bar{z}_i^W\cdot\sum_{j=1}^{n} W_{ij}\right) + \sum_{j=1}^{n} W_{ij}\frac{\varphi_i''(\xi_{ij}^W)}{2}(z_j - \bar{z}_i^W)^2 \\
&= \varphi_i(\bar{z}_i^W)\cdot 1 + \varphi_i'(\bar{z}_i^W)\left(\sum_{j=1}^{n} W_{ij}z_j - \sum_{k=1}^{n} W_{ik}z_k\cdot 1\right) + \sum_{j=1}^{n} W_{ij}\frac{\varphi_i''(\xi_{ij}^W)}{2}(z_j - \bar{z}_i^W)^2 \\
&= \varphi_i(\bar{z}_i^W) + \sum_{j=1}^{n} W_{ij}\frac{\varphi_i''(\xi_{ij}^W)}{2}(z_j - \bar{z}_i^W)^2,
\end{aligned}$$

where $\sum_{j=1}^{n} W_{ij} = 1$ as $\mathbf{W}$ is row stochastic. Let $Q_{ij}^W \triangleq \sum_{j=1}^{n} W_{ij}\frac{\varphi_i''(\xi_{ij}^W)}{2}(z_j - \bar{z}_i^W)^2$ denote the quadratic remainder term. Next, we expand $\varphi_i^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_i(z_j)\right)$ centered at $\varphi_i(\bar{z}_i^W)$:

$$\varphi_i^{-1}\left(\sum_{j=1}^{n} W_{ij}\varphi_i(z_j)\right) = \varphi_i^{-1}\big(\varphi_i(\bar{z}_i^W)\big) + \varphi_i^{-1\prime}\big(\varphi_i(\bar{z}_i^W)\big)Q_{ij}^W + \frac{\varphi_i^{-1\prime\prime}(\varsigma_{ij}^W)}{2}{Q_{ij}^W}^2$$

$$= \bar{z}_i^W + {\varphi_i^{-1}}'\big(\varphi_i(\bar{z}_i^W)\big)Q_{ij}^W + \frac{{\varphi_i^{-1}}''(\varsigma_{ij}^W)}{2}{Q_{ij}^W}^2$$

for some number $\varsigma_{ij}^W$ between $\varphi_i(\bar{z}_i^W)$ and $\sum_{j=1}^n W_{ij}\varphi_i(z_j)$. With the expansions, the term inside the absolute value sign in (B.7) then becomes

$$\begin{aligned}
&\sum_{i=1}^n \left[\varphi_{i,t,l}^{-1}\left(\sum_{j=1}^n W_{ij}\varphi_{i,t,l}(z_{j,l}[t])\right) - z_{i,l}[t]\right] \\
&= \sum_{i=1}^n \left[\sum_{k=1}^n W_{ik}z_{k,l}[t] + {\varphi_{i,t,l}^{-1}}'\big(\varphi_{i,t,l}(\bar{z}_{i,l}^W[t])\big)Q_{ij,l}^W[t] + \frac{{\varphi_{i,t,l}^{-1}}''(\varsigma_{ij,l}^W[t])}{2}Q_{ij,l}^W[t]^2 - z_{i,l}[t]\right] \\
&= \sum_{i=1}^n \left[{\varphi_{i,t,l}^{-1}}'\big(\varphi_{i,t,l}(\bar{z}_{i,l}^W[t])\big)Q_{ij,l}^W[t] + \frac{{\varphi_{i,t,l}^{-1}}''(\varsigma_{ij,l}^W[t])}{2}Q_{ij,l}^W[t]^2\right] \\
&\qquad\qquad\qquad\qquad + \sum_{k=1}^n z_{k,l}[t]\cdot\sum_{i=1}^n W_{ik} - \sum_{i=1}^n z_{i,l}[t] \\
&= \sum_{i=1}^n \left[{\varphi_{i,t,l}^{-1}}'\big(\varphi_{i,t,l}(\bar{z}_{i,l}^W[t])\big)Q_{ij,l}^W[t] + \frac{{\varphi_{i,t,l}^{-1}}''(\varsigma_{ij,l}^W[t])}{2}Q_{ij,l}^W[t]^2\right]. \qquad \text{(B.8)}
\end{aligned}$$

Recall that the magnitudes of the first and second derivatives of $\varphi^{-1}$ (resp. $\varphi$) are assumed to be uniformly bounded by $L_-$ (resp. $L_+$). In addition, the quadratic remainder term $Q_{ij,l}^W[t]$ can be bounded by

$$\left|Q_{ij,l}^W[t]\right| \triangleq \left|\sum_{j=1}^n W_{ij}\frac{\varphi_{i,l}''(\xi_{ij,l}^W[t])}{2}(z_{j,l}[t] - \bar{z}_{i,l}^W[t])^2\right| \leq \frac{L_+}{2}sp(\mathbf{Z}_l[t])^2. \qquad \text{(B.9)}$$

Substituting (B.8) and (B.9) back into (B.7), we get

$$\begin{aligned}
&\|\Delta\bar{\mathbf{x}}[t] + \alpha\bar{\mathbf{g}}[t]\| \\
&\leq \alpha\|\mathbf{g}[t] - \bar{\mathbf{g}}[t]\| + \frac{1}{n}\sum_{l=1}^d \left[\frac{nL_+L_-}{2}sp(\mathbf{Z}_l[t])^2 + \frac{nL_+^2L_-}{8}sp(\mathbf{Z}_l[t])^4\right] \\
&\leq \alpha\|\mathbf{g}[t] - \bar{\mathbf{g}}[t]\| + \frac{L_+L_-}{2}sp(\mathbf{Z}[t])^2 + \frac{L_+^2L_-}{8}sp(\mathbf{Z}[t])^4 \\
&\leq \alpha\|\mathbf{g}[t] - \bar{\mathbf{g}}[t]\| + \frac{L_+L_-}{2}\big[sp(\mathbf{Z}[t]) + \alpha B\big]^2 + \frac{L_+^2L_-}{8}\big[sp(\mathbf{Z}[t]) + \alpha B\big]^4 \\
&\leq \alpha L_f\left[\rho^{\lfloor t/D\rfloor}\cdot sp(\mathbf{X}[0]) + \frac{2\alpha BL_+L_-}{1-\hat{\rho}}\right]
\end{aligned}$$

$$
\begin{aligned}
&\qquad + \frac{L_+L_-}{2}\big[sp(\mathbf{Z}[t]) + \alpha B\big]^2 + \frac{L_+^2L_-}{8}\big[sp(\mathbf{Z}[t]) + \alpha B\big]^4 \qquad \text{(by (B.4))}\\
&\le \alpha L_f \left[\rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0]) + \frac{2\alpha B L_+ L_-}{1-\hat{\rho}}\right]\\
&\qquad + \frac{L_+L_-}{2}\left[\rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0]) + \frac{2\alpha B L_+ L_-}{1-\hat{\rho}} + \alpha B\right]^2\\
&\qquad + \frac{L_+^2L_-}{8}\left[\rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0]) + \frac{2\alpha B L_+ L_-}{1-\hat{\rho}} + \alpha B\right]^4 \qquad \text{(by (B.2))}\\
&= O(\alpha^2)
\end{aligned}
\tag{B.10}
$$

when $t$ is large and $\alpha$ is small. This is because $\rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0]) \longrightarrow 0$ when $t \to \infty$, and all the variables other than $\alpha$ are just fixed constants; after removing the terms containing $\rho^{\lfloor t/D \rfloor} \cdot sp(\mathbf{X}[0])$, the remaining terms are either $O(\alpha^2)$ or $O(\alpha^4)$, and the former dominates.

Meanwhile, as $f_i$'s are strongly convex with constant $\nu$, it follows that $F$ is strongly convex with constant $n\nu$, so that

$$
F(\mathbf{x}^*) \ge F(\bar{\mathbf{x}}[t]) - \frac{1}{2n\nu}\|\nabla F(\bar{\mathbf{x}}[t])\|^2 = F(\bar{\mathbf{x}}[t]) - \frac{n}{2\nu}\|\bar{\mathbf{g}}[t]\|^2,
$$

which gives

$$
-\frac{n\alpha}{2}\|\bar{\mathbf{g}}[t]\|^2 \le -\alpha\nu\big[F(\bar{\mathbf{x}}[t]) - F(\mathbf{x}^*)\big]. \tag{B.11}
$$

Substituting (B.10) and (B.11) into (B.6) gives

$$
\begin{aligned}
F(\bar{\mathbf{x}}[t+1]) - F(\mathbf{x}^*) &\le (1-\alpha\nu)\big[F(\bar{\mathbf{x}}[t]) - F(\mathbf{x}^*)\big] + \frac{n}{2\alpha}\big[O(\alpha^2)\big]^2\\
&= \bar{\rho}\big[F(\bar{\mathbf{x}}[t]) - F(\mathbf{x}^*)\big] + O(\alpha^3)\\
&\le \bar{\rho}^t\big[F(\bar{\mathbf{x}}[0]) - F(\mathbf{x}^*)\big] + \sum_{s=1}^{t}\bar{\rho}^{t-s}O(\alpha^3)\\
&\le \bar{\rho}^t\big[F(\bar{\mathbf{x}}[0]) - F(\mathbf{x}^*)\big] + \frac{1}{1-\bar{\rho}}O(\alpha^3)\\
&= \bar{\rho}^t\big[F(\bar{\mathbf{x}}[0]) - F(\mathbf{x}^*)\big] + \frac{n}{\alpha\nu}O(\alpha^3)\\
&= \bar{\rho}^t\big[F(\bar{\mathbf{x}}[0]) - F(\mathbf{x}^*)\big] + O(\alpha^2),
\end{aligned}
\tag{B.12}
$$

where the factor $\bar{\rho} \triangleq 1 - \alpha\nu$ is in $[0, 1)$ by taking any $\alpha < \frac{1}{L_f}$ so that $\alpha\nu < \frac{\nu}{L_f} \in (0, 1]$. The second inequality follows from repeatedly applying the inequality backwards from $t$ to 0. In addition, the strong convexity of $F$ implies

$$\frac{n\nu}{2}\|\bar{\mathbf{x}}[t+1] - \mathbf{x}^*\|^2 \leq F(\bar{\mathbf{x}}[t+1]) - F(\mathbf{x}^*). \tag{B.13}$$

■

**Proof of Theorem 3.10:** Notice that any point in $\delta \circ co(\{\mathbf{z}_j[t] : j \in Nb(i)\})$ can be written as the convex combination of $\{\mathbf{z}_j[t] : j \in Nb(i)\}$ with positive weights. Hence, given any choice of $\mathbf{x}_i[n+1]$ we can find a distribution $\{W^x_{ij}[t]\}_{j=1}^n$ such that $\mathbf{x}_i[t+1] = \sum_{j=1}^n W^x_{ij}[t]\mathbf{z}_j[t]$, where $W^x_{ij}[t] \geq \frac{1-\delta}{|N(i)|} \geq \frac{1-\delta}{n} > 0$ for $j \in Nb(i)$ and $W^x_{ij}[t] = 0$ for $j \notin Nb(i)$. Simply speaking, if we group these weights to form a matrix $\mathbf{W}^x[t]$, it satisfies the constraints described in Section 2.2.1 except it may not be column stochastic.

It is a standard result [129] that there exists a distribution $\{\tilde{z}_i\}_{i=1}^n$ such that

$$\lim_{t\to\infty} \mathbf{W}^x[t]\mathbf{W}^x[t-1]\cdots\mathbf{W}^x[1] = \mathbf{1}_n\tilde{\mathbf{z}}^\top.$$

From Lemma B.2, we know that applying $\mathbf{W}^x[t]$ to any column vector $\mathbf{v}$, the range of the vector $sp(\mathbf{v})$ will decrease at least by a factor of $1 - \frac{2(1-\delta)}{n}$ if $\mathbf{W}^x[t]$ is row stochastic. Since all the matrices $\mathbf{W}^x[2], \mathbf{W}^x[3], \ldots$ are row stochastic, each column of $\mathbf{W}^x[1]$ will align with $\mathbf{1}_n$ when multiplied by $\Pi_{s=2}^{\infty}\mathbf{W}^x[s]$ as the range of the column converges to 0 while the entries in the column asymptotically agree. The overall product is hence in the form of $\mathbf{1}_n\tilde{\mathbf{z}}^\top$. Note that when all $\mathbf{W}^x[t]$'s are doubly stochastic, the distribution $\tilde{\mathbf{z}}$ is uniform.

The original proof presented in [80] can be directly used here with the following revisions: $\bar{\mathbf{x}}[t]$ replaced by $\mathbf{x}_c[t] = \sum_{i=1}^n \tilde{z}_i\mathbf{x}_i[t]$, and all $\frac{1}{n}\mathbf{1}\mathbf{1}^\top$ related to $\mathbf{x}$ including in the definition of $\mathbb{J}$ should be replaced by $\mathbf{1}\tilde{\mathbf{z}}^\top$. ■

# APPENDIX C

# Proofs for Chapter IV

## C.1 Missing Proofs in Section 4.3.2

The following lemma shows that given the FPS, what actions the chosen prescription maps to for other FPSs does not affect the next step statistics.

**Lemma C.1:** *Let* $h_t^0 \in \Omega(H_t^0)$*,* $h \in \Omega(H_t^{1:N})$*,* $\gamma \in \Omega(\Gamma_t)$*, and* $a = \gamma(h) \in \Omega(A_t)$*. Then*

$$\mathbb{P}(S_{t+1}, H_{t+1}^{1:N} | H_t^0 = h_t^0, H_t^{1:N} = h, \Gamma_t = \gamma) = \mathbb{P}(S_{t+1}, H_{t+1}^{1:N} | H_t^0 = h_t^0, H_t^{1:N} = h, A_t = a).$$

**Proof:** We will omit specifying the original random variables when their realizations are given in the proof.

$$\begin{aligned}
&\mathbb{P}(S_{t+1}, H_{t+1}^{1:N} | h_t^0, h, \gamma) = \mathbb{P}(S_{t+1}, H_{t+1}^{1:N} | h_t^0, h, \gamma, a) \\
&= \sum_{s_t} \mathbb{P}(s_t | h_t^0, h, \gamma, a) \cdot \mathbb{P}(S_{t+1}, H_{t+1}^{1:N} | h_t^0, h, \gamma, a, s_t) \\
&= \sum_{s_t} \mathbb{P}(s_t | h_t^0, h, a) \cdot \mathbb{P}(S_{t+1} | h_t^0, h, \gamma, a, s_t) \cdot \mathbb{P}(H_{t+1}^{1:N} | h_t^0, h, \gamma, a, s_t, S_{t+1}) \quad (\gamma \text{ is after } s_t) \\
&= \sum_{s_t} \mathbb{P}(s_t | h_t^0, h, a) \cdot \mathbb{P}(S_{t+1} | a, s_t) \cdot \mathbb{P}(H_{t+1}^{1:N} | h_t^0, h, \gamma, a, s_t, S_{t+1}) \\
&\hspace{22em} (\mathbb{P}_T \text{ specifies } S_{t+1} \text{ given } S_t \text{ and } A_t)
\end{aligned}$$

$$= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h, a) \cdot \mathbb{P}(S_{t+1}|a, s_t) \cdot \mathbb{P}(O_{t+1}^{1:N}|h_t^0, h, \gamma, a, s_t, S_{t+1})$$

$$(H_{t+1}^{1:N} = (H_t^{1:N}, A_t, O_{t+1}^{1:N}))$$

$$= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h, a) \cdot \mathbb{P}(S_{t+1}|a, s_t) \cdot \mathbb{P}(O_{t+1}^{1:N}|S_{t+1}) \qquad (\mathbb{P}_O \text{ specifies } O_{t+1}^{1:N} \text{ given } S_{t+1})$$

$$= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h, a) \cdot \mathbb{P}(S_{t+1}, H_{t+1}^{1:N}|h_t^0, h, a, s_t)$$

$$= \mathbb{P}(S_{t+1}, H_{t+1}^{1:N}|h_t^0, h, a). \tag{C.1}$$

■

**Proof of Lemma 4.11:** The proof for the instantaneous part is straightforward as $S_t$ is irrelevant to the choice of $\Gamma_t$

$$\mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N}))|h_t^0, h, \gamma_1\right] = \sum_{s_t} \mathbb{P}(s_t|h_t^0, h, \gamma_1) R_t(s_t, \gamma_1(h))$$

$$= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h, \gamma_1(h)) R_t(s_t, a)$$

$$= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h) R_t(s_t, a) \qquad (\gamma_1 \text{ and } \gamma_2 \text{ are exogenously given})$$

$$= \mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N}))|h_t^0, h, \gamma_2\right]. \qquad \text{(by symmetry)}$$

To show equality for the continuation part, we first define the following policy for all $\tau = t+1, \ldots, T$:

$$d'_\tau(h_\tau^0) = \begin{cases} d^*_\tau(h_t^0, \gamma_1, h_{t+1:\tau}^0) & \text{if } h_\tau^0 = (h_t^0, \gamma_2, h_{t+1:\tau}^0) \quad \forall\ h_{t+1:\tau}^0, \\ d^*_\tau(h_\tau^0) & \text{otherwise,} \end{cases}$$

where $d^*_\tau$ is an optimal policy at time step $\tau$. Also, we have $h_{t+1:\tau}^0 = o_{t+1}^0$ when $\tau = t+1$, and $h_{t+1:\tau}^0 = (o_{t+1}^0, \gamma_{t+1}, \ldots, o_\tau^0)$ when $\tau > t+1$, so that the entire $(h_t^0, \gamma_1, h_{t+1:\tau}^0) \in \Omega(H_\tau^0)$. This policy performs the optimal policy at all times, except when $\gamma_2$ is chosen at time $t$ it will mimic what the optimal policy would have done if $\gamma_1$ was chosen instead; owing to perfect recall, future prescriptions can depend on past ones. Then

$$
\begin{aligned}
&\mathbb{E}\left[V^S_{t+1}(H^0_{t+1}, H^{1:N}_{t+1})|h^0_t, h, \gamma_1\right] = \mathbb{E}\left[\sum_{\tau=t+1}^{T} R_\tau(S_\tau, A_\tau)\middle| h^0_t, h, \gamma_1, d^*_{t+1:T}\right] \\
&= \mathbb{E}\left[\sum_{\tau=t+1}^{T} R_\tau(S_\tau, A_\tau)\middle| h^0_t, h, \gamma_1, d'_{t+1:T}\right] \overset{(*)}{=} \mathbb{E}\left[\sum_{\tau=t+1}^{T} R_\tau(S_\tau, A_\tau)\middle| h^0_t, h, \gamma_2, d'_{t+1:T}\right] \\
&\leq \mathbb{E}\left[\sum_{\tau=t+1}^{T} R_\tau(S_\tau, A_\tau)\middle| h^0_t, h, \gamma_2, d^*_{t+1:T}\right] = \mathbb{E}\left[V^S_{t+1}(H^0_{t+1}, H^{1:N}_{t+1})|h^0_t, h, \gamma_2\right],
\end{aligned}
$$

where the inequality holds as $d'_{t+1:T}$ is not an optimal choice from the current history. By symmetry, the inequality implies that

$$
\mathbb{E}\left[V^S_{t+1}(H^0_{t+1}, H^{1:N}_{t+1})|h^0_t, h, \gamma_1\right] = \mathbb{E}\left[V^S_{t+1}(H^0_{t+1}, H^{1:N}_{t+1})|h^0_t, h, \gamma_2\right].
$$

The equality labeled by $(*)$ follows from the fact that under the policy $d'_{t+1:T}$, choosing $\gamma_1$ and $\gamma_2$ will generate the exact future statistics. We will show this in the following. We first prove the following claim using mathematical induction.

**Claim:** for all $\tau = t+1, \ldots, T$, we have

$$
\mathbb{P}(S_{t+1:\tau}, O^{0:N}_{t+1:\tau}, A_{t+1:\tau}|h^0_t, h, \gamma_1, a, d'_{t+1:T}) = \mathbb{P}(S_{t+1:\tau}, O^{0:N}_{t+1:\tau}, A_{t+1:\tau}|h^0_t, h, \gamma_2, a, d'_{t+1:T}).
$$

<u>Base case</u>: the claim holds for $\tau = t+1$.

$$
\begin{aligned}
&\mathbb{P}(S_{t+1}, O^{0:N}_{t+1}, A_{t+1}|h^0_t, h, \gamma_1, a, d'_{t+1:T}) \\
&= \sum_{s_t} \mathbb{P}(s_t|h^0_t, h, \gamma_1, a, d'_{t+1:T}) \cdot \mathbb{P}(S_{t+1}, O^{0:N}_{t+1}, A_{t+1}|h^0_t, h, \gamma_1, a, s_t, d'_{t+1:T}) \\
&= \sum_{s_t} \mathbb{P}(s_t|h^0_t, h) \cdot \mathbb{P}(S_{t+1}, O^{0:N}_{t+1}, A_{t+1}|h^0_t, h, \gamma_1, a, s_t, d'_{t+1}) \\
&\hspace{20em} (s_t \text{ is independent of a given } \Gamma_t) \\
&= \sum_{s_t} \mathbb{P}(s_t|h^0_t, h) \cdot \mathbb{P}(S_{t+1}|h^0_t, h, \gamma_1, a, s_t, d'_{t+1}) \cdot \mathbb{P}(O^{0:N}_{t+1}, A_{t+1}|h^0_t, h, \gamma_1, a, s_t, d'_{t+1}, S_{t+1}) \\
&= \sum_{s_t} \mathbb{P}(s_t|h^0_t, h) \cdot \mathbb{P}(S_{t+1}|s_t, a) \cdot \mathbb{P}(O^{0:N}_{t+1}, A_{t+1}|h^0_t, h, \gamma_1, a, s_t, d'_{t+1}, S_{t+1}) \\
&\hspace{20em} (\mathbb{P}_T \text{ specifies } S_{t+1} \text{ given } S_t \text{ and } A_t)
\end{aligned}
$$

$$\begin{aligned}
&= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h) \cdot \mathbb{P}(S_{t+1}|s_t, a) \cdot \mathbb{P}(O_{t+1}^{0:N}|h_t^0, h, \gamma_1, a, s_t, d'_{t+1}, S_{t+1}) \\
&\qquad \cdot \mathbb{P}(A_{t+1}|h_t^0, h, \gamma_1, a, s_t, d'_{t+1}, S_{t+1}, O_{t+1}^{0:N}) \\
&= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h) \cdot \mathbb{P}(S_{t+1}|s_t, a) \cdot \mathbb{P}(O_{t+1}^{0:N}|S_{t+1}) \cdot \mathbb{P}(A_{t+1}|h_t^0, h, \gamma_1, a, s_t, d'_{t+1}, S_{t+1}, O_{t+1}^{0:N}) \\
&\hspace{20em} (\mathbb{P}_O \text{ specifies } O_{t+1}^{0:N} \text{ given } S_{t+1}) \\
&= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h) \cdot \mathbb{P}(S_{t+1}|s_t, a) \cdot \mathbb{P}(O_{t+1}^{0:N}|S_{t+1}) \\
&\qquad\qquad \cdot \mathbb{I}\left\{A_{t+1} = d'_{t+1}(h_0, \gamma_1, O_{t+1}^0)(h, a, O_{t+1}^{1:N})\right\} \\
&= \sum_{s_t} \mathbb{P}(s_t|h_t^0, h) \cdot \mathbb{P}(S_{t+1}|s_t, a) \cdot \mathbb{P}(O_{t+1}^{0:N}|S_{t+1}) \\
&\qquad\qquad \cdot \mathbb{I}\left\{A_{t+1} = d'_{t+1}(h_0, \gamma_2, O_{t+1}^0)(h, a, O_{t+1}^{1:N})\right\} && (\text{definition of } d'_{t+1}) \\
&= \mathbb{P}(S_{t+1}, O_{t+1}^{0:N}, A_{t+1}|h_t^0, h, \gamma_2, a, d'_{t+1:T}). && (\text{symmetric argument})
\end{aligned}$$

Induction step: assuming the claim holds for $\tau$, we show it holds for $\tau + 1$ as well.

$$\begin{aligned}
&\mathbb{P}(S_{t+1:\tau+1}, O_{t+1:\tau+1}^{0:N}, A_{t+1:\tau+1}|h_t^0, h, \gamma_1, a, d'_{t+1:T}) \\
&= \mathbb{P}(S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}|h_t^0, h, \gamma_1, a, d'_{t+1:T}) \\
&\qquad \cdot \mathbb{P}(S_{\tau+1}, O_{\tau+1}^{0:N}, A_{\tau+1}|h_t^0, h, \gamma_1, a, d'_{t+1:T}, S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}) \\
&= \mathbb{P}(S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}|h_t^0, h, \gamma_2, a, d'_{t+1:T}) \\
&\qquad \cdot \mathbb{P}(S_{\tau+1}, O_{\tau+1}^{0:N}, A_{\tau+1}|h_t^0, h, \gamma_1, a, d'_{t+1:T}, S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}) \\
&\hspace{22em} (\text{induction hypothesis}) \\
&= \mathbb{P}(S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}|h_t^0, h, \gamma_2, a, d'_{t+1:T}) \cdot \mathbb{P}(S_{\tau+1}|S_\tau, A_\tau) \cdot \mathbb{P}(O_{\tau+1}^{0:N}|S_{\tau+1}) \\
&\quad \cdot \mathbb{I}\left\{A_{\tau+1} = d'_{\tau+1}\left(h_t^0, \gamma_1, O_{t+1}^0, d'_{t+1}(h_t^0, \gamma_1, O_{t+1}^0), O_{t+2}^0, \ldots, O_{\tau+1}^0\right)(h, a, O_{t+1}^{1:N}, A_{t+1}, \ldots, O_{\tau+1}^{1:N})\right\} \\
&\overset{(\dagger)}{=} \mathbb{P}(S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}|h_t^0, h, \gamma_2, a, d'_{t+1:T}) \cdot \mathbb{P}(S_{\tau+1}|S_\tau, A_\tau) \cdot \mathbb{P}(O_{\tau+1}^{0:N}|S_{\tau+1}) \\
&\quad \cdot \mathbb{I}\left\{A_{\tau+1} = d'_{\tau+1}\left(h_t^0, \gamma_2, O_{t+1}^0, d'_{t+1}(h_t^0, \gamma_2, O_{t+1}^0), O_{t+2}^0, \ldots, O_{\tau+1}^0\right)(h, a, O_{t+1}^{1:N}, A_{t+1}, \ldots, O_{\tau+1}^{1:N})\right\} \\
&= \mathbb{P}(S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}|h_t^0, h, \gamma_2, a, d'_{t+1:T}) \\
&\qquad \cdot \mathbb{P}(S_{\tau+1}, O_{\tau+1}^{0:N}, A_{\tau+1}|h_t^0, h, \gamma_2, a, d'_{t+1:T}, S_{t+1:\tau}, O_{t+1:\tau}^{0:N}, A_{t+1:\tau}) \\
&= \mathbb{P}(S_{t+1:\tau+1}, O_{t+1:\tau+1}^{0:N}, A_{t+1:\tau+1}|h_t^0, h, \gamma_2, a, d'_{t+1:T}),
\end{aligned}$$

where the equality in (†) holds due to the definition of policy $d'$. Note that with the CI-based approach, a generic policy $d_t$ first maps an FCS $H_t^0$ to a prescription $\Gamma_t$, which in term maps an FPS $H_t^{1:N}$ to an action $A_t$; therefore, $d_t(H_t^0)(H_t^{1:N}) = \Gamma_t(H_t^{1:N}) = A_t$

refers to the final action $A_t$ under the policy $d_t$ and the supervisor's state $(H_t^0, H_t^{1:N})$. The claim implies that $\mathbb{P}(S_\tau, A_\tau | h_t^0, h, \gamma_1, d'_{t+1:T}) = \mathbb{P}(S_\tau, A_\tau | h_t^0, h, \gamma_2, d'_{t+1:T})$ for all $\tau = t+1, \ldots, T$, i.e. conditioning on $h_t^0, h, d'_{t+1:T}$, the distribution of $(S_\tau, A_\tau)$ is exactly the same given $\gamma_1$ or $\gamma_2$; and $(S_\tau, A_\tau)$ where $\tau = t+1, \ldots, T$ is what the expectations on both sides of $(*)$ are taken on. ■

**Proof of Lemma 4.12:** We proceed the proof by mathematical induction. The expectation in the $Q$-function expression consists of two parts – the instantaneous part $(R_t)$ and the continuation part $(V_{t+1})$. For the instantaneous part as well as the base case $t = T$, we have

$$\begin{aligned}
&\mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N})) | h_t^0, h_1, \gamma\right] = \sum_{s_t} \mathbb{P}(s_t | h_t^0, h_1, \gamma) R_t(s_t, \gamma(h_1)) \\
&= \sum_{s_t} \mathbb{P}(s_t | h_t^0, h_1, \gamma, a) R_t(s_t, a) && (a = \gamma(h)) \\
&= \sum_{s_t} \mathbb{P}(s_t | h_t^0, h_1, a) R_t(s_t, a) && (s_t \text{ is before } \gamma) \\
&= \mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N})) | h_t^0, h_1, a\right] \\
&= \mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N})) | h_t^0, z, a\right] && (\text{(SPS2) with } z = \vartheta_t(h_t^0, h_1)) \\
&= \mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N})) | h_t^0, h_2, a\right] && (\text{(SPS2) with } z = \vartheta_t(h_t^0, h_2)) \\
&= \mathbb{E}\left[R_t(S_t, \Gamma_t(H_t^{1:N})) | h_t^0, h_2, \gamma\right].
\end{aligned}$$

For the continuation part, we have

$$\begin{aligned}
&\mathbb{E}\left[V_{t+1}^S((H_t^0, \Gamma_t, O_{t+1}^0), H_{t+1}^{1:N}) | h_t^0, h_1, \gamma\right] \\
&= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N} | h_t^0, h_1, \gamma) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \\
&= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(o_{t+1}^{0:N}, s_{t+1} | h_t^0, h_1, \gamma) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \\
&= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(o_{t+1}^{0:N} | h_t^0, h_1, \gamma, s_{t+1}) \cdot \mathbb{P}(s_{t+1} | h_t^0, h_1, \gamma) \cdot V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \\
&= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(o_{t+1}^{0:N} | s_{t+1}) \cdot \mathbb{P}(s_{t+1} | h_t^0, h_1, \gamma) \cdot V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \\
&&& (\mathbb{P}_O \text{ specifies } O_{t+1}^{0:N} \text{ given } S_{t+1})
\end{aligned}$$

$$= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(o_{t+1}^{0:N}|s_{t+1}) \cdot \mathbb{P}(s_{t+1}|h_t^0, h_1, a) \cdot V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N}))$$

$$\text{(Lemma C.1)}$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_1, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N}))$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, z, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \quad \text{((SPS3) with } z = \vartheta_t(h_t^0, h_1))$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_2, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \quad \text{((SPS3) with } z = \vartheta_t(h_t^0, h_2))$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_2, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N})) \quad \text{((SPS1) and Corollary 4.13)}$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_2, \gamma) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N}))) \quad \text{(symmetric arguement)}$$

$$= \mathbb{E}\left[V_{t+1}^S((H_t^0, \Gamma_t, O_{t+1}^0), H_{t+1}^{1:N})|h_t^0, h_2, \gamma\right].$$

Note that since $z$ could be updated recursively by (SPS1), the two SPSs $(h_1, a, o_{t+1}^{1:N})$ and $(h_2, a, o_{t+1}^{1:N})$ in $\Omega(H_{t+1}^{1:N})$ should be mapped to the same $z_{t+1}^{1:N} \in \Omega(Z_{t+1}^{1:N})$ under the same FCS $(h_t^0, \gamma, o_{t+1}^0)$ as they are updated from the same $z = \vartheta_t(h_t^0, h_1) = \vartheta_t(h_t^0, h_2) \in \Omega(Z_t^{1:N})$. Thus, from Corollary 4.13 we have

$$V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1})) = V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1})),$$

so that the third last equality follows.

In the proof we use the result of Corollary 4.13, which in turn derives from Lemma 4.12 which we are proving. However, this is not a tautology. The logic flow is the mathematical induction of Lemma 4.12. The base case $t = T$ only contains the instantaneous part (since $V_{t+1} \triangleq 0$) and can be established directly. Then in the induction step, the induction hypothesis of Lemma 4.12 holding true for $t + 1$ implies that Corollary 4.13 is also true for $t + 1$, which is then used to show the continuation part of Lemma 4.12's claim for $t$. ■

**Proof of Corollary 4.13:** Assume the optimal prescription $\gamma^*$ prescribes different actions on the two FPSs $h_1, h_2 \in \Omega(H_t^{1:N})$, so that $\gamma^*(h_1) = a_1$ and $\gamma^*(h_2) = a_2$ where $a_1 \neq a_2$; otherwise, the claim directly follows by Lemma 4.12. Also, define

$\gamma', \gamma'' \in \Omega(\Gamma_t)$ by

$$\gamma'(h) = \begin{cases} a_1 & \text{if } h = h_2, \\ \gamma^*(h) & \text{otherwise,} \end{cases} \qquad \gamma''(h) = \begin{cases} a_2 & \text{if } h = h_1, \\ \gamma^*(h) & \text{otherwise.} \end{cases}$$

Let $\upsilon_1 = Q_t^S(h_t^0, h_1, \gamma^*)$ and $\upsilon_2 = Q_t^S(h_t^0, h_2, \gamma^*)$. Using (4.11), we can expand $Q_t(h_t^0, \gamma^*)$ to (throughout the proof only admissible FPSs are considered – we omit specifications of the constraint for cleaner notations)

$$\begin{aligned} Q_t(h_t^0, \gamma^*) &= \sum_h \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) \\ &= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma^*) + \mathbb{P}(h_2|h_t^0) Q_t^S(h_t^0, h_2, \gamma^*) \\ &= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0)\upsilon_1 + \mathbb{P}(h_2|h_t^0)\upsilon_2. \end{aligned}$$

Using the same definition, we can also expand $Q_t(h_t, \gamma')$ to

$$\begin{aligned} Q_t(h_t^0, \gamma') &= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma') + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma') + \mathbb{P}(h_2|h_t^0) Q_t^S(h_t^0, h_2, \gamma') \\ &= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma') + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma') + \mathbb{P}(h_2|h_t^0) Q_t^S(h_t^0, h_1, \gamma') \\ &\qquad \text{(Lemma 4.12)} \\ &= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma^*) + \mathbb{P}(h_2|h_t^0) Q_t^S(h_t^0, h_1, \gamma^*) \\ &\qquad \text{(Lemma 4.11)} \\ &= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0)\upsilon_1 + \mathbb{P}(h_2|h_t^0)\upsilon_1; \end{aligned}$$

similarly, we have the following by symmetry

$$Q_t(h_t^0, \gamma'') = \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0)\upsilon_2 + \mathbb{P}(h_2|h_t^0)\upsilon_2.$$

It is evident that $Q_t(h_t^0, \gamma^*)$ is a convex combination of $Q_t(h_t^0, \gamma')$ and $Q_t(h_t^0, \gamma'')$. Hence, at least one of $Q_t(h_t^0, \gamma')$ and $Q_t(h_t^0, \gamma'')$ is larger than or equal to $Q_t(h_t^0, \gamma^*)$. Suppose $Q_t(h_t^0, \gamma') \geq Q_t(h_t^0, \gamma^*)$; but $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h_t^0, \gamma)$, so they must be equal, which implies $\upsilon_1 = \upsilon_2$. Notice that all three quantities $Q_t(h_t^0, \gamma^*)$, $Q_t(h_t^0, \gamma')$, and $Q_t(h_t^0, \gamma'')$ will be equal. ■

**Proof of Proposition 4.14:** We first show the proof for the instantaneous part

$$\begin{aligned}
\mathbb{E}\left[R_t(S_t, A_t)|h_1^0, \gamma_{\lambda,h_1^0}\right] &= \sum_{s_t, z_t^{1:N}} \mathbb{P}(s_t, z_t^{1:N}|h_1^0, \gamma_{\lambda,h_1^0}) R_t(s_t, \lambda(z_t^{1:N})) \\
&= \sum_{s_t, z_t^{1:N}} \mathbb{P}(s_t, z_t^{1:N}|h_1^0) R_t(s_t, \lambda(z_t^{1:N})) && (\lambda \text{ is given exogenously}) \\
&= \sum_{s_t, z_t^{1:N}} \mathbb{P}(s_t, z_t^{1:N}|h_2^0) R_t(s_t, \lambda(z_t^{1:N})) && (\text{assumption}) \\
&= \mathbb{E}\left[R_t(S_t, A_t)|h_2^0, \gamma_{\lambda,h_2^0}\right].
\end{aligned}$$

We finish the rest of the proof by mathematical induction. Note that the base case $t = T$ is already proved above as the values at $t = T$ are solely instantaneous. Now we only have to show the induction step, assuming the statement for $t + 1$ is true and proving it holds for $t$ as well, for the continuation part. Note that the two FCSs $h_1^0$ and $h_2^0$ observe the same distribution of $(S_t, Z_t^{1:N}, O_{t+1}^0)$ when the same SPS-based prescription $\lambda$ is picked

$$\begin{aligned}
&\mathbb{P}(S_t, Z_t^{1:N}, O_{t+1}^0|h_1^0, \gamma_{\lambda,h_1^0}) \\
&= \sum_{s_{t+1}} \mathbb{P}(S_t, Z_t^{1:N}|h_1^0, \gamma_{\lambda,h_1^0}) \cdot \mathbb{P}(s_{t+1}|h_1^0, \gamma_{\lambda,h_1^0}, S_t, Z_t^{1:N}) \cdot \mathbb{P}(O_{t+1}^0|h_1^0, \gamma_{\lambda,h_1^0}, S_t, Z_t^{1:N}, s_{t+1}) \\
&= \sum_{s_{t+1}} \mathbb{P}(S_t, Z_t^{1:N}|h_1^0) \cdot \mathbb{P}(s_{t+1}|h_1^0, \gamma_{\lambda,h_1^0}, S_t, Z_t^{1:N}) \cdot \mathbb{P}(O_{t+1}^0|h_1^0, \gamma_{\lambda,h_1^0}, S_t, Z_t^{1:N}, s_{t+1}) \\
&\hspace{25em} (\lambda \text{ is given exogenously}) \\
&= \sum_{s_{t+1}} \mathbb{P}(S_t, Z_t^{1:N}|h_1^0) \cdot \mathbb{P}(s_{t+1}|S_t, \lambda(Z_t^{1:N})) \cdot \mathbb{P}(O_{t+1}^0|s_{t+1}) \quad (\text{specified by } \mathbb{P}_T \text{ and } \mathbb{P}_O) \\
&= \sum_{s_{t+1}} \mathbb{P}(S_t, Z_t^{1:N}|h_2^0) \cdot \mathbb{P}(s_{t+1}|S_t, \lambda(Z_t^{1:N})) \cdot \mathbb{P}(O_{t+1}^0|s_{t+1}) \quad (\text{assumption}) \\
&= \mathbb{P}(S_t, Z_t^{1:N}, O_{t+1}^0|h_2^0, \gamma_{\lambda,h_2^0}).
\end{aligned}$$

Moreover, for any fixed $o_{t+1}^0 \in \Omega(O_{t+1}^0)$, $(h_1^0, \gamma_{\lambda,h_1^0}, o_{t+1}^0)$ and $(h_2^0, \gamma_{\lambda,h_2^0}, o_{t+1}^0)$ will be two FCSs corresponding to the same BCS, since

$$\begin{aligned}
&\mathbb{P}(S_{t+1}, Z^{1:N}_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}, o^0_{t+1}) = \frac{\mathbb{P}(S_{t+1}, Z^{1:N}_{t+1}, o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1})}{\mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1})} \\
&= \frac{1}{\mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1})} \sum_{s_t, z^{1:N}_t} \mathbb{P}(s_t, z^{1:N}_t|h^0_1, \gamma_{\lambda,h^0_1}) \cdot \mathbb{P}(S_{t+1}, Z^{1:N}_{t+1}, o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}, s_t, z^{1:N}_t) \\
&= \frac{1}{\mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1})} \sum_{s_t, z^{1:N}_t} \mathbb{P}(s_t, z^{1:N}_t|h^0_1, \gamma_{\lambda,h^0_1}) \cdot \mathbb{P}(S_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}, s_t, z^{1:N}_t) \\
&\quad \cdot \mathbb{P}(Z^{1:N}_{t+1}, o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}, s_t, z^{1:N}_t, S_{t+1}) \\
&= \frac{1}{\mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1})} \sum_{s_t, z^{1:N}_t} \mathbb{P}(s_t, z^{1:N}_t|h^0_1, \gamma_{\lambda,h^0_1}) \cdot \mathbb{P}(S_{t+1}|s_t, \lambda(z^{1:N}_t)) \\
&\quad \cdot \mathbb{P}(Z^{1:N}_{t+1}, o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}, s_t, z_t, S_{t+1}) \qquad \text{(specified by } \mathbb{P}_T\text{)} \\
&= \frac{1}{\mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1})} \sum_{s_t, z^{1:N}_t} \mathbb{P}(s_t, z^{1:N}_t|h^0_1, \gamma_{\lambda,h^0_1}) \cdot \mathbb{P}(S_{t+1}|s_t, \lambda(z^{1:N}_t)) \\
&\quad \cdot \sum_{o_{t+1}} \mathbb{P}(o^{0:N}_{t+1}|S_{t+1}) \cdot \mathbb{I}\{Z^{1:N}_{t+1} = \phi_{t+1}(z^{1:N}_t, \lambda(z_t), o^{0:N}_{t+1})\} \\
&\qquad \text{(specified by } \mathbb{P}_O \text{ and (SPS1))} \\
&= \frac{1}{\mathbb{P}(o^0_{t+1}|h^0_2, \gamma_{\lambda,h^0_2})} \sum_{s_t, z^{1:N}_t} \mathbb{P}(s_t, z^{1:N}_t|h^0_2, \gamma_{\lambda,h^0_2}) \cdot \mathbb{P}(S_{t+1}|s_t, \lambda(z^{1:N}_t)) \\
&\quad \cdot \sum_{o_{t+1}} \mathbb{P}(o^{0:N}_{t+1}|S_{t+1}) \cdot \mathbb{I}\{Z^{1:N}_{t+1} = \phi_{t+1}(z^{1:N}_t, \lambda(z_t), o^{0:N}_{t+1})\} \qquad \text{(induction hypothesis)} \\
&= \mathbb{P}(S_{t+1}, Z^{1:N}_{t+1}|h^0_2, \gamma_{\lambda,h^0_2}, o^0_{t+1}). \qquad \text{(symmetric argument)}
\end{aligned}$$

Therefore, for the continuation part we have

$$\begin{aligned}
&\mathbb{E}\left[V_{t+1}((H^0_t, \Gamma_t, O^0_{t+1}))|h^0_1, \gamma_{\lambda,h^0_1}\right] = \sum_{o^0_{t+1}} \mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}) V_{t+1}((h^0_1, \gamma_{\lambda,h^0_1}, o^0_{t+1})) \\
&= \sum_{o^0_{t+1}} \mathbb{P}(o^0_{t+1}|h^0_1, \gamma_{\lambda,h^0_1}) \cdot \max_{\lambda' \in \Omega(\Lambda_{t+1})} Q_{t+1}\left((h^0_1, \gamma_{\lambda,h^0_1}, o^0_{t+1}), \gamma_{\lambda',(h^0_1,\gamma_{\lambda,h^0_1},o^0_{t+1})})\right) \\
&= \sum_{o^0_{t+1}} \mathbb{P}(o^0_{t+1}|h^0_2, \gamma_{\lambda,h^0_2}) \cdot \max_{\lambda' \in \Omega(\Lambda_{t+1})} Q_{t+1}\left((h^0_2, \gamma_{\lambda,h^0_2}, o^0_{t+1}), \gamma_{\lambda',(h^0_2,\gamma_{\lambda,h^0_2},o^0_{t+1})})\right) \\
&\qquad \text{(induction hypothesis and } \widetilde{\Pi}_{t+1}((h^0_1, \gamma_{\lambda,h^0_1}, o^0_{t+1})) = \widetilde{\Pi}_{t+1}((h^0_2, \gamma_{\lambda,h^0_2}, o^0_{t+1}))\text{)} \\
&= \sum_{o^0_{t+1}} \mathbb{P}(o^0_{t+1}|h^0_2, \gamma_{\lambda,h^0_2}) V_{t+1}((h^0_2, \gamma_{\lambda,h^0_2}, o^0_{t+1})) = \mathbb{E}\left[V_{t+1}((H^0_t, \Gamma_t, O^0_{t+1}))|h^0_2, \gamma_{\lambda,h^0_2}\right].
\end{aligned}$$

■

## C.2 Missing Proofs in Section 4.3.3

**Proof of Proposition 4.17:** We proceed the proof again by mathematical induction. The instantaneous part as well as the base case $t = T$ trivially follow by (SCS2)

$$\begin{aligned}
&\mathbb{E}\left[R_t(S_t, A_t)|h_1^0, \gamma_{\lambda,h_1^0}\right] = \mathbb{E}\left[R_t(S_t, A_t)|h_1^0, \lambda\right] \\
&= \mathbb{E}\left[R_t(S_t, A_t)|\vartheta_t^0(h_1^0), \lambda\right] && \text{((SCS2))} \\
&= \mathbb{E}\left[R_t(S_t, A_t)|\vartheta_t^0(h_2^0), \lambda\right] && \text{(assumption)} \\
&= \mathbb{E}\left[R_t(S_t, A_t)|h_2^0, \lambda\right] && \text{((SCS2))} \\
&= \mathbb{E}\left[R_t(S_t, A_t)|h_2^0, \gamma_{\lambda,h_2^0}\right].
\end{aligned}$$

For the continuation part in the induction step, we have

$$\begin{aligned}
&\mathbb{E}\left[V_{t+1}((H_t^0, \Gamma_t, O_{t+1}^0))|h_1^0, \gamma_{\lambda,h_1^0}\right] = \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_1^0, \gamma_{\lambda,h_1^0}) V_{t+1}((h_1^0, \gamma_{\lambda,h_1^0}, o_{t+1}^0)) \\
&= \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_1^0, \lambda) \cdot \max_{\lambda' \in \Omega(\Lambda_{t+1})} Q_{t+1}\left((h_1^0, \gamma_{\lambda,h_1^0}, o_{t+1}^0), \gamma_{\lambda',(h_1^0,\gamma_{\lambda,h_1^0},o_{t+1}^0)}\right) \\
&\overset{(\sharp)}{=} \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_1^0, \lambda) \cdot \max_{\lambda' \in \Omega(\Lambda_{t+1})} Q_{t+1}\left((h_2^0, \gamma_{\lambda,h_2^0}, o_{t+1}^0), \gamma_{\lambda',(h_2^0,\gamma_{\lambda,h_2^0},o_{t+1}^0)}\right) \\
& && \text{(induction hypothesis and (SCS1))} \\
&= \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_2^0, \lambda) \cdot \max_{\lambda' \in \Omega(\Lambda_{t+1})} Q_{t+1}\left((h_2^0, \gamma_{\lambda,h_2^0}, o_{t+1}^0), \gamma_{\lambda',(h_2^0,\gamma_{\lambda,h_2^0},o_{t+1}^0)}\right) && \text{((SCS3))} \\
&= \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_2^0, \gamma_{\lambda,h_2^0}) V_{t+1}((h_2^0, \gamma_{\lambda,h_2^0}, o_{t+1}^0)) = \mathbb{E}\left[V_{t+1}((H_t^0, \Gamma_t, O_{t+1}^0))|h_2^0, \gamma_{\lambda,h_2^0}\right].
\end{aligned}$$

The exploitation of (SCS1) in the equality labeled by ($\sharp$) is due to

$$\vartheta_{t+1}^0((h_1^0, \gamma_{\lambda,h_1^0}, o_{t+1}^0)) = \phi_{t+1}^0(\vartheta_t^0(h_1^0), \lambda, o_{t+1}^0) = \phi_{t+1}^0(\vartheta_t^0(h_2^0), \lambda, o_{t+1}^0) = \vartheta_{t+1}^0((h_2^0, \gamma_{\lambda,h_2^0}, o_{t+1}^0)).$$

■

## C.3 Missing Proofs in Section 4.4.1

**Proof of Lemma 4.22:** We preceed the proof by mathematical induction. The instantaneous part and the base case $t = T$ follow trivially from (ASPS2)

$$\begin{aligned}
&\left|\mathbb{E}\left[R_t(S_t, A_t)|h_t^0, h_1, \gamma\right] - \mathbb{E}\left[R_t(S_t, A_t)|h_t^0, h_2, \gamma\right]\right| \\
\leq& \left|\mathbb{E}\left[R_t(S_t, A_t)|h_t^0, h_1, \gamma\right] - \mathbb{E}\left[R_t(S_t, A_t)|h_t^0, \widehat{z}, \gamma\right]\right| \\
&\quad + \left|\mathbb{E}\left[R_t(S_t, A_t)|h_t^0, \widehat{z}, \gamma\right] - \mathbb{E}\left[R_t(S_t, A_t)|h_t^0, h_2, \gamma\right]\right| \\
\leq& \, \epsilon_p/4 + \epsilon_p/4 && \text{((ASPS2))} \\
=& \, \epsilon_p/2.
\end{aligned}$$

For the continuation part, we have

$$\begin{aligned}
&\left|\mathbb{E}\left[V_{t+1}^S((H_t^0, \Gamma_t, O_{t+1}^0), H_{t+1}^{1:N})|h_t^0, h_1, \gamma\right] - \mathbb{E}\left[V_{t+1}^S((H_t^0, \Gamma_t, O_{t+1}^0), H_{t+1}^{1:N})|h_t^0, h_2, \gamma\right]\right| \\
=& \left|\sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_1, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \right. \\
&\quad \left. - \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_2, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N}))\right| && \text{(proof of Lemma 4.12)} \\
\leq& \left|\sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_1, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \right. \\
&\quad \left. - \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, \widehat{z}, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N}))\right| \\
&+ \left|\sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, \widehat{z}, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \right. \\
&\quad \left. - \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_2, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N}))\right| \\
&+ \left|\sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_2, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \right.
\end{aligned}$$

$$
\left. - \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N} | h_t^0, h_2, a) V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N})) \right|
$$

$$
:= ① + ② + ③.
$$

For the first two terms, we have

$$
①, ② \leq 2\|V_{t+1}\|_\infty \cdot \delta_p/8 \leq T\bar{R}\delta_p/4
$$

by (ASPS3). Note that the above equation follows if $\mathcal{K}(\cdot,\cdot)$ is the total variation distance. If it is instead the Wasserstein metric, then the total variation distance will still be bounded by $\delta_p \big/ \min_{x,y \in \Omega(O_{t+1}^{0:N}), x \neq y} \|x - y\|$; we can redefine this value as $\delta_p$ so that the total variation distance is still bounded by $\delta_p$.

Now consider a fixed realization of $o_{t+1}^{0:N}$. We have

$$
\begin{aligned}
&\widehat{\vartheta}_{t+1}^{1:N}((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) \\
&= \widehat{\phi}_{t+1}^{1:N}(\widehat{\vartheta}_t^{1:N}(h_1), h_t^0, \gamma, o_{t+1}^{0:N}) && \text{((ASPS1))} \\
&= \widehat{\phi}_{t+1}^{1:N}(\widehat{\vartheta}_t^{1:N}(h_2), h_t^0, \gamma, o_{t+1}^{0:N}) && \text{(assumption)} \\
&= \widehat{\vartheta}_{t+1}^{1:N}((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N})), && \text{((ASPS1))}
\end{aligned}
$$

so that under the public FCS $(h_t^0, \gamma, o_{t+1}^0)$, the two FPSs $(h_1, a, o_{t+1}^{1:N})$ and $(h_2, a, o_{t+1}^{1:N})$ will be mapped to the same ASPS as well. Hence, by the induction hypothesis of Lemma 4.22 which leads to Corollary 4.23 at the $t+1$ step, we obtain

$$
\begin{aligned}
&\left| V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) - V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N})) \right| \\
&\leq (T - t - 1)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2.
\end{aligned}
$$

The last term can thus be bounded by

$$
\begin{aligned}
&③ \\
&\leq \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N} | h_t^0, h_2, a) \left| V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_1, a, o_{t+1}^{1:N})) - V_{t+1}^S((h_t^0, \gamma, o_{t+1}^0), (h_2, a, o_{t+1}^{1:N})) \right| \\
&\leq \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N} | h_t^0, h_2, a) [(T - t - 1)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2]
\end{aligned}
$$

$$= (T-t-1)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2.$$

Combining the three terms plus the instantaneous part, it follows that

$$\begin{aligned}
&\left|Q_t^S(h_t^0, h_1, \gamma) - Q_t^S(h_t^0, h_2, \gamma)\right| \\
\le\ &\epsilon_p/2 + 2 \cdot T\bar{R}\delta_p/4 + (T-t-1)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2 \\
=\ &(T-t)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2.
\end{aligned}$$

■

**Proof of Corollary 4.23:** Assume the optimal prescription $\gamma^*$ prescribes different actions on the two FPSs $h_1, h_2 \in \Omega(H_t^{1:N})$, so that $\gamma^*(h_1) = a_1$ and $\gamma^*(h_2) = a_2$ where $a_1 \neq a_2$; otherwise, the claim directly follows by Lemma 4.22. Also, define $\gamma', \gamma'' \in \Omega(\Gamma_t)$ by

$$\gamma'(h) = \begin{cases} a_1 & \text{if } h = h_2, \\ \gamma^*(h) & \text{otherwise,} \end{cases} \qquad \gamma''(h) = \begin{cases} a_2 & \text{if } h = h_1, \\ \gamma^*(h) & \text{otherwise.} \end{cases}$$

Again let $\upsilon_1 = Q_t^S(h_t^0, h_1, \gamma^*)$ and $\upsilon_2 = Q_t^S(h_t^0, h_2, \gamma^*)$. Denote $B_t \triangleq (T-t)(\epsilon_p + T\bar{R}\delta_p)/2 + \epsilon_p/2$ for simplicity. Similar to the proof of Corollary 4.13, $Q_t(h_t, \gamma^*)$ is expanded as (throughout the proof only admissible FPSs are considered – we omit specifications of the constraint for cleaner notations)

$$Q_t(h_t^0, \gamma^*) = \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0)\upsilon_1 + \mathbb{P}(h_2|h_t^0)\upsilon_2.$$

Likewise, we can also expand $Q_t(h_t, \gamma')$ to

$$
\begin{aligned}
& Q_t(h_t^0, \gamma') \\
&= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma') + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma') + \mathbb{P}(h_2|h_t^0) Q_t^S(h_t^0, h_2, \gamma') \\
&\geq \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma') + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma') + \mathbb{P}(h_2|h_t^0) \left[ Q_t^S(h_t^0, h_1, \gamma') - B_t \right] \\
&\hspace{30em} \text{(Lemma 4.22)} \\
&= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) Q_t^S(h_t^0, h_1, \gamma^*) + \mathbb{P}(h_2|h_t^0) \left[ Q_t^S(h_t^0, h_1, \gamma^*) - B_t \right] \\
&\hspace{30em} \text{(Lemma 4.11)} \\
&= \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) \upsilon_1 + \mathbb{P}(h_2|h_t^0)(\upsilon_1 - B_t);
\end{aligned}
$$

by symmetry

$$
Q_t(h_t^0, \gamma'') \geq \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0)(\upsilon_2 - B_t) + \mathbb{P}(h_2|h_t^0) \upsilon_2.
$$

We have

$$
\begin{aligned}
[4] \qquad & \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) \upsilon_1 + \mathbb{P}(h_2|h_t^0)(\upsilon_1 - B_t) \leq Q_t(h_t^0, \gamma') \\
& \leq Q_t(h_t^0, \gamma^*) = \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) \upsilon_1 + \mathbb{P}(h_2|h_t^0) \upsilon_2
\end{aligned}
$$

and

$$
\begin{aligned}
& \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0)(\upsilon_2 - B_t) + \mathbb{P}(h_2|h_t^0) \upsilon_2 \leq Q_t(h_t^0, \gamma'') \\
& \leq Q_t(h_t^0, \gamma^*) = \sum_{h \neq h_1, h_2} \mathbb{P}(h|h_t^0) Q_t^S(h_t^0, h, \gamma^*) + \mathbb{P}(h_1|h_t^0) \upsilon_1 + \mathbb{P}(h_2|h_t^0) \upsilon_2.
\end{aligned}
$$

Canceling and rearranging the terms yield

$$
-B_t \leq \upsilon_1 - \upsilon_2 \leq B_t.
$$

■

**Proof of Theorem 4.24:** We prove the result by induction. The base case trivially follows from Proposition 4.21. Note that the continuation values at $T+1$ are defined to be 0, i.e. $V_{T+1}(h^0_{T+1}) \triangleq 0$ and $\widehat{V}_{T+1}(h^0_{T+1}) \triangleq 0$ for any $h^0_{T+1} \in \Omega(H^0_{T+1})$. Hence, for any $h^0_T \in \Omega(H^0_T)$ and $\gamma^* \in \Omega(\Gamma_t)$, we have

$$Q_T(h^0_T, \gamma^*) - \epsilon_p = V_T(h^0_T) - \epsilon_p \leq \max_{\widehat{\lambda} \in \Omega(\widehat{\Lambda}_T)} Q_T(h^0_T, \gamma_{\widehat{\lambda}, h^0_T}) = \max_{\widehat{\lambda} \in \Omega(\widehat{\Lambda}_T)} \widehat{Q}_T(h^0_T, \widehat{\lambda}) = \widehat{V}_T(h^0_T).$$

In the equation, $Q_T(h^0_T, \gamma_{\widehat{\lambda}, h^0_T}) = \widehat{Q}_T(h^0_T, \widehat{\lambda})$ because there is no continuation value for $T$. Now for the induction step, we assume the induction hypothesis, i.e. the claim holds for some $t+1 \leq T$ so that we have for any $h^0_{t+1} \in \Omega(H^0_{t+1})$,

$$V_{t+1}(h^0_{t+1}) - \widehat{V}_{t+1}(h^0_{t+1}) \leq \frac{(T-t-1)(T-t)}{2}(\epsilon_p + T\bar{R}\delta_p) + (T-t)\epsilon_p.$$

Proposition 4.21 states that for any $h^0_t \in \Omega(H^0_t)$ and optimal prescription $\gamma^* \in \operatorname{argmax}_\gamma Q_t(h^0_t, \gamma)$, there exists a $\widehat{\lambda} \in \Omega(\widehat{\Lambda}_t)$ such that

$$Q_t(h^0_t, \gamma^*) - Q_t(h^0_t, \gamma_{\widehat{\lambda}, h^0_t}) \leq (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p.$$

Write $\mathfrak{C}_t \triangleq (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p$ for shorthand of notation. Then for this $\widehat{\lambda}$, we have

$$\begin{aligned}
&Q_t(h^0_t, \gamma^*) - \widehat{Q}_t(h^0_t, \widehat{\lambda}) = Q_t(h^0_t, \gamma^*) - Q_t(h^0_t, \gamma_{\widehat{\lambda}, h^0_t}) + Q_t(h^0_t, \gamma_{\widehat{\lambda}, h^0_t}) - \widehat{Q}_t(h^0_t, \widehat{\lambda}) \\
&\leq \mathfrak{C}_t + \mathbb{E}[R_t + V_{t+1}(H^0_{t+1}) | h^0_t, \gamma_{\widehat{\lambda}, h^0_t}] - \mathbb{E}[R_t + \widehat{V}_{t+1}(H^0_{t+1}) | h^0_t, \gamma_{\widehat{\lambda}, h^0_t}] \\
&= \mathfrak{C}_t + \sum_{h^0_{t+1}} \mathbb{P}(h^0_{t+1} | h^0_t, \gamma_{\widehat{\lambda}, h^0_t}) \left[ V_{t+1}(h^0_{t+1}) - \widehat{V}_{t+1}(h^0_{t+1}) \right] \\
&\leq \mathfrak{C}_t + \sum_{h^0_{t+1}} \mathbb{P}(h^0_{t+1} | h^0_t, \gamma_{\widehat{\lambda}, h^0_t}) \left[ \frac{(T-t-1)(T-t)}{2}(\epsilon_p + T\bar{R}\delta_p) + (T-t)\epsilon_p \right] \\
&= (T-t)(\epsilon_p + T\bar{R}\delta_p) + \epsilon_p + \frac{(T-t-1)(T-t)}{2}(\epsilon_p + T\bar{R}\delta_p) + (T-t)\epsilon_p \\
&= \frac{(T-t)(T-t+1)}{2}(\epsilon_p + T\bar{R}\delta_p) + (T-t+1)\epsilon_p.
\end{aligned}$$

■

## C.4 Missing Proofs in Section 4.4.2

**Proof of Proposition 4.26:** We proceed the proof again by mathematical induction. The instantaneous part as well as the base case $t = T$ trivially follow from (ASCS2)

$$\begin{aligned}
&\left|\mathbb{E}\left[R_t(S_t, A_t)|h_1^0, \widehat{\lambda}\right] - \mathbb{E}\left[R_t(S_t, A_t)|h_2^0, \widehat{\lambda}\right]\right| \\
\leq &\left|\mathbb{E}\left[R_t(S_t, A_t)|h_1^0, \widehat{\lambda}\right] - \mathbb{E}\left[R_t(S_t, A_t)|\widehat{z}^0, \widehat{\lambda}\right]\right| \\
&\quad + \left|\mathbb{E}\left[R_t(S_t, A_t)|\widehat{z}^0, \widehat{\lambda}\right] - \mathbb{E}\left[R_t(S_t, A_t)|h_2^0, \widehat{\lambda}\right]\right| \\
\leq &\, \epsilon_c + \epsilon_c = 2\epsilon_c. && \text{((ASCS2))}
\end{aligned}$$

For the continuation part in the induction step, we have

$$\begin{aligned}
&\left|\mathbb{E}\left[\widehat{V}_{t+1}((H_t^0, \widehat{\Lambda}_t, O_{t+1}^0))|h_1^0, \widehat{\lambda}\right] - \mathbb{E}\left[\widehat{V}_{t+1}((H_t^0, \widehat{\Lambda}_t, O_{t+1}^0))|h_2^0, \widehat{\lambda}\right]\right| \\
= &\left|\sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_1^0, \widehat{\lambda})\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0)) - \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_2^0, \widehat{\lambda})\widehat{V}_{t+1}((h_2^0, \widehat{\lambda}, o_{t+1}^0))\right| \\
\leq &\left|\sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_1^0, \widehat{\lambda})\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0)) - \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|\widehat{z}^0, \widehat{\lambda})\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0))\right| \\
&\quad + \left|\sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|\widehat{z}^0, \widehat{\lambda})\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0)) - \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_2^0, \widehat{\lambda})\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0))\right| \\
&\quad + \left|\sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_2^0, \widehat{\lambda})\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0)) - \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0|h_2^0, \widehat{\lambda})\widehat{V}_{t+1}((h_2^0, \widehat{\lambda}, o_{t+1}^0))\right| \\
:= &\, ① + ② + ③.
\end{aligned}$$

For the first two terms, we have

$$①, ② \leq 2\|\widehat{V}_{t+1}\|_\infty \cdot \delta_c/2 \leq T\bar{R}\delta_c$$

by (ASCS3).

Now consider a fixed realization of $o_{t+1}^0$. We have

$$\begin{aligned}
&\widehat{\vartheta}_{t+1}^0((h_1^0, \widehat{\lambda}, o_{t+1}^0)) \\
&= \widehat{\phi}_{t+1}^0(\widehat{\vartheta}_t^0(h_1), \widehat{\lambda}, o_{t+1}^0) && \text{((ASCS1))} \\
&= \widehat{\phi}_{t+1}^0(\widehat{\vartheta}_t^0(h_2), \widehat{\lambda}, o_{t+1}^0) && \text{(assumption)} \\
&= \widehat{\vartheta}_{t+1}^0((h_2^0, \widehat{\lambda}, o_{t+1}^0)), && \text{((ASCS1))}
\end{aligned}$$

so that the two FCSs (with ASPS-based prescription) $(h_1^0, \widehat{\lambda}, o_{t+1}^0)$ and $(h_2^0, \widehat{\lambda}, o_{t+1}^0)$ will be mapped to the same ASCS as well. Hence, by the induction hypothesis, we obtain

$$\left|\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0)) - \widehat{V}_{t+1}((h_2^0, \widehat{\lambda}, o_{t+1}^0))\right| \leq 2(T-t-1)(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c.$$

The last term can thus be bounded by

$$\begin{aligned}
③ &\leq \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0 | h_2^0, \widehat{\lambda}) \left|\widehat{V}_{t+1}((h_1^0, \widehat{\lambda}, o_{t+1}^0)) - \widehat{V}_{t+1}((h_2^0, \widehat{\lambda}, o_{t+1}^0))\right| \\
&\leq \sum_{o_{t+1}^0} \mathbb{P}(o_{t+1}^0 | h_2^0, \widehat{\lambda})[2(T-t-1)(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c] = 2(T-t-1)(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c.
\end{aligned}$$

Combining the three terms plus the instantaneous part, it follows that

$$\begin{aligned}
\left|\widehat{Q}_t(h_1^0, \widehat{\lambda}) - \widehat{Q}_t(h_2^0, \widehat{\lambda})\right| &\leq 2\epsilon_c + 2 \cdot T\bar{R}\delta_c + 2(T-t-1)(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c \\
&= 2(T-t)(\epsilon_c + T\bar{R}\delta_c) + 2\epsilon_c.
\end{aligned}$$

■

**Proof of Theorem 4.27:** We proceed the proof again by mathematical induction.

The base case $t = T$ trivially follows from (ASCS2). For the induction step, we have

$$\begin{aligned}
&\widehat{Q}_t(h_t^0, \widehat{\lambda}) - \breve{Q}_t(\widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}) \\
&= \mathbb{E}\left[R_t(S_t, A_t) | h_t^0, \widehat{\lambda}\right] - \mathbb{E}\left[R_t(S_t, A_t) | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}\right] \\
&\quad + \mathbb{E}\left[\widehat{V}_{t+1}(H_{t+1}^0) | h_t^0, \widehat{\lambda}\right] - \mathbb{E}\left[\breve{V}_{t+1}(\widehat{\vartheta}_{t+1}^0(H_{t+1}^0)) | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}\right] \\
&\le \epsilon_c + \mathbb{E}\left[\widehat{V}_{t+1}(H_{t+1}^0) | h_t^0, \widehat{\lambda}\right] - \mathbb{E}\left[\breve{V}_{t+1}(\widehat{\vartheta}_{t+1}^0(H_{t+1}^0)) | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}\right] \qquad \text{((ASCS2))} \\
&= \epsilon_c + \mathbb{E}\left[\widehat{V}_{t+1}(H_{t+1}^0) | h_t^0, \widehat{\lambda}\right] - \mathbb{E}\left[\widehat{V}_{t+1}(H_{t+1}^0) | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}\right] \\
&\quad + \mathbb{E}\left[\widehat{V}_{t+1}(H_{t+1}^0) | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}\right] - \mathbb{E}\left[\breve{V}_{t+1}(\widehat{\vartheta}_{t+1}^0(H_{t+1}^0)) | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}\right] \\
&= \epsilon_c + \sum_{h_{t+1}^0} \left[\mathbb{P}(h_{t+1}^0 | h_t^0, \widehat{\lambda}) - \mathbb{P}(h_{t+1}^0 | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda})\right] \widehat{V}_{t+1}(h_{t+1}^0) \\
&\quad + \sum_{h_{t+1}^0} \mathbb{P}(h_{t+1}^0 | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}) \left[\widehat{V}_{t+1}(h_{t+1}^0) - \breve{V}_{t+1}(\widehat{\vartheta}_{t+1}^0(h_{t+1}^0))\right] \\
&:= \epsilon_c + ① + ②.
\end{aligned}$$

The first term is bounded by (ASCS3)

$$\begin{aligned}
① &= \sum_{o_{t+1}^0} \left[\mathbb{P}(o_{t+1}^0 | h_t^0, \widehat{\lambda}) - \mathbb{P}(o_{t+1}^0 | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda})\right] \widehat{V}_{t+1}((h_t^0, \widehat{\lambda}, o_{t+1}^0)) \\
&\le 2\|\widehat{V}_{t+1}\|_\infty \cdot \delta_c/2 \le T\bar{R}\delta_c,
\end{aligned}$$

while the second term can be bounded by the induction hypothesis

$$\begin{aligned}
② &\le \sum_{h_{t+1}^0} \mathbb{P}(h_{t+1}^0 | \widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}) \left[(T - t - 1)(\epsilon_c + T\bar{R}\delta_c) + \epsilon_c\right] \\
&= (T - t - 1)(\epsilon_c + T\bar{R}\delta_c) + \epsilon_c.
\end{aligned}$$

Combining the terms, it follows that

$$\begin{aligned}
\widehat{Q}_t(h_t^0, \widehat{\lambda}) - \breve{Q}_t(\widehat{\vartheta}_t^0(h_t^0), \widehat{\lambda}) &\le \epsilon_c + T\bar{R}\delta_c + (T - t - 1)(\epsilon_c + T\bar{R}\delta_c) + \epsilon_c \\
&= (T - t)(\epsilon_c + T\bar{R}\delta_c) + \epsilon_c.
\end{aligned}$$

The $V$ part of the claim can be obtained by considering an optimal prescription

$\widehat{\lambda}^* \in \operatorname{argmax}_{\widehat{\lambda}\in\Omega(\widehat{\Lambda}_t)} \widehat{Q}_t(h_t^0, \widehat{\lambda})$ in the $Q$ part. ■

## C.5 Missing Proofs in Section 4.5.1

**Proof of Proposition 4.29:** For (SCS1), BCSs can be updated recursively through Bayesian updates [88]. For (SCS2), notice that

$$\mathbb{E}[R_t(S_t, A_t)|h_t^0, \lambda_t] = \sum_{s_t, h_t^{1:N}} \mathbb{P}(s_t, h_t^{1:N}|h_t^0) R(s_t, \gamma_t(h_t^{1:N})),$$

and the ensemble of $\mathbb{P}(s_t, h_t^{1:N}|h_t^0)$ through their spaces is exactly $\Pi_t(h_t^0) = \mathbb{P}(S_t, H_t^{1:N}|h_t^0)$. Similarly, it satisfies (SCS3) as well, since

$$\mathbb{P}(O_{t+1}^0|h_t^0, \gamma_t) = \sum_{s_t, h_t^{1:N}} \mathbb{P}(s_t, h_t^{1:N}|h_t^0) \cdot \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|s_t, \gamma_t(h_t^{1:N})) \cdot \mathbb{P}(O_{t+1}^0|s_{t+1}).$$

The quantity $\Pi_t(h_t^0) = \mathbb{P}(S_t, H_t^{1:N}|h_t^0)$ again exactly encapsulates what is needed to compute $\mathbb{P}(O_{t+1}^0|h_t^0, \gamma_t)$. ■

**Proof of Proposition 4.30:**

$$\mathbb{P}(z_{t+1}^{1:N}, o_{t+1}^0|h_t^0, h_t^{1:N}, \gamma_t, a_t) = \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(z_{t+1}^{1:N}, o_{t+1}^{0:N}, s_{t+1}|h_t^0, h_t^{1:N}, \gamma_t, a_t)$$

$$= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|h_t^0, h_t^{1:N}, \gamma_t, a_t) \cdot \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_t^{1:N}, \gamma_t, a_t, s_{t+1}) \cdot \mathbb{P}(z_{t+1}^{1:N}|h_t^0, h_t^{1:N}, \gamma_t, a_t, s_{t+1}, o_{t+1}^{0:N})$$

$$= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|h_t^0, h_t^{1:N}, \gamma_t) \cdot \mathbb{P}(o_{t+1}^{0:N}|s_{t+1}) \cdot \mathbb{P}(z_{t+1}^{1:N}|h_t^0, h_t^{1:N}, \gamma_t, s_{t+1}, o_{t+1}^{0:N})$$

(redundancy of $a_t$ and $P_O$ specifies $O_{t+1}$ given $S_{t+1}$)

$$= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|h_t^0, h_t^{1:N}, a_t) \cdot \mathbb{P}(o_{t+1}^{0:N}|s_{t+1}) \cdot \mathbb{P}(z_{t+1}^{1:N}|h_t^0, h_t^{1:N}, \gamma_t, s_{t+1}, o_{t+1}^{0:N})$$

(Lemma C.1)

$$= \sum_{o_{t+1}^{0:N}} \sum_{s_{t+1}} \mathbb{P}(s_{t+1}, o_{t+1}^{0:N}|h_t^0, h_t^{1:N}, a_t) \cdot \mathbb{I}\{z_{t+1}^{1:N} = \phi_{t+1}^{1:N}(\vartheta_t(h_t^0, h_t^{1:N}), \gamma_t, o_{t+1}^{0:N})\} \qquad \text{((SPS1))}$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, h_t^{1:N}, a_t) \cdot \mathbb{I}\{z_{t+1}^{1:N} = \phi_{t+1}^{1:N}(z_t^{1:N}, \gamma_t, o_{t+1}^{0:N})\}$$

$$= \sum_{o_{t+1}^{0:N}} \mathbb{P}(o_{t+1}^{0:N}|h_t^0, z_t^{1:N}, a_t) \cdot \mathbb{I}\{z_{t+1}^{1:N} = \phi_{t+1}^{1:N}(z_t^{1:N}, \gamma_t, o_{t+1}^{0:N})\} \qquad \text{((SPS3))}$$

$$= \mathbb{P}(z_{t+1}^{1:N}, o_{t+1}^0|h_t^0, z_t^{1:N}, \gamma_t, a_t).$$

Note the last equality follows as in (SPS3) it is implicitly assumed that $z_t^{1:N} = \vartheta_t(h_t^0, h_t^{1:N})$. ■

**Proof of Proposition 4.31:**

$$\mathbb{E}[R(S_t, A_t)|h_t^0, h_t^n, a_t] = \sum_{s_t} R(s_t, a_t)\mathbb{P}(s_t|h_t^0, h_t^n) = \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(s_t, z_t^{-n}|h_t^0, h_t^n)$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \cdot \mathbb{P}(s_t|h_t^0, h_t^n, z_t^{-n})$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(s_t, h_t^{-n}|h_t^0, h_t^n, z_t^{-n})$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n}) \cdot \mathbb{P}(s_t|h_t^0, h_t^n, z_t^{-n}, h_t^{-n})$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n}) \cdot \mathbb{P}(s_t|h_t^0, h_t^n, h_t^{-n})$$

$$(z_t^{-n} = \vartheta_t^{-n}(h_t^0, h_t^{-n}))$$

$$= \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n}) \sum_{s_t} R(s_t, a_t)\mathbb{P}(s_t|h_t^0, h_t^n, h_t^{-n})$$

$$= \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n})\mathbb{E}[R(S_t, A_t)|h_t^0, h_t^{1:N}, a_t]$$

$$= \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n})\mathbb{E}[R(S_t, A_t)|h_t^0, z_t^{1:N}, a_t] \qquad \text{((SPS2)))}$$

$$= \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n}) \sum_{s_t} R(s_t, a_t)\mathbb{P}(s_t|h_t^0, z_t^n, z_t^{-n})$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, h_t^n) \cdot \mathbb{P}(s_t|h_t^0, z_t^n, z_t^{-n}) \sum_{h_t^{-n}} \mathbb{P}(h_t^{-n}|h_t^0, h_t^n, z_t^{-n})$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(z_t^{-n}|h_t^0, z_t^n) \cdot \mathbb{P}(s_t|h_t^0, z_t^n, z_t^{-n}) \cdot 1 \qquad \text{((SPI4))}$$

$$= \sum_{s_t} R(s_t, a_t) \sum_{z_t^{-n}} \mathbb{P}(s_t, z_t^{-n}|h_t^0, z_t^n) = \sum_{s_t} R(s_t, a_t)\mathbb{P}(s_t|h_t^0, z_t^n) = \mathbb{E}[R(S_t, A_t)|h_t^0, z_t^n, a_t].$$

Note that the superscript $-n$ only contains $[N] \setminus \{n\}$ and does not contain 0. ■

## APPENDIX D

# Proofs for Chapter V

### D.1 The $K$-arm UCB

In this subsection, we prove one regret bound of the UCB1 algorithm [5] that will be used later. At time $t$, define $\hat{\mu}_t(k) = \frac{1}{n_t(k)} \sum_{s=1}^{t-1} R_s(k_s)\mathbf{1}\{k_s = k\}$ to be the empirical mean of the $k$-th arm based on it being played $n_t(k) = \sum_{s=1}^{t-1} \mathbf{1}\{k_s = k\}$ times up until $t-1$. As in the two-stage setting, we assume the best arm is the first.

**Algorithm D.1** $K$-arm UCB

```
input: $K$ (#actions)
for $t = 1, \dots, K$ do
    Play each arm $k$ once, where $k \in [K]$.
for $t = K+1, \dots, T$ do
    Chooses $k_t = \arg\max_{k \in [K]} \hat{\mu}_t(k) + 2\sqrt{\frac{\log(KT/\delta)}{n_t(k)}}$.
```

In the following, all the $t$'s and $T$'s within any bound is assumed to be sufficiently large (in particular $> K$), so that we do not have the initialization issue.

**Lemma D.1:** *For any $t \in [T] \setminus [K]$ and $k \in [K]$, with probability at least $1 - \delta$, we have*

$$|\mu_k - \hat{\mu}_t(k)| \le 2\sqrt{\frac{\log(1/\delta)}{n_t(k)}}.$$

**Proof:** This is Hoeffding's inequality. ■

**Corollary D.2:** *With probability at least* $1-\delta$*, the inequality*

$$|\mu_k - \hat{\mu}_t(k)| \le 2\sqrt{\frac{\log(KT/\delta)}{n_t(k)}}.$$

*holds for all* $t \in [T] \setminus [K]$ *and* $k \in [K]$ *simultaneously.*

**Proof:** From Lemma D.1, the inequality holds for any $t \in [T] \setminus [K]$ and $k \in [K]$ with probability at least $1 - \delta/KT$. Hence, it holds for all $t \in [T] \setminus [K]$ and $k \in [K]$ simultaneously with probability at least $1-\delta$ by the union bound. ■

**Lemma D.3:** *With probability at least* $1-2\delta$*, the regret is bounded by*

$$\sum_{t=1}^{T}[\mu_1 - R_t(k_t)] \le 11\sqrt{KT\log(KT/\delta)}.$$

**Proof:** With probability at least $1-\delta$, for all $t \in [T] \setminus [K]$ simultaneously,

$$\begin{aligned}
\mu_1 &\le \hat{\mu}_t(1) + 2\sqrt{\frac{\log(KT/\delta)}{n_t(1)}} && \text{(Corollary D.2)}\\
&\le \hat{\mu}_t(k_t) + 2\sqrt{\frac{\log(KT/\delta)}{n_t(k_t)}} && \text{(by the way } k_t \text{ is selected)}\\
&\le \mu_{k_t} + 4\sqrt{\frac{\log(KT/\delta)}{n_t(k_t)}}. && \text{(Corollary D.2)}
\end{aligned}$$

Therefore, with probability at least $1-\delta$,

$$\begin{aligned}
\sum_{t=1}^{T}(\mu_1 - \mu_{k_t}) &\le K + 4\sum_{t=K+1}^{T}\sqrt{\frac{\log(KT/\delta)}{n_t(k_t)}}\\
&= K + 4\sum_{k\in[K]}\sum_{t=K+1}^{T}\mathbf{1}\{k_t = k\}\sqrt{\frac{\log(KT/\delta)}{n_t(k_t)}}\\
&= K + 4\sum_{k\in[K]}\left(\sqrt{\frac{1}{1}} + \sqrt{\frac{1}{2}} + \cdots + \sqrt{\frac{1}{n_T(k)}}\right)\sqrt{\log(KT/\delta)}\\
&\le K + 8\sum_{k\in[K]}\sqrt{n_T(k)}\cdot\sqrt{\log(KT/\delta)} && \text{(Integral test for series)}
\end{aligned}$$

$$\le K + 8\sqrt{K \cdot \sum_{k\in[K]} n_T(k)} \cdot \sqrt{\log(KT/\delta)}$$

(Cauchy-Schwartz inequality)

$$= 8\sqrt{KT\log(KT/\delta)} + K.$$

Notice that $\{\sum_{s=1}^{t}[\mu_{k_s} - R_s(k_s)]\}$ forms a martingale, and $|\mu_{k_t} - R_t(k_t)| \le 1$. Thus, by Azuma's inequality, with probability at least $1-\delta$,

$$\sum_{t=1}^{T}[\mu_{k_t} - R_t(k_t)] \le 2\sqrt{T\log(1/\delta)}.$$

Using the union bound, we have with probability at least $1-2\delta$,

$$\begin{aligned}\sum_{t=1}^{T}[\mu_1 - R_t(k_t)] &\le 8\sqrt{KT\log(KT/\delta)} + 2\sqrt{T\log(1/\delta)} + K \\ &\le 11\sqrt{KT\log(KT/\delta)}.\end{aligned}$$

■

## D.2 Missing Proofs in Section 5.3

**Proof of Theorem 5.2:**

$$\begin{aligned}&\sum_{t=1}^{T}[\mu_{1,1} - R_t(k_t, l_t)] \\ &\le KL + \sum_{t=KL+1}^{T}\left\{\max_k\left[\hat{\mu}_t^1(k) + 11\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k)}}\right] - \mu_{k_t,1} + \mu_{k_t,1} - R_t(k_t, l_t)\right\}\end{aligned}$$

((5.4) and taking max)

$$= KL + \sum_{t=KL+1}^{T}\left\{\hat{\mu}_t^1(k_t) + 11\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k_t)}} - \mu_{k_t,1} + \mu_{k_t,1} - R_t(k_t, l_t)\right\}.$$

(by the way $k_t$ is selected)

Notice that for any $t \in [T]$ and $k \in [K]$,

$$\begin{aligned}\hat{\mu}_t^1(k) - \mu_{k,1} &= \frac{1}{n_t^1(k)} \sum_{s=1}^{t-1} \mathbf{1}\{k_s = k\}[R_s(k_s, l_s) - \mu_{k,1}] \\ &\le \frac{1}{n_t^1(k)} \sum_{s=1}^{t-1} \mathbf{1}\{k_s = k\}[R_s(k_s, l_s) - \mu_{k,l_s}].\end{aligned}$$

The term $\mathbf{1}\{k_s = k\}[R_s(k_s, l_s) - \mu_{k,l_s}]$ has zero mean, and its absolute value is bounded by $\mathbf{1}\{k_s = k\}$. Hence, by Azuma's inequality, with high probability at least $1 - \delta$,

$$\sum_{s=1}^{t-1} \mathbf{1}\{k_s = k\}[R_s(k_s, l_s) - \mu_{k,l_s}] \le 2\sqrt{\sum_{s=1}^{t-1} \mathbf{1}\{k_s = k\} \log(1/\delta)} = 2\sqrt{n_t^1(k) \log(1/\delta)}.$$

Combining the two inequalities gives

$$\hat{\mu}_t^1(k_t) - \mu_{k_t,1} \le 2\sqrt{\frac{\log(1/\delta)}{n_t^1(k_t)}},$$

so that with probability at least $1 - \delta$,

$$\hat{\mu}_t^1(k_t) - \mu_{k_t,1} \le 2\sqrt{\frac{\log(T/\delta)}{n_t^1(k_t)}},$$

holds for all $t \in [T]$ simultaneously by union bound. So the sum of the first three terms is bounded by

$$\sum_{t=KL+1}^{T} 13\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k_t)}} \le 26\sqrt{KT} \cdot \sqrt{L\log(LT/\delta)} \le O(\sqrt{KLT\log(LT/\delta)}).$$

For how to bound $\sum_{t=KL+1}^{T} \sqrt{\frac{1}{n_t^1(k_t)}}$ by $2\sqrt{KT}$, see the proof of Lemma D.3. The last two terms sum up to

$$\begin{aligned}
\sum_{t=KL+1}^{T} \mu_{k_t,1} - R_t(k_t, l_t) &= \sum_{k\in[K]} \sum_{t=1}^{T} \mathbf{1}\{k_t = k\} \left[\mu_{k,1} - R_t(k, l_t)\right] && \text{(Lemma 5.1)} \\
&\leq \sum_{k\in[K]} O\left(\sqrt{Ln_T^1(k)\log(LT/\delta)}\right) \\
& && \text{(Cauchy-Schwartz inequality)} \\
&= O\left(\sqrt{KLT\log(LT/\delta)}\right).
\end{aligned}$$

Combining all parts, with probability at least $1-5\delta$, or simply $1-\delta$ as the constant term can be absorbed into the big-O notation,

$$\sum_{t=1}^{T} [\mu_{1,1} - R_t(k_t, l_t)] \leq O\left(\sqrt{KLT\log(LT/\delta)}\right).$$

■

**Proof of Lemma 5.3:** Suppose $k$ has been drawn for $\frac{26^2 L\log(LT/\delta)}{(\mu_{1,1}-\mu_{k,1})^2}$ times before time $t$. That is, $n_t^1(k) \geq \frac{26^2 L\log(LT/\delta)}{(\mu_{1,1}-\mu_{k,1})^2}$. Then with probability at least $1-\delta$,

$$\begin{aligned}
&\hat{\mu}_t^1(k) + 11\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k)}} \\
&\leq \mu_{k,l_t} + 2\sqrt{\frac{\log(LT/\delta)}{n_t^1(k)}} + 11\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k)}} && \text{(Corollary D.2)} \\
&\leq \mu_{k,1} + 13\sqrt{\frac{L\log(LT/\delta)}{n_t^1(k)}} && \text{(First arm is optimal)} \\
&\leq \mu_{k,1} + (\mu_{1,1} - \mu_{k,1})/2 < \mu_{1,1}.
\end{aligned}$$

But by (5.4), with probability at least $1-\delta$,

$$\mu_{1,1} \leq \hat{\mu}_t^1(1) + 11\sqrt{\frac{L\log(LT/\delta)}{n_t^1(1)}}.$$

By the algorithm's way of choosing $k_t$, U1 will not choose arm $k$ for all $t \in [T]$ with probability at least $1-2\delta$, as long as the condition of $n_t^1(k)$ is satisfied. ■

**Lemma D.4:** *Let $l \neq 1$ be a sub-optimal arm of U2 given U1 chooses $k$. Then with probability at least $1-2\delta$,*

$$\sum_{t=1}^{T} \mathbf{1}\{k_t = k, l_t = l\} \leq O\left(\frac{\log(LT/\delta)}{(\mu_{k,1}-\mu_{k,l})^2}\right).$$

**Proof:** Suppose at time $t$, $n_t^2(k,l) \geq \frac{64\log(LT/\delta)}{(\mu_{k,1}-\mu_{k,l})^2}$. Then with probability at least $1-\delta$,

$$\begin{aligned}
&\hat{\mu}_t^2(k,l) + 2\sqrt{\frac{\log(LT/\delta)}{n_t^2(k,l)}} \\
&\leq \mu_{k,l} + 4\sqrt{\frac{\log(LT/\delta)}{n_t^2(k,l)}} && \text{(Corollary D.2)} \\
&\leq \mu_{k,l} + \frac{(\mu_{k,1}-\mu_{k,l})}{2} < \mu_{k,1}.
\end{aligned}$$

But by Corollary D.2, with probability at least $1-\delta$,

$$\mu_{k,1} \leq \hat{\mu}_t^2(k,1) + 2\sqrt{\frac{\log(LT/\delta)}{n_t^2(k,1)}}.$$

So for all the round $t \in [T]$, even if U1 chooses $k$, with probability at least $1-2\delta$, U2 will not choose $l$ given the condition of $n_t^2(k,l)$ is satisfied. ■

**Proof of Theorem 5.4:** We show that the regret is upper bounded by

$$\sum_{(k,l)\neq(1,1)} (\mu_{1,1}-\mu_{k,l}) \cdot n_{T+1}^2(k,l) \leq L\sum_{k\neq 1} O\left(\frac{\log(LT)}{\mu_{1,1}-\mu_{k,1}}\right) + \sum_{k}\sum_{l\neq 1} O\left(\frac{\log(LT)}{\mu_{k,1}-\mu_{k,l}}\right).$$

Lemma 5.3 gives with probability at least $1-2\delta$,

$$\sum_{l} n_{T+1}^2(k,l) \leq O\left(\frac{L\log(LT/\delta)}{(\mu_{1,1}-\mu_{k,1})^2}\right) \quad \forall k.$$

Lemma D.4 gives with probability at least $1-2\delta$,

$$n_{T+1}^2(k,l) \leq O\left(\frac{\log(LT/\delta)}{(\mu_{k,1}-\mu_{k,l})^2}\right).$$

Thus, with probability at least $1-4\delta$, we can bound the regret by

$$\begin{aligned}
&\sum_{(k,l)\in[K]\times[L]\setminus(1,1)} (\mu_{1,1}-\mu_{k,l})\cdot n_{T+1}^2(k,l) \\
&= \sum_{(k,l)\in[K]\times[L]\setminus(1,1)} (\mu_{1,1}-\mu_{k,1}+\mu_{k,1}-\mu_{k,l})\cdot n_{T+1}^2(k,l) \\
&= \sum_{k\in[K]\setminus\{1\}} (\mu_{1,1}-\mu_{k,1})\cdot \sum_{l\in[L]} n_{T+1}^2(k,l) + \sum_{k\in[K]}\sum_{l\in[L]\setminus\{1\}} (\mu_{k,1}-\mu_{k,l})\cdot n_{T+1}^2(k,l) \\
&\le \sum_{k\in[K]\setminus\{1\}} (\mu_{1,1}-\mu_{k,1})\cdot O\left(\frac{L\log(LT/\delta)}{(\mu_{1,1}-\mu_{k,1})^2}\right) + \sum_{k\in[K]}\sum_{l\in[L]\setminus\{1\}} (\mu_{k,1}-\mu_{k,l})\cdot O\left(\frac{\log(LT/\delta)}{(\mu_{k,1}-\mu_{k,l})^2}\right) \\
&= L\sum_{k\in[K]\setminus\{1\}} O\left(\frac{\log(LT/\delta)}{\mu_{1,1}-\mu_{k,1}}\right) + \sum_{k\in[K]}\sum_{l\in[L]\setminus\{1\}} O\left(\frac{\log(LT/\delta)}{\mu_{k,1}-\mu_{k,l}}\right).
\end{aligned}$$

The constant term of 4 can be easily absorbed into the big-O notation. ■

## D.3 Missing Proofs in Section 5.4

**Proof of Lemma 5.5:**

Base $d = D-1$:

$$\begin{aligned}
&\sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:D-1} = k^{1:D-1}\}\left[\mu_{(\mathbf{k}^{1:D-1},1)} - R_s(\mathbf{k}_s)\right] \\
&\le C_{D-1}\sqrt{Kn_t^{D-1}(\mathbf{k}^{1:D-1})\log(Kn_t^{D-1}(\mathbf{k}^{1:D-1})/\delta)} \le C_{D-1}\sqrt{Kn_t^{D-1}(\mathbf{k}^{1:D-1})\log(KT/\delta)}
\end{aligned}$$

by common UCB bound (Lemma D.3) and the fact that $C_{D-1} \ge 11$ and $C_D \ge 2$.

Induction step: assume the statement is true for $d+1$, we show that it holds for $d$ as well when $d > 0$. Dividing both sides of the inequality in the induction hypothesis by $n_t^{d+1}(\mathbf{k}^{1:d+1})$ to get

$$\mu_{(\mathbf{k}^{1:d+1},\mathbf{1}_{D-d-1})} \le \hat{\mu}_t^{d+1}(\mathbf{k}^{1:d+1}) + C_{d+1}\sqrt{\frac{K^{D-d-1}\log(K^D T/\delta)}{n_t^{d+1}(\mathbf{k}^{1:d+1})}}. \tag{D.1}$$

Now for the left hand side of the inequality for $d$, we have

$$\sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\}\left[\mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})} - R_s(\mathbf{k}_s)\right]$$

$$\begin{aligned}
&\leq \sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \Bigg\{ \max_{k^{d+1}} \Bigg[ \hat{\mu}_t^{d+1}(\mathbf{k}_s^{1:d}, \mathbf{k}^{d+1}) \\
&\qquad + C_{d+1} \sqrt{\frac{K^{D-d-1} \log(K^D T/\delta)}{n_t^{d+1}(\mathbf{k}_s^{1:d}, k^{d+1})}} \Bigg] - R_s(\mathbf{k}_s) \Bigg\} \\
&\leq \sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \Bigg[ \hat{\mu}_t^{d+1}(\mathbf{k}_s^{1:d+1}) + C_{d+1} \sqrt{\frac{K^{D-d-1} \log(K^D T/\delta)}{n_t^{d+1}(\mathbf{k}_s^{1:d+1})}} - \mu_{(\mathbf{k}_s^{1:d+1}, \mathbf{1}_{D-d-1})} \\
&\qquad + \mu_{(\mathbf{k}_s^{1:d+1}, \mathbf{1}_{D-d-1})} - R_s(\mathbf{k}_s) \Bigg],
\end{aligned} \tag{D.2}$$

where the first inequality follows from (D.1) and taking max, and the second inequality follows from the way $k_s^{d+1}$ is selected.

For any $s \in [T]$, $\mathbf{k}^{1:d+1} \in [K]^{d+1}$, we have

$$\begin{aligned}
&\hat{\mu}_s^{d+1}(\mathbf{k}^{1:d+1}) - \mu_{(\mathbf{k}^{1:d+1}, \mathbf{1}_{D-d-1})} \\
&= \frac{1}{n_s^{d+1}(\mathbf{k}^{1:d+1})} \sum_{u=1}^{s-1} \mathbb{I}\{\mathbf{k}_r^{1:d+1} = \mathbf{k}^{1:d+1}\} \left[ R_u(\mathbf{k}_u) - \mu_{(\mathbf{k}^{1:d+1}, \mathbf{1}_{D-d-1})} \right] \\
&\leq \frac{1}{n_s^{d+1}(\mathbf{k}^{1:d+1})} \sum_{u=1}^{s-1} \mathbb{I}\{\mathbf{k}_r^{1:d+1} = \mathbf{k}^{1:d+1}\} \left[ R_u(\mathbf{k}_u) - \mu_{\mathbf{k}_u} \right] \\
&\leq \frac{2}{n_s^{d+1}(\mathbf{k}^{1:d+1})} \sqrt{\sum_{u=1}^{s-1} \mathbb{I}\{\mathbf{k}_r^{1:d+1} = \mathbf{k}^{1:d+1}\} \log(1/\delta)} = 2 \sqrt{\frac{\log(1/\delta)}{n_s^{d+1}(\mathbf{k}^{1:d+1})}}
\end{aligned}$$

with probability $1 - \delta$. Then, the inequality

$$\hat{\mu}_s^{d+1}(\mathbf{k}^{1:d+1}) - \mu_{(\mathbf{k}^{1:d+1}, \mathbf{1}_{D-d-1})} \leq 2 \sqrt{\frac{\log(K^{d+1} T/\delta)}{n_s^{d+1}(\mathbf{k}^{1:d+1})}} \tag{D.3}$$

holds for all $s \in [T]$ and all $\mathbf{k}^{1:d+1} \in [K]^{d+1}$ simultaneously by the union bound. Hence, the sum of the first three terms of (D.2) can be bounded by

$$
\begin{aligned}
&\sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \left[\hat{\mu}_t^{d+1}(\mathbf{k}_s^{1:d+1}) + C_{d+1}\sqrt{\frac{K^{D-d-1}\log(K^D T/\delta)}{n_t^{d+1}(\mathbf{k}_s^{1:d+1})}} - \mu_{(\mathbf{k}_s^{1:d+1}, \mathbf{1}_{D-d-1})}\right] \\
&\leq \sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \left[2\sqrt{\frac{\log(K^{d+1}T/\delta)}{n_t^{d+1}(\mathbf{k}_s^{1:d+1})}} + C_{d+1}\sqrt{\frac{K^{D-d-1}\log(K^D T/\delta)}{n_t^{d+1}(\mathbf{k}_s^{1:d+1})}}\right] \\
&\leq (C_{d+1}+2)\sqrt{K^{D-d-1}\log(K^D T/\delta)} \cdot \sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \sqrt{\frac{1}{n_s^{d+1}(\mathbf{k}_s^{1:d+1})}} \\
&\leq 2(C_{d+1}+2)\sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta)}.
\end{aligned}
$$

On the other hand, the last two terms sum up to

$$
\begin{aligned}
&\sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \left[\mu_{(\mathbf{k}_s^{1:d+1}, \mathbf{1}_{D-d-1})} - R_s(\mathbf{k}_s)\right] \\
&= \sum_{k^{d+1}\in[K]} \sum_{s=K^D+1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d+1} = \mathbf{k}^{1:d+1}\} \left[\mu_{(\mathbf{k}_s^{1:d+1}, \mathbf{1}_{D-d-1})} - R_s(\mathbf{k}_s)\right] \\
&\leq \sum_{k^{d+1}\in[K]} C_{d+1}\sqrt{K^{D-d-1} n_t^{d+1}(\mathbf{k}^{1:d}) \log(K^D T/\delta)} \\
&\leq C_{d+1}\sqrt{K^{D-d-1} \cdot \left[\sum_{k^{d+1}\in[K]} n_t^{d+1}(\mathbf{k}^{1:d})\right] \log(K^D T/\delta)} \\
&\leq C_{d+1}\sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta)}.
\end{aligned}
$$

Combining both parts, with probability $1 - 3\delta := 1 - \delta'$, we have

$$
\begin{aligned}
&\sum_{s=1}^{t} \mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\} \left[\mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} - R_s(\mathbf{k}_s)\right] \\
&\leq [2(C_{d+1}+2) + C_{d+1}]\sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta)} \\
&\leq \sqrt{3}(3C_{d+1}+4)\sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta')} \\
&\leq (6C_{d+1}+8)\sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta')} \\
&\leq C_d\sqrt{K^{D-d} n_t^d(\mathbf{k}^{1:d}) \log(K^D T/\delta')}
\end{aligned}
$$

whenever $C_d \geq 6C_{d+1} + 8$.

When $d = 0$, the same arguments follow except that $\mathbb{I}\{\mathbf{k}_s^{1:d} = \mathbf{k}^{1:d}\}$ will degenerate to just 1, $n_t^d(\mathbf{k}^{1:d})$ will degenerate to $t$, and the inequality will end with $(6C_1 + 8)\sqrt{K^D t \log(K^D T/\delta')}$ as there is no $C_0$. ■

**Proof of Lemma 5.7:** Suppose at time $t$,

$$n_t^d(\mathbf{k}^{1:d}) \geq \frac{4(C_d+2)^2 K^{D-d} \log(K^D T/\delta)}{\left[\mu_{(\mathbf{k}^{1:d-1},\mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})}\right]^2}. \tag{D.4}$$

Then with probability at least $1-\delta$,

$$\begin{aligned}
&\hat{\mu}_t^d(\mathbf{k}^{1:d}) + C_d\sqrt{\frac{K^{D-d}\log(K^D T/\delta)}{n_t^d(\mathbf{k}^{1:d})}} \\
&\leq \mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})} + 2\sqrt{\frac{\log(K^d T/\delta)}{n_t^d(\mathbf{k}^{1:d})}} + C_d\sqrt{\frac{K^{D-d}\log(K^D T/\delta)}{n_t^d(\mathbf{k}^{1:d})}} && \text{(by (D.3))} \\
&\leq \mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})} + (C_d+2)\sqrt{\frac{K^{D-d}\log(K^D T/\delta)}{n_t^d(\mathbf{k}^{1:d})}} \\
&\leq \mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})} + \frac{\mu_{(\mathbf{k}^{1:d-1},\mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})}}{2} < \mu_{(\mathbf{k}^{1:d-1},\mathbf{1}_{D-d+1})}
\end{aligned}$$

On the other hand, by Lemma 5.5 with $(t, d, (\mathbf{k}^{1:d-1}, 1))$, with probability at least $1-\delta$,

$$\mu_{(\mathbf{k}^{1:d-1},\mathbf{1}_{D-d+1})} \leq \hat{\mu}_t^d(\mathbf{k}^{1:d-1}, 1) + C_d\sqrt{\frac{K^{D-d}\log(K^D T/\delta)}{n_t^d(\mathbf{k}^{1:d-1}, 1)}}.$$

Hence, by the design of Algorithm 5.6, given that the first $d-1$ agents choose $\mathbf{k}^{1:d-1}$, agent $d$ will not choose the sub-optimal arm $k^d \neq 1$ over the first arm before time $T$ when the condition (D.4) is satisfied. ■

**Proof of Theorem 5.8:** From Lemma 5.7 and the union bound, the inequality (5.12) will hold simultaneously for all $d \in [D]$ and $\mathbf{k}^{1:d} \in [K]^d$ with probability at least $1 - 2K^D D\delta$. Hence, with probability at least $1 - 2K^D D\delta$,

$$\begin{aligned}
&\sum_{t=1}^{T}[\mu_{\mathbf{1}_D} - R_t(k_t^1, \ldots, k_t^D)] = \sum_{\mathbf{k}\neq\mathbf{1}_D} (\mu_{\mathbf{1}_D} - \mu_{\mathbf{k}}) \cdot n_{T+1}^D(\mathbf{k}) \\
&= \sum_{\mathbf{k}\neq\mathbf{1}_D}\sum_{d=1}^{D}\left\{\left[\mu_{(\mathbf{k}^{1:d-1},\mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d},\mathbf{1}_{D-d})}\right] \cdot n_{T+1}^D(\mathbf{k})\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{d=1}^{D} \sum_{\mathbf{k}^{1:d} \neq \mathbf{1}_d} \left\{ \left[ \mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} \right] \cdot \left[ \sum_{\mathbf{k}^{d+1:D}} n_{T+1}^{D}(\mathbf{k}) \right] \right\} \\
&= \sum_{d=1}^{D} \sum_{\mathbf{k}^{1:d} \neq \mathbf{1}_d} \left\{ \left[ \mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} \right] \cdot \left[ \sum_{\mathbf{k}^{d+1:D}} n_{T+1}^{D}(\mathbf{k}) \right] \right\} \\
&= \sum_{d=1}^{D} \sum_{\mathbf{k}^{1:d} \neq \mathbf{1}_d} \left\{ \left[ \mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} \right] \cdot \left[ \sum_{t=1}^{T} \mathbb{I}\{\mathbf{k}_t^{1:d} = \mathbf{k}^{1:d}\} \right] \right\} \\
&\le \sum_{d=1}^{D} \sum_{\mathbf{k}^{1:d} \neq \mathbf{1}_d} \left\{ \left[ \mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} \right] \cdot \frac{4(C_d+2)^2 K^{D-d} \log(K^D T/\delta)}{\left[ \mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})} \right]^2} \right\} \\
&= \sum_{d=1}^{D} \sum_{\mathbf{k}^{1:d} \neq \mathbf{1}_d} \frac{4(C_d+2)^2 K^{D-d} \log(K^D T/\delta)}{\mu_{(\mathbf{k}^{1:d-1}, \mathbf{1}_{D-d+1})} - \mu_{(\mathbf{k}^{1:d}, \mathbf{1}_{D-d})}}.
\end{aligned}
$$

Letting $\delta' = 2K^D D \delta$ gives the claim. ■

## D.4 Missing Proofs in Section 5.5

**Proof of Lemma 5.10:** First, we use induction to prove the following inequalities:

$$
0 \le Q_{t,h}^2(s,a,b) - Q_{*,h}(s,a,b) \le \mathbb{E}_{s' \sim P(\cdot|s,a,b)} \left[ V_{t,h+1}^2(s') - V_{*,h+1}(s') \right] + 2\mathsf{bns}_\tau^2, \quad \text{(D.5)}
$$

where $\tau = n_{t,h}(s,a,b)$, for all $s, a, b$. The order of induction is from $t = 1$ to $t = T$, and (within each $t$) from $h = H$ to $h = 1$.

For $t = 1$, we have $Q_{1,h}^2(s,a,b) - Q_{*,h}(s,a,b) = H - Q_{*,h}(s,a,b) \ge 0$ and that $Q_{1,h}^2(s,a,b) - Q_{*,h}(s,a,b) \le H \le 2\mathsf{bns}_0^2$. Suppose that the inequality holds for all $(t', h')$ with either $t' < t$, or $t' = t$ and $h' > h$. Fix a $(s,a,b)$ and let $\tau = n_{t,h}(s,a,b)$. By the update rule of $Q_{t,h}^2(s,a,b)$, we have $Q_{t,h}^2(s,a,b) = \min\left\{ \hat{Q}_{t,h}^2(s,a,b), Q_{t-1,h}^2(s,a,b) \right\}$ where

$$
\hat{Q}_{t,h}^2(s,a,b) = \hat{R}_{t,h}(s,a,b) + \mathbb{E}_{s' \sim \hat{P}_{t,h}(\cdot|s,a,b)} \left[ V_{t,h+1}^2(s') \right] + \mathsf{bns}_\tau^2. \qquad (\tau = n_{t,h}(s,a,b))
$$

Besides,

$$
Q_{*,h}(s,a,b) = R(s,a,b) + \mathbb{E}_{s' \sim P(\cdot|s,a,b)} \left[ V_{*,h}(s') \right].
$$

Taking their difference, we get

$$\begin{aligned}
&\hat{Q}^2_{t,h}(s,a,b) - Q_{*,h}(s,a,b) \\
&= \left(\hat{R}_{t,h}(s,a,b) - R(s,a,b)\right) + \underbrace{\mathbb{E}_{s'\sim P(\cdot|s,a,b)}\left[V^2_{t,h+1}(s') - V_{*,h+1}(s')\right]}_{\mathbf{term}_1} \\
&\qquad + \underbrace{\left(\mathbb{E}_{s'\sim \hat{P}_{t,h}(\cdot|s,a,b)}\left[V^2_{t,h+1}(s')\right] - \mathbb{E}_{s'\sim P(\cdot|s,a,b)}\left[V^2_{t,h+1}(s')\right]\right)}_{\mathbf{term}_2} + \mathsf{bns}^2_\tau. \qquad \text{(D.6)}
\end{aligned}$$

By Lemma D.7 and Lemma D.8, for some universal constant $c > 0$,

$$\left|\hat{R}_{t,h}(s,a,b) - R(s,a,b)\right| \le \frac{1}{2}c\sqrt{\frac{\log(T/\delta)}{\tau^+}}, \qquad \text{(D.7)}$$

$$|\mathbf{term}_2| \le \left\|\hat{P}_{t,h}(\cdot|s,a,b) - P(\cdot|s,a,b)\right\|_1 \|V^2_{t,h+1}\|_\infty \le \frac{1}{2}cH\sqrt{\frac{S\log(T/\delta)}{\tau^+}}, \qquad \text{(D.8)}$$

and therefore $\left|\hat{R}_{t,h}(s,a,b) - R(s,a,b)\right| + |\mathbf{term}_2| \le \mathsf{bns}^2_\tau$. Combining this with (D.6), we get

$$\mathbf{term}_1 \le \hat{Q}^2_{t,h}(s,a,b) - Q_{*,h}(s,a,b) \le \mathbf{term}_1 + 2\mathsf{bns}^2_\tau\,. \qquad \text{(D.9)}$$

Using $Q^2_{t,h}(s,a,b) \le \hat{Q}^2_{t,h}(s,a,b)$, (D.9) implies the right inequality in (D.5).

To prove the left inequality in (D.5), notice that if $h = H$, then $\mathbf{term}_1 = 0$; if $h < H$,

$$\begin{aligned}
&\mathbf{term}_1 \\
&\ge \min_{s'} V^2_{t,h+1}(s') - V_{*,h+1}(s') \\
&= \min_{s'}\left(\min\left\{\max_{a',b'} Q^2_{t,h+1}(s',a',b'), H\right\} - \max_{a',b'} Q_{*,h+1}(s',a',b')\right) \\
&= \min_{s'}\left(\min\left\{\max_{a',b'} Q^2_{t,h+1}(s',a',b') - \max_{a',b'} Q_{*,h+1}(s',a',b'), H - \max_{a',b'} Q_{*,h+1}(s',a',b')\right\}\right) \\
&\ge 0.
\end{aligned}$$

where the last inequality is by the induction hypothesis.

Thus, $\mathbf{term}_1 \ge 0$. Together with (D.9), we get $\hat{Q}^2_{t,h}(s,a,b) - Q_{*,h}(s,a,b) \ge 0$. By the induction hypothesis, we also have $Q^2_{t-1,h}(s,a,b) \ge Q_{*,h}(s,a,b)$. Therefore,

$Q^2_{t,h}(s,a,b) = \min\left\{\hat{Q}^2_{t,h}(s,a,b), Q^2_{t-1,h}(s,a,b)\right\} \geq Q_{*,h}(s,a,b)$, proving the left inequality in (D.5).

Based on (D.5), we can write

$$\begin{aligned}
&\sum_{t=1}^{T}\left(Q^2_{t,h}(s_{t,h},a_{t,h},b_{t,h}) - Q_{*,h}(s_{t,h},a_{t,h},b_{t,h})\right) \\
&\leq \sum_{t=1}^{T}\mathbb{E}_{s'\sim P(\cdot|s_{t,h},a_{t,h},b_{t,h})}\left[V^2_{t,h+1}(s') - V_{*,h+1}(s')\right] + \sum_{t=1}^{T} 2\mathsf{bns}^2_{\tau_{t,h}} \\
&\hspace{20em}(\text{define } \tau_{t,h} = n_{t,h}(s_{t,h},a_{t,h},b_{t,h})) \\
&\triangleq \sum_{t=1}^{T}\left(V^2_{t,h+1}(s_{t,h+1}) - V_{*,h+1}(s_{t,h+1}) + \varepsilon_{t,h}\right) + \sum_{t=1}^{T} 2\mathsf{bns}^2_{\tau_{t,h}} \\
&\hspace{20em}(\text{define } \varepsilon_{t,h} \text{ to be the difference})
\end{aligned}$$

Since $\varepsilon_{t,h}$ is zero-mean, by Lemma D.7,

$$\sum_{t=1}^{T}\varepsilon_{t,h} = \mathcal{O}\left(H\sqrt{T\log(T/\delta)}\right).$$

Besides,

$$\sum_{t=1}^{T}\mathsf{bns}^2_{\tau_{t,h}} = \mathcal{O}\left(\sum_{t=1}^{T} H\sqrt{\frac{S\log(T/\delta)}{n_{t,h}(s_{t,h},a_{t,h},b_{t,h})^+}}\right) = \mathcal{O}\left(HS\sqrt{ABT\log(T/\delta)}\right).$$

Combining the three bounds above proves the second conclusion in the lemma. ■

**Proof of Lemma 5.11:** We use induction to show the desired inequality. Again, the order of induction is from $t = 1$ to $t = T$, and (within each $t$) from $h = H$ to $h = 1$. When $t = 1$, $Q^1_{1,h}(s,a) = H = \max_b Q^2_{1,h}(s,a,b)$.

Suppose that the inequality holds for all $(t',h')$ with $t' < t$, or $t' = t$ and $h' > h$. Let $\tau = n_{t,h}(s,a)$, and let $1 \leq t_1 < t_2 < \cdots < t_\tau < t$ be the episodes in which $(s,a)$ is visited at step $h$. By the update rule of $Q^1_{t,h}(\cdot,\cdot)$, we have

$$Q^1_{t,h}(s,a)$$

$$= \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( r_{t_i,h} + V^1_{t_i,h+1}(s_{t_i,h+1}) + \mathsf{bns}^1_i \right)$$

(define $\alpha_\tau^i = \alpha_i \Pi_{j=i+1}^\tau (1-\alpha_j)$ for $1 \le i \le \tau$ and $\alpha_\tau^0 = \Pi_{j=1}^\tau (1-\alpha_j)$)

$$\ge \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( R(s,a,b_{t_i,h}) + \sum_{s'} P(\cdot|s,a,b_{t_i,h}) V^1_{t_i,h+1}(s') \right) + \frac{1}{2}\mathsf{bns}^1_\tau$$

(see the explanation below indexed $\star$)

$$\ge \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( R(s,a,b_{t_i,h}) + \sum_{s'} P(\cdot|s,a,b_{t_i,h}) V^2_{t_i,h+1}(s') \right) + \frac{1}{2}\mathsf{bns}^1_\tau$$

(by the induction hypothesis)

$$\ge \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( \hat{R}_{t_i,h}(s,a,b_{t_i,h}) + \sum_{s'} \hat{P}_{t_i,h}(\cdot|s,a,b_{t_i,h}) V^2_{t_i,h+1}(s') - \mathsf{bns}^2_{\xi_i} \right) + \frac{1}{2}\mathsf{bns}^1_\tau$$

(define $\xi_i = n_{t_i,h}(s,a,b_{t_i,h})$ and use (D.7) and (D.8))

$$\ge \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i Q^2_{t_i,h}(s,a,b_{t_i,h}) - 2\sum_{i=1}^{\tau} \alpha_\tau^i \mathsf{bns}^2_{\xi_i} + \frac{1}{2}\mathsf{bns}^1_\tau$$

(by the definition of $Q^2_{t,h}(s,a,b)$)

$$\ge \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \max_b Q^2_{t_i,h}(s,a,b)$$

(see the explanation below indexed $\star\star$)

$$\ge \max_b Q^2_{t,h}(s,a,b)$$

(because $Q^2_{t,h}$ is non-increasing in $t$ and $\sum_{i=0}^\tau \alpha_\tau^i = 1$)

In the first inequality ($\star$), we use the fact that $\sum_{i=1}^\tau \alpha_\tau^i \mathsf{bns}^1_i \ge \mathsf{bns}^1_\tau$ by the first item in Lemma D.9, and that

$$\left| \sum_{i=1}^{\tau} \alpha_\tau^i \left( R(s,a,b_{t_i,h}) + \sum_{s'} P(s'|s,a,b_{t_i,h}) V^1_{t_i,h+1}(s') - r_{t_i,h} - V^1_{t_i,h+1}(s_{t_i,h+1}) \right) \right|$$
$$\le \frac{1}{2} c' H \sqrt{\frac{HS\log(T/\delta)}{\tau^+}} \le \frac{1}{2}\mathsf{bns}^1_\tau \tag{D.10}$$

for some universal constant $c' > 0$ by Lemma D.10. In the penultimate inequality ($\star\star$), we first use the selection rule of $b_{t,h} = \operatorname{argmax}_b Q^2_{t,h}(s_{t,h}, a_{t,h}, b)$, and then use the following Lemma D.5 to bound

$$2\sum_{i=1}^{\tau} \alpha_\tau^i \mathsf{bns}^2_{\xi_i} = 2cH\sqrt{S\log(T/\delta)} \sum_{i=1}^{\tau} \alpha_\tau^i \frac{1}{\sqrt{n_{t_i,h}(s,a,b_{t_i,h})^+}}$$

$$\leq 2cH\sqrt{S\log(T/\delta)} \times 4\sqrt{\frac{BH}{n_{t,h}(s,a)^+}}$$

$$\leq \frac{1}{2}c'H\sqrt{\frac{HSB\log(T/\delta)}{n_{t,h}(s,a)^+}} = \frac{1}{2}\mathsf{bns}^1_\tau.$$

■

**Lemma D.5:** *Let $\{\tau_1, \ldots, \tau_B\}$ be non-negative integers such that $\sum_{b=1}^{B} \tau_b = \tau$. Define for all $b = 1, \ldots, B$:*

$$Y_{bi} = \frac{1}{\sqrt{(i-1)^+}}, \qquad \text{for } i = 1, 2, \ldots, \tau_b.$$

*Let $\{Z_1, Z_2, \ldots, Z_\tau\}$ be any permutation of*

$$\{Y_{11}, Y_{12}, \ldots, Y_{1\tau_1}, Y_{21}, Y_{22}, \ldots, Y_{2\tau_2}, \ldots\ldots Y_{B1}, Y_{B2}, \ldots, Y_{B\tau_B}\}$$

*Then*

$$\sum_{i=1}^{\tau} \alpha_\tau^i Z_i \leq 4\sqrt{\frac{BH}{\tau^+}}.$$

**Proof:** We write $i = \phi(b, j)$ if $Y_{bj}$ is mapped to $Z_i$. Also, define $\Phi(b) = \{\phi(b, j) : j \in [\tau_b]\}$ as the set of indices in $\{Z_i\}$ that are mapped from $\{Y_{b1}, \ldots, Y_{b\tau_b}\}$.

We first show the following claim: for all $b$,

$$\sum_{i\in\Phi(b)} \alpha_\tau^i Z_i \leq 2\sqrt{2\alpha_\tau \sum_{i\in\Phi(b)} \alpha_\tau^i}. \tag{D.11}$$

To show (D.11), observe that the left-hand side is equal to

$$\sum_{i\in\Phi(b)} \alpha_\tau^i Z_i = \sum_{j=1}^{\tau_b} \alpha_\tau^{\phi(b,j)} Y_{bj} = \sum_{j=1}^{\tau_b} \alpha_\tau^{\phi(b,j)} \frac{1}{\sqrt{(j-1)^+}} \leq \sum_{j=1}^{\tau_b} \alpha_\tau^{\phi(b,j)} \sqrt{\frac{2}{j}}. \tag{D.12}$$

By the definition of $\alpha_\tau^i$, we have $\alpha_\tau^i \leq \alpha_\tau$ for any $i$. We see that the last expression in (D.12) is upper bounded by the optimal solution of the following programming:

$$\begin{aligned}\max_{\beta_j} &\sum_{j=1}^{\tau_b} \beta_j \sqrt{\frac{2}{j}} \\ \text{s.t. } &\sum_{j=1}^{\tau_b} \beta_j \le \sum_{i\in\Phi(b)} \alpha_\tau^i \\ &0 \le \beta_j \le \alpha_\tau \quad \forall j\end{aligned}$$

This programming exhibits a greedy solution that sets $\beta_j = \alpha_\tau$ for $j \le j^\star \triangleq \left\lfloor \frac{1}{\alpha_\tau} \sum_{i\in\Phi(b)} \alpha_\tau^i \right\rfloor$, $\beta_j = \sum_{i\in\Phi(b)} \alpha_\tau^i - \alpha_\tau j^\star$ for $j = j^\star + 1$, and $\beta_j = 0$ otherwise. The optimal value of this solution is upper bounded by

$$\alpha_\tau \sum_{j=1}^{j^\star} \sqrt{\frac{2}{j}} + \left( \sum_{i\in\Phi(b)} \alpha_\tau^i - \alpha_\tau j^\star \right) \sqrt{\frac{2}{j^\star + 1}} \le \alpha_\tau \int_0^{\frac{1}{\alpha_\tau}\sum_{i\in\Phi(b)} \alpha_\tau^i} \sqrt{\frac{2}{x}} \mathrm{d}x = 2\sqrt{2\alpha_\tau \sum_{i\in\Phi(b)} \alpha_\tau^i},$$

showing (D.11). To get the final bound, we sum this bound over $b$ and use the definition of $\alpha_\tau$:

$$\begin{aligned}\sum_{i=1}^{\tau} \alpha_\tau^i Z_i &= \sum_{b=1}^{B} \sum_{i\in\Phi(b)} \alpha_\tau^i Z_i \le \sum_{b=1}^{B} 2\sqrt{2\alpha_\tau \sum_{i\in\Phi(b)} \alpha_\tau^i} \\ &\le 2\sqrt{2B\alpha_\tau \sum_{i=1}^{\tau} \alpha_\tau^i} = 2\sqrt{\frac{2B(H+1)}{H+\tau}} \le 4\sqrt{\frac{BH}{\tau^+}},\end{aligned}$$

where in the second inequality we use the AM-GM inequality and in the last equality we use $\sum_{i=1}^{\tau} \alpha_\tau^i = 1$ for $\tau \ge 1$. ■

**Proof of Lemma 5.12:** Fix $t, h, s, a$. Let $\tau = n_{t,h}(s,a)$, and let $1 \le t_1 < t_2 < \cdots < t_\tau < t$ be the episodes in which $(s,a)$ is visited at layer $h$. By the update rule of $Q^1_{t,h}(\cdot,\cdot)$, we have

$$\begin{aligned}&Q^1_{t,h}(s,a) \\ &= \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( r_{t_i,h} + V^1_{t_i,h+1}(s_{t_i,h+1}) + \mathsf{bns}_i^1 \right)\end{aligned}$$

$$
\begin{aligned}
&\leq \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( R(s, a, b_{t_i,h}) + \mathbb{E}_{s' \sim P(\cdot|s,a,b_{t_i,h})} \left[ V_{t_i,h+1}^1(s') \right] \right) + \mathcal{O}(\mathsf{bns}_\tau^1) \\
&\qquad \text{(by Lemma D.10 and that } \textstyle\sum_{i=1}^{\tau} \alpha_\tau^i \mathsf{bns}_i^1 \leq 2\mathsf{bns}_\tau^1 \text{ by the first item in Lemma D.9)} \\
&= \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( R(s, a, b_{t_i,h}) + \mathbb{E}_{s' \sim P(\cdot|s,a,b_{t_i,h})} \left[ V_{*,h+1}(s') \right] \right) \\
&\qquad + \sum_{i=1}^{\tau} \alpha_\tau^i \left( \mathbb{E}_{s' \sim P(\cdot|s,a,b_{t_i,h})} \left[ V_{t_i,h+1}^1(s') - V_{*,h+1}(s') \right] \right) + \mathcal{O}(\mathsf{bns}_\tau^1) \\
&= \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i Q_{*,h}(s, a, b_{t_i,h}) + \sum_{i=1}^{\tau} \alpha_\tau^i \left( \mathbb{E}_{s' \sim P(\cdot|s,a,b_{t_i,h})} \left[ V_{t_i,h+1}^1(s') - V_{*,h+1}(s') \right] \right) + \mathcal{O}(\mathsf{bns}_\tau^1) \\
&\leq \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i Q_{*,h}(s, a, b_{t_i,h}) + \sum_{i=1}^{\tau} \alpha_\tau^i \left( V_{t_i,h+1}^1(s_{t_i,h+1}) - V_{*,h+1}(s_{t_i,h+1}) \right) + \mathcal{O}(\mathsf{bns}_\tau^1) \\
&\qquad \text{(by Lemma D.10)}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&Q_{t,h}^1(s, a) - Q_{*,h}(s, a) \\
&= \alpha_\tau^0 (H - Q_{*,h}(s, a)) + \sum_{i=1}^{\tau} \alpha_\tau^i \left( Q_{*,h}(s, a, b_{t_i,h}) - Q_{*,h}(s, a) \right) \\
&\qquad + \sum_{i=1}^{\tau} \alpha_\tau^i \left( V_{t_i,h+1}^1(s_{t_i,h+1}) - V_{*,h+1}(s_{t_i,h+1}) \right) + \mathcal{O}(\mathsf{bns}_\tau^1) \\
&\leq \alpha_\tau^0 H + \sum_{i=1}^{\tau} \alpha_\tau^i \left( V_{t_i,h+1}^1(s_{t_i,h+1}) - V_{*,h+1}(s_{t_i,h+1}) \right) + \mathcal{O}(\mathsf{bns}_\tau^1) \qquad \text{(D.13)}
\end{aligned}
$$

Now consider the cumulative sum, and define $t_i(s, a)$ to be the index of the episode when it is the $i$-th time $(s, a)$ is visited at layer $h$.

$$
\begin{aligned}
&\sum_{t=1}^{T} \left( Q_{t,h}^1(s_{t,h}, a_{t,h}) - Q_{*,h}(s_{t,h}, a_{t,h}) \right) \\
&= \sum_{s,a} \sum_{i=1}^{n_{T+1}(s,a)} \left( Q_{t_i(s,a),h}^1(s_{t_i(s,a),h}, a_{t_i(s,a),h}) - Q_{*,h}(s_{t_i(s,a),h}, a_{t_i(s,a),h}) \right) \\
&\leq \sum_{s,a} \sum_{i=1}^{n_{T+1}(s,a)} \left( \alpha_i^0 H + \sum_{j=1}^{i-1} \alpha_i^j \left( V_{t_j(s,a),h+1}^1(s_{t_j(s,a),h+1}) - V_{*,h+1}(s_{t_j(s,a),h+1}) \right) + \mathsf{bns}_{i-1}^1 \right) \\
&\qquad \text{(by (D.13))}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{s,a} \sum_{i=1}^{n_{T+1}(s,a)} \alpha_i^0 H + \sum_{s,a} \sum_{j=1}^{n_{T+1}(s,a)-1} \sum_{i=j+1}^{n_{T+1}(s,a)} \alpha_i^j \left( V^1_{t_j(s,a),h+1}(s_{t_j(s,a),h+1}) - V_{*,h+1}(s_{t_j(s,a),h+1}) \right) \\
&\qquad + \sum_{s,a} \sum_{i=1}^{n_{T+1}(s,a)} \mathcal{O}\left( \sqrt{\frac{H^3 SB \log(T/\delta)}{\max\{i-1,1\}}} \right) \\
&\le HSA + \sum_{s,a} \sum_{j=1}^{n_{T+1}(s,a)-1} \left(1+\frac{1}{H}\right) \left( V^1_{t_j(s,a),h+1}(s_{t_j(s,a),h+1}) - V_{*,h+1}(s_{t_j(s,a),h+1}) \right) \\
&\qquad + \mathcal{O}\left( \sqrt{H^3 S^2 ABT} \right) \qquad\qquad \text{(by the third item of Lemma D.9)} \\
&\le \left(1+\frac{1}{H}\right) \sum_{t=1}^{T} \left( V^1_{t,h+1}(s_{t,h+1}) - V^1_{*,h+1}(s_{t,h+1}) \right) + \mathcal{O}\left( \sqrt{H^3 S^2 ABT} + HSA \right).
\end{aligned}
$$

■

**Proof of Corollary 5.13:** By Lemma 5.12 and the fact that $V^1_{t,h}(s_{t,h}) = Q^1_{t,h}(s_{t,h}, a_{t,h})$, we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \left( Q^1_{t,h}(s_{t,h}, a_{t,h}) - Q_{*,h}(s_{t,h}, a_{t,h}) \right) \\
&\le \left(1+\frac{1}{H}\right) \sum_{t=1}^{T} \left( Q^1_{t,h+1}(s_{t,h+1}, a_{t,h+1}) - Q_{*,h+1}(s_{t,h+1}, a_{t,h+1}) \right) \\
&\qquad + \mathcal{O}\left( \sqrt{H^3 S^2 ABT} + HSA \right),
\end{aligned}
$$

which gives

$$
\begin{aligned}
\sum_{t=1}^{T} \left( Q^1_{t,h}(s_{t,h}, a_{t,h}) - Q_{*,h}(s_{t,h}, a_{t,h}) \right) &\le H \times \left(1+\frac{1}{H}\right)^H \times \mathcal{O}\left( \sqrt{H^3 S^2 ABT} + HSA \right) \\
&= \mathcal{O}\left( \sqrt{H^5 S^2 ABT} + H^2 SA \right)
\end{aligned}
$$

by expanding the recursion. ■

## D.5 Proofs of the Structural Results of the Matching Problem

### D.5.1 Proof of Lemma 5.15

We only prove the case for the first sub-step. The proof of the second sub-step is the same. Notice that whenever $Z_t \neq 0$ or $Y_t \neq 0$, $S_t$ is in the CI – if $Z_t = U_t S_t \neq 0$, it means $U_t = 1$ and $Z_t = S_t$, and since $Z_t$ is CI, so is $S_t$; on the other hand, if $Y_t = \mathbf{1}\{S_t = A_t^1 = A_t^2\} = 1 \neq 0$, then both agents can deduce $S_t$ from their private information, namely past actions $A_t^1$ and $A_t^2$.

Now the CI is $(Z_{1:t-1}, Y_{1:t-1}, S_{\bar{t}^{\mathrm{i}}})$. Note that since we are considering an MDP, given a strategy over time (that is a $g_{1:\infty}$), $S_{\bar{t}^{\mathrm{i}}}$ will be the sufficient statistics of all variables prior to $\bar{t}^{\mathrm{i}}$ when it comes to the statistics of any variables after $\bar{t}^{\mathrm{i}}$. Consider an optimal prescription $\gamma_t^{\mathrm{i}*} \in \Omega(\Omega(S_{1:t}) \times \Omega(A^1_{1:t-1}) \to \Omega(U_t))$ and two admissible histories $m_1, m_2 \in \Omega(S_{1:\bar{t}^{\mathrm{i}}-1}) \times \Omega(A^1_{1:\bar{t}^{\mathrm{i}}-1})$ and $m_3 \in \Omega(S_{\bar{t}^{\mathrm{i}}:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1})$ so that both $(m_1, m_3)$ and $(m_2, m_3)$ are consistent with $(Z_{1:t-1}, Y_{1:t-1}, S_{\bar{t}^{\mathrm{i}}})$ (which means the probabilities of the two histories are positive given the CI). Denote the value of the first equation of (5.18) before taking the supremum as $V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}})$. Then we claim it must be that

$$V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}*}(m_1, m_3), M_{1:t} = (m_1, m_3)) = V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}*}(m_2, m_3), M_{1:t} = (m_2, m_3)). \quad \text{(D.14)}$$

We use proof by contradiction. Suppose the left hand side is larger. Then we can construct a new prescription $\gamma_t^{\mathrm{i}**}$ such that it is the same as $\gamma_t^{\mathrm{i}*}$ for all histories except that $\gamma_t^{\mathrm{i}**}(m_2, m_3) = \gamma_t^{\mathrm{i}*}(m_1, m_3)$. Then when $(m_2, m_3)$ happens, the future dynamics of adopting $\gamma_t^{\mathrm{i}**}$ will be exactly the same as when $(m_1, m_3)$ happens and $\gamma_t^{\mathrm{i}*}$ is adopted since $S_{\bar{t}^{\mathrm{i}}}$ is known; and since $(m_2, m_3)$ happens with a positive probability, we have $V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}**}) > V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}*})$, a contradiction. We can then construct an optimal strategy $\gamma_t^{\mathrm{i}***} \in \Omega(\Omega(S_{\bar{t}^{\mathrm{i}}+1:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1}) \to \Omega(U_t))$ from $\gamma_t^{\mathrm{i}*}$ such that for any $m' \in \Omega(S_{\bar{t}^{\mathrm{i}}+1:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1}) \to \Omega(U_t)$, we choose arbitrary consistent $m \in \Omega(S_{1:\bar{t}^{\mathrm{i}}-1}) \times \Omega(A^1_{1:\bar{t}^{\mathrm{i}}-1})$ such that $\gamma_t^{\mathrm{i}***}(m') = \gamma_t^{\mathrm{i}*}(m, m')$, and from (D.14) we have $V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}*}) = V_t^{\mathrm{i}}(\pi_t^{\mathrm{i}}, \gamma_t^{\mathrm{i}***})$, which implies $\gamma_t^{\mathrm{i}***}$ is also an optimal prescription.

### D.5.2 Proof of Lemma 5.17

Again we only prove the case for the first sub-step. Note that given $(Z_{1:t-1}, Y_{1:t-1})$ and the fact that $\bar{t}^{\mathrm{i}}$ is well-defined, $S_{\bar{t}^{\mathrm{i}}}$ is also in the CI, so that

$$
\begin{aligned}
&\Pi_t^{\mathrm{i}}(m, m') \\
&= \mathbb{P}^{g_{1:t-1}}(M_{1:t} = (m, m') | Z_{1:t-1}, Y_{1:t-1}) \\
&= \mathbb{P}^{g_{1:t-1}}(M_{1:t} = (m, m') | S_{\bar{t}^{\mathrm{i}}}, Z_{1:t-1}, Y_{1:t-1}) \\
&= \mathbb{P}^{g_{1:\bar{t}^{\mathrm{i}}-1}}(M_{1:\bar{t}^{\mathrm{i}}} = m | S_{\bar{t}^{\mathrm{i}}}, Z_{1:t-1}, Y_{1:t-1}) \cdot \mathbb{P}^{g_{\bar{t}^{\mathrm{i}}:t-1}}(M_{\bar{t}^{\mathrm{i}}+1:t} = m' | M_{1:\bar{t}^{\mathrm{i}}} = m, S_{\bar{t}^{\mathrm{i}}}, Z_{1:t-1}, Y_{1:t-1}) \\
&= \mathbb{P}^{g_{1:\bar{t}^{\mathrm{i}}-1}}(M_{1:\bar{t}^{\mathrm{i}}} = m | Z_{1:\bar{t}^{\mathrm{i}}-1}, Y_{1:\bar{t}^{\mathrm{i}}-1}) \cdot \mathbb{P}^{g_{\bar{t}^{\mathrm{i}}:t-1}}(M_{\bar{t}^{\mathrm{i}}+1:t} = m' | S_{\bar{t}^{\mathrm{i}}}, Z_{\bar{t}^{\mathrm{i}}:t-1}, Y_{\bar{t}^{\mathrm{i}}:t-1}) \\
&= \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}}(m) \cdot \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}(m').
\end{aligned}
$$

The penultimate equality follows from two crucial facts. First, $S_{\bar{t}^{\mathrm{i}}}$ is the sufficient statistics for the previous variables given the strategies. Second, using the strategy space reduction from Lemma 5.15, the strategies also do not depend on $M_{1:\bar{t}^{\mathrm{i}}} = m$; hence, it can be removed from the conditioning.

### D.5.3 An Intermediate Proposition D.6 and Its Proof

**Proposition D.6:** *Consider time step $t$. Given the CI $(Z_{1:t-1}, Y_{1:t-1})$, the DP equation for the first sub-step can be written as*

$$
V_t^{\mathrm{i}}(\pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}) = \sup_{\gamma_t^{\mathrm{i}}} \mathbb{E}\left[ -c\gamma_t^{\mathrm{i}}(M_{\bar{t}^{\mathrm{i}}+1:t}) + V_t^{\mathrm{ii}}(\eta^{\mathrm{i}}_{\bar{t}^{\mathrm{i}},t}(\Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}, \gamma_t^{\mathrm{i}}, Z_t)) \middle| \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t} = \pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t} \right], \tag{D.15}
$$

*where $\gamma_t^{\mathrm{i}} \in \Omega(\Omega(S_{\bar{t}^{\mathrm{i}}+1:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1}) \to \Omega(U_t))$ and*

$$
\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}+1:t}(m) = \eta^{\mathrm{i}}_{\bar{t}^{\mathrm{i}},t}(\Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}, \gamma_t^{\mathrm{i}}, Z_t)(m) = \frac{\Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}(m)\mathbb{P}(Z_t | M_{\bar{t}^{\mathrm{i}}+1:t} = m, \gamma_t^{\mathrm{i}})}{\sum_{m'} \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}(m')\mathbb{P}(Z_t | M_{\bar{t}^{\mathrm{i}}+1:t} = m', \gamma_t^{\mathrm{i}})}, \tag{D.16}
$$

*where $m, m' \in \Omega(M_{\bar{t}^{\mathrm{i}}+1:t})$. Similarly, the DP equation for the second sub-step can be written as*

$$
V_t^{\mathrm{ii}}(\pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}) = \sup_{\gamma_t^{\mathrm{ii}}} \mathbb{E}\left[ Y_t(S_t, \gamma_t^{\mathrm{ii}}(M_{\bar{t}^{\mathrm{ii}}+1:t})) + \lambda V^{\mathrm{i}}_{t+1}(\eta^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}},t}(\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}, \gamma_t^{\mathrm{ii}}, Y_t)) \middle| \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t} = \pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t} \right], \tag{D.17}
$$

*where* $\gamma_t^{\mathrm{ii}} \in \Omega(\Omega(M_{\bar{t}^{\mathrm{ii}}+1:t}) \to \Omega(A_t))$ *and*

$$\begin{aligned}\Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{ii}}+1:t+1}(m,m') &= \eta^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}},t}(\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}, \gamma_t^{\mathrm{ii}}, Y_t)(m,m') \\ &= \frac{\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m)\mathbb{P}(M_{t+1}=m', Y_t | M_{\bar{t}^{\mathrm{ii}}+1:t}=m, \gamma_t^{\mathrm{ii}})}{\sum_{\bar{m},\bar{m}'} \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(\bar{m})\mathbb{P}(M_{t+1}=\bar{m}', Y_t | M_{\bar{t}^{\mathrm{ii}}+1:t}=\bar{m}, \gamma_t^{\mathrm{ii}})},\end{aligned} \tag{D.18}$$

*where* $m, \bar{m} \in \Omega(M_{\bar{t}^{\mathrm{ii}}+1:t})$ *and* $m', \bar{m}' \in \Omega(M_{t+1})$*. If* $Z_t = 0$*, then* $\bar{t}^{\mathrm{ii}} = \bar{t}^{\mathrm{i}} \le t-1$ *and* $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m) = \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}+1:t}(m)$*; otherwise,* $\bar{t}^{\mathrm{ii}} = t > t-1 \ge \bar{t}^{\mathrm{i}}$*, then we let* $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m)$ *to be the* null information state $\Pi^{\mathrm{ii}}_0$*, which is a distribution on nothing*[1]*. Similarly, if* $Y_t = 0$*, then* $\overline{t+1}^{\mathrm{i}} = \bar{t}^{\mathrm{ii}} \le t$ *and* $\Pi^{\mathrm{i}}_{\overline{t+1}^{\mathrm{i}}+1:t+1}(m) = \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{ii}}+1:t+1}(m)$*; otherwise,* $\overline{t+1}^{\mathrm{i}} = t \ge \bar{t}^{\mathrm{ii}}$ *and we split* $\Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{ii}}+1:t+1}(m,m') = \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{ii}}+1:\overline{t+1}^{\mathrm{i}}}(m) \cdot \Pi^{\mathrm{i}}_{\overline{t+1}^{\mathrm{i}}+1:t+1}(m')$ *and let the second half be the new* $\Pi^{\mathrm{i}}_{\overline{t+1}^{\mathrm{i}}+1:t+1}(m')$*, throwing away the first half*[2]*.*

**Proof:** First, from Lemma 5.15 we can restrict attention to $\gamma_t^{\mathrm{i}} \in \Omega(\Omega(S_{\bar{t}^{\mathrm{i}}+1:t}) \times \Omega(A^1_{\bar{t}^{\mathrm{i}}:t-1}) \to \Omega(U_t))$. Notice that

$$\begin{aligned}&\Pi^{\mathrm{i}}_t(m,m') \\ &= \mathbb{P}^{g_{1:\bar{t}^{\mathrm{i}}-1}, g^{\mathrm{i}}_{\bar{t}}}(M_{1:\bar{t}^{\mathrm{i}}} = m | Z_{1:\bar{t}^{\mathrm{i}}-1}, Y_{1:\bar{t}^{\mathrm{i}}-1}) \cdot \mathbb{P}^{g_{\bar{t}^{\mathrm{i}}:t-1}}(M_{\bar{t}^{\mathrm{i}}+1:t} = m' | S_{\bar{t}^{\mathrm{i}}}, Z_{\bar{t}^{\mathrm{i}}:t}, Y_{\bar{t}^{\mathrm{i}}:t-1}) \\ &= \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}}(m) \cdot \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t}(m'),\end{aligned}$$

as $\bar{t}^{\mathrm{i}} \le t-1$. This can be seen as a variant of (5.23) from Lemma 5.17. We can use this and (5.24) from Lemma 5.17 and split all future information states $\Pi^{\mathrm{i}}_{t'}(m,m') = \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}}(m) \cdot \Pi^{\mathrm{i}}_{\bar{t}^{\mathrm{i}}+1:t'}(m')$ and $\Pi^{\mathrm{ii}}_{t'}(m,m') = \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}}(m) \cdot \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}+1:t'}(m')$. The first part of the states $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}}(m)$ is henceforth irrelevant and can be thrown away. The key is the update rule in (5.19) for the information state can be done without $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}}(m)$. Consider the right hand side of (5.19). The term $\mathbb{P}(Z_t | M_{1:t} = (m,m'), \gamma_t^{\mathrm{i}})$ is actually $\mathbb{P}(Z_t | M_{\bar{t}+1:t} = m', \gamma_t^{\mathrm{i}})$

---

[1] Intuitively, we just split $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}+1:t}(m,m') = \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}+1:\bar{t}^{\mathrm{ii}}}(m) \cdot \Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m')$ and throw away the first half and let the second half be $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m')$. Note that $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{ii}}+1:t}(m') = \Pi^{\mathrm{ii}}_{t+1:t}(m')$, so we actually throw away the whole $\Pi^{\mathrm{ii}}_{\bar{t}^{\mathrm{i}}+1:t}(m)$.

[2] The first half is *null* (which means there is no first half) if $\overline{t+1}^{\mathrm{i}} = t = \bar{t}^{\mathrm{ii}}$.

since $\gamma_t^{\rm i} \in \Omega(\Omega(S_{\bar t^{\rm i}+1:t}) \times \Omega(A^1_{\bar t^{\rm i}:t-1}) \to \Omega(U_t))$. Hence, it can be rewritten as

$$\begin{aligned}
\Pi_t^{\rm ii}(m,m') &= \Pi^{\rm ii}_{\bar t^{\rm i}}(m) \cdot \Pi^{\rm ii}_{\bar t^{\rm i}+1:t}(m') \\
&= \frac{\Pi_t^{\rm i}(m,m')\mathbb{P}(Z_t|M_{1:t} = (m,m'), \gamma_t^{\rm i})}{\sum_{\bar m, \bar m'} \Pi_t^{\rm i}(\bar m, \bar m')\mathbb{P}(Z_t|M_{1:t} = (\bar m, \bar m'), \gamma_t^{\rm i})} \\
&= \frac{\Pi^{\rm ii}_{\bar t^{\rm i}}(m) \cdot \Pi^{\rm i}_{\bar t^{\rm i}+1:t}(m') \cdot \mathbb{P}(Z_t|M_{\bar t+1:t} = m', \gamma_t^{\rm i})}{\sum_{\bar m, \bar m'} \Pi^{\rm ii}_{\bar t^{\rm i}}(\bar m) \cdot \Pi^{\rm i}_{\bar t^{\rm i}+1:t}(\bar m') \cdot \mathbb{P}(Z_t|M_{\bar t+1:t} = \bar m', \gamma_t^{\rm i})} \\
&= \Pi^{\rm ii}_{\bar t^{\rm i}}(m) \cdot \frac{\Pi^{\rm i}_{\bar t^{\rm i}+1:t}(m')\mathbb{P}(Z_t|M_{\bar t+1:t} = m', \gamma_t^{\rm i})}{\sum_{\bar m'} \Pi^{\rm i}_{\bar t^{\rm i}+1:t}(\bar m') \cdot \mathbb{P}(Z_t|M_{\bar t+1:t} = \bar m', \gamma_t^{\rm i})},
\end{aligned}$$

which simplifies to (D.16). ■

### D.5.4 Proof of Proposition 5.18

We need the stationarity of the model here – in every time step, the transition kernel, observation binary symmetric channel (BSC), as well as the reward/cost function are exactly the same. Hence, the dynamic equations (D.15) and (D.17) in Proposition D.6 do not depend on the values of $\bar t^{\rm i}$ and $\bar t^{\rm ii}$. Consider $V_t^{\rm i}(\pi^{\rm i}_{\bar t^{\rm i}+1:t})$ in Proposition D.6 and an optimal prescription $\gamma_t^{\rm i*}$ that achieves it. If we have some $t' \neq t$, the CI so that $t' - \bar t' = t - \bar t$, and the same belief state $\pi^{\rm i}_{\bar t'^{\rm i}+1:t'} = \pi^{\rm i}_{\bar t^{\rm i}+1:t}$. Then $\gamma_t^{\rm i*}$ will work as the optimal prescription here, since it will lead to the same dynamics as the old $t$ problem, and one could not do any better because any improvement in the $t'$ problem works as an improvement in the old $t$ problem. We can see that the DP problems are time-invariant, and the DP equations only depend on the difference between $\bar t^{\rm i}$ and $t$ and between $\bar t^{\rm ii}$ and $t$.

## D.6 Proofs of Characterizing the Optimal Policy of the Matching Problem

### D.6.1 Proof of Theorem 5.19

A complete policy is of the form $g_{1:\infty} = (g^{\rm i}_{1:\infty}, g^{\rm ii,1}_{1:\infty}, g^{\rm ii,2}_{1:\infty})$. Denote the policy that U1 always matches the state for all $t$ as $g^{\rm ii,1\sharp}_{1:\infty}$. For the remaining parts of the complete policy, we can find an optimal one, denoted as $(g^{\rm i\sharp}_{1:\infty}, g^{\rm ii,2\sharp}_{1:\infty})$. Note that with $g^{\sharp}_{1:\infty} = (g^{\rm i\sharp}_{1:\infty}, g^{\rm ii,1\sharp}_{1:\infty}, g^{\rm ii,2\sharp}_{1:\infty})$, we have Corollary 5.20 and Remark 5.21, that is, the previous state $S_{t-1}$ is CI and that the policy is stationary, which means there is a $g^{\sharp}_{1:\infty}$ consisting of playing the same policy $g^{\sharp} = (g^{\mathrm{i}\sharp}, g^{\mathrm{ii},1\sharp}, g^{\mathrm{ii},2\sharp})$ at each time instant [3].

The following proof will adopt the indexing of 0 as the current time step. We merge the second sub-step into the first sub-step so that we consider $V^{\mathrm{i}}$ and $\pi^{\mathrm{i}}$ but drop their superscripts. We consider two policies. The first one is $g^{opt} = (g^{\mathrm{i}\sharp}, g^{\mathrm{ii},1\sharp}, g^{\mathrm{ii},2\sharp}, g^{\sharp}_{1:\infty})$, i.e., it plays $g^{\sharp}$ at the current time step and all the time steps afterwards (so it is just a shifted version of $g^{\sharp}_{1:\infty}$). The second one is $g^{dev} = (g^{\mathrm{i}}_0, g^{\mathrm{ii},1\flat}, g^{\mathrm{ii},2}_0, g^{\mathrm{i}}_1, g^{\mathrm{ii},1\sharp}, g^{\mathrm{ii},2}_1, g^{\sharp}_{1:\infty})$ for arbitrary $g^{\mathrm{i}}_0$, $g^{\mathrm{ii},2}_0$, $g^{\mathrm{i}}_1$, and $g^{\mathrm{ii},2}_1$, where $g^{\mathrm{ii},1\flat}$ is the policy that U1 does not match the state at the current time step. Denote the value function for an information state $\pi$ under a complete policy $g$ as $V(\pi, g)$, so that $V(\pi) = \sup_g V(\pi, g)$. We show that for arbitrary $\pi$ in the first sub-step, $V(\pi, g^{opt}) > V(\pi, g^{dev})$.

Recall that by Proposition 5.18, the information state in the first sub-step is of the form $\pi_{\tau,s,v}$ where $\tau \geq 0$, $s \in \{+1, -1\}$, and $v \in \{y, z\}$ (we drop the $^{\mathrm{i}}$ superscript). Then by Corollary 5.20, whenever $g^{\sharp}$ is adopted in the previous time step, we have $\tau = 1$ and $v = y$, which means there are only two possible information states, namely, $\pi_{1,+1,y}$ and $\pi_{1,-1,y}$ (actually four, if one further splits the cases of $y = 0$ and $y = 1$). By symmetries of the transition and observation probabilities, we must have $V(\pi_{1,+1,y}, g^{\sharp}_{1:\infty}) = V(\pi_{1,-1,y}, g^{\sharp}_{1:\infty})$. Moreover, if we denote the instantaneous reward for an information state $\pi$ under a stage policy $g$ as $R(\pi, g)$ [4], then

$$R(\pi_{1,+1,y}, g^{\sharp}) = R(\pi_{1,-1,y}, g^{\sharp}) = \xi \geq 0.5 > 0, \tag{D.19}$$

and

$$\begin{aligned} V(\pi_{1,+1,y}, g^{\sharp}_{1:\infty}) &= R(\pi_{1,+1,y}, g^{\sharp}) + \sum_{t=1}^{\infty} \lambda^t \left[(1-\epsilon) R(\pi^1_t, g^{\sharp}) + \epsilon R(\pi^2_t, g^{\sharp})\right] \\ &= \xi + \frac{\lambda}{1-\lambda}\xi = \frac{1}{1-\lambda}\xi, \end{aligned} \tag{D.20}$$

where one of $\pi^1_t$ and $\pi^2_t$ is $\pi_{1,+1,y}$ and the other is $\pi_{1,-1,y}$.

It now becomes clear that $V(\pi, g^{opt}) = \frac{1}{1-\lambda}\xi$ for arbitrary $\pi$ (we can assume $g^{\sharp}$ was adopted before the current time step as well). On the other hand, for arbitrary $\pi$,

---

[3] This does not mean the policy is *myopic* – the policy could still take later stages into account, it just does not change over time.

[4] A complete policy is that for the entire process, and a stage policy is only for a time step.

$g^{dev}$ receives 0 in the current time step, not more than $\lambda\xi$ in the next time step, and then $\frac{\lambda^2}{1-\lambda}\xi$ afterwards (since U1 starts to match again in the next time step). In the next time step, they could not perform better than $\xi$, since whatever they do could have been done by $g^\sharp$ as well. The fact that U1 does not match in the current step will change the dynamics; however, U2's a priory knowledge of $S_1$ is at best $\epsilon$ and $1-\epsilon$ (on $+1$ and $-1$, one way or the other) due to symmetry, and it might be worse if U1 does not communicate in both time steps. Therefore,

$$V(\pi, g^{dev}) \leq 0 + \lambda\xi + \frac{\lambda^2}{1-\lambda}\xi = \frac{\lambda}{1-\lambda}\xi < V(\pi, g^{opt}). \tag{D.21}$$

We conclude that they cannot gain more by U1's deviation of not matching in one step, but we already assume that the other parts of the policy $g^\sharp_{1:\infty}$ are optimal under $g^{\text{ii},1\sharp}_{1:\infty}$. Hence, by the Policy Improvement Theorem [115], $g^\sharp_{1:\infty}$ is an optimal policy.

### D.6.2 Proof of Corollary 5.20

We know from Theorem 5.19 that under an optimal strategy U1 always plays $A^1_t = S_t$. At time step $t$, U2 already observed $Y_{t-1} = \mathbf{1}\{S_{t-1} = A^1_{t-1} = A^2_{t-1}\} = \mathbf{1}\{S_{t-1} = A^2_{t-1}\}$, and can thus deduce $S_{t-1}$ by $S_{t-1} = A^2_{t-1} \cdot (-1)^{Y_{t-1}+1}$ from its private information $A^2_{t-1}$. Now that both agents know $S_{t-1}$, it is CI.

Recall the simplification results from Lemma 5.15 to Proposition 5.18 using $\bar{t}$ are basically done by exploiting the crucial fact of $S_{\bar{t}}$ being in the CI. Now that we have $S_{t-1}$ being in the CI, Proposition 5.18 directly follows with $\bar{t}^{\text{i}} = t-1$, and $\bar{t}^{\text{ii}} = t-1$ if $U_t = 0$ and $\bar{t}^{\text{ii}} = t$ if $U_t = 1$. These lead to the information state $\Pi^{\text{i}}_1$ in the first sub-step, and information state $\Pi^{\text{ii}}_1$ if $U_t = 0$ and $\Pi^{\text{ii}}_0$ if $U_t = 1$.

### D.6.3 Proof of Theorem 5.22

We will again adopt the 0 indexing and merge the second sub-step into the first sub-step while omitting the $^{\text{i}}$ superscript. As illustrated in Corollary 5.20, $S_{-1}$ is part of the CI, so that any history before $t = -1$ can be dropped. Furthermore, since $S_{-1}$ and $Y_{-1}$ alone characterize the statistics of $S_0$, $Z_{-1}$ which is also in the CI can be dropped as well. We hence write $\pi_{s,y}$ to denote the realization of the information state $\Pi$ ($\Pi^{\text{i}}_1$ to be precise), where $s \in \{+1, -1\}$ is the realization of $S_{-1}$ and $y \in \{0, 1\}$

is the realization of $Y_{-1}$. As stated in (D.19), due to symmetries, all four states have the same instantaneous reward and long term value under an optimal policy, and this reward can be achieved by the same structure of policy again because of the symmetric transition structure. Specifically, for all four realizations of $S_{-1}$ and $Y_{-1}$, U2's prior (or the coordinator's prior, or the prior obtained from CI) of $S_0$ is always $\epsilon$ on one of the state and $1-\epsilon$ on the other:

$$S_0 = \begin{cases} S_{-1}\cdot(-1)^{Y_{-1}+1}, & \text{with probability } 1-\epsilon, \\ S_{-1}\cdot(-1)^{Y_{-1}}, & \text{with probability } \epsilon. \end{cases} \tag{D.22}$$

Let us consider U1's policy $U_0 = g^{\mathrm{i}}(S_{-1}, Y_{-1}, S_0)$ first. Let $\bar{S}_0 = S_{-1}\cdot(-1)^{Y_{-1}+1}$. Then U2's prior of $S_0$ is $1-\epsilon$ on $\bar{S}_0$, and $\epsilon$ on $-\bar{S}_0$. This means the case of $S_{-1} = +1$ and $Y_{-1} = 1$ will lead to the same prior as the case of $S_{-1} = -1$ and $Y_{-1} = 0$ and can share the same optimal policy. Moreover, the case of $S_{-1} = +1$ and $Y_{-1} = 0$ also shares the same prior as the case of $S_{-1} = +1$ and $Y_{-1} = 1$ if one reverses the states $S_0 = +1$ and $S_0 = -1$; hence, one could just adopt the previous policy with the two states reversed. In sum, we need only one optimal policy structure, which in general is of the form

$$U_0 = g^{\mathrm{i}}_{a,b}(S_0, \bar{S}_0) = \begin{cases} 1, & \text{if } S_0 = \bar{S}_0, \text{with probability } a, \\ 0, & \text{if } S_0 = \bar{S}_0, \text{with probability } 1-a, \\ 1, & \text{if } S_0 = -\bar{S}_0, \text{with probability } b, \\ 0, & \text{if } S_0 = -\bar{S}_0, \text{with probability } 1-b. \end{cases} \tag{D.23}$$

Define $g^{\mathrm{i}}[j,k]$ to be the policy that U1 plays $U_0 = j$ deterministically when $S_0 = \bar{S}_0$ and $U_0 = k$ deterministically when $S_0 = -\bar{S}_0$, where $j, k \in \{0, 1\}$. Then the policy in (D.23) can be seen as a "mixed policy" of the four "pure policy" $g^{\mathrm{i}}[0,0]$, $g^{\mathrm{i}}[0,1]$, $g^{\mathrm{i}}[1,0]$, and $g^{\mathrm{i}}[1,1]$; we know from stochastic control that the value of $g^{\mathrm{i}}_{a,b}$ will be a linear combination of those of $g^{\mathrm{i}}[0,0]$, $g^{\mathrm{i}}[0,1]$, $g^{\mathrm{i}}[1,0]$, and $g^{\mathrm{i}}[1,1]$. It is clear that one of the 'pure policy" must be optimal. Note that $g^{\mathrm{i}}[1,1]$ will never be the unique optimal policy. This is because both policies $g^{\mathrm{i}}[0,1]$ and $g^{\mathrm{i}}[1,0]$ can transmit one bit of information so that U2 can completely decode $S_0$ from $U_0$ and thus $S_0$ becomes CI.

Adopting the policy $g^{\mathrm{i}}[1,1]$ could potentially increase the communication cost if $c > 0$ and $\epsilon > 0$, but cannot increase the expected reward, since with $g^{\mathrm{i}}[0,1]$ and $g^{\mathrm{i}}[1,0]$ they will already be able to getting the matching reward 1 for sure. Although when U1 does communicate ($U_0 = 1$), U2 directly know the state from $Z_0 = U_0 S_0 = S_0$, this feature is not necessary – U2 can simply infer $S_0$ from $U_0$ and the model parameters.

It is clear that when $g^{\mathrm{i}}[0,1]$ or $g^{\mathrm{i}}[1,0]$ is adopted, $S_0$ also becomes CI, and U2 simply matches it – not matching $S_0$ will only lead to a zero instantaneous reward, and this does not increase the "continuation reward" (expected future discounted reward) either, since it is fixed as mentioned previously. On the other hand, when $g^{\mathrm{i}}[0,0]$ is adopted, U2 can only make a maximum likelihood estimation based on the knowledge of the realizations of $\bar{S}_0$ and $O_0$; that is, it compares $\mathbb{P}(S_0 = \bar{S}_0 | \bar{S}_0, O_0)$ and $\mathbb{P}(S_0 = -\bar{S}_0 | \bar{S}_0, O_0)$, and matches the $S_0$ with the higher probability. The expected instantaneous reward of adopting $g^{\mathrm{i}}[0,0]$ is thus $\mathbb{P}(Y_0 = 1)$, which falls into the following cases. If U2 observes $O_0 = \bar{S}_0 = S_{-1} \cdot (-1)^{Y_{-1}+1}$ (the $1-\epsilon$ branch), there is a chance of $(1-\epsilon)\beta$ where its observation is actually correct, and a chance of $\epsilon(1-\beta)$ where it is wrong. Hence, to maximize $\mathbb{P}(Y_0 = 1)$, U2 only follows the observation if $(1-\epsilon)\beta > \epsilon(1-\beta)$, or $\epsilon < \beta$. Similarly, if U2 observes $O_0 = -\bar{S}_0 = S_{-1} \cdot (-1)^{Y_{-1}}$ (the $\epsilon$ branch), there is a chance of $\epsilon\beta$ where its observation is actually correct, and a chance of $(1-\epsilon)(1-\beta)$ where it is wrong. Hence, to maximize $\mathbb{P}(Y_0 = 1)$, U2 only follows the observation if $\epsilon\beta > (1-\epsilon)(1-\beta)$, or $\epsilon + \beta > 1$. Depending on whether U2 follows the observation when $O_t = \bar{S}_0 = S_{t-1} \cdot (-1)^{Y_{t-1}+1}$ and whether U2 follows the observation when $O_t = -\bar{S}_0 = S_{t-1} \cdot (-1)^{Y_{t-1}}$, a given model falls into one of the following four cases:

- $\epsilon < \beta$ and $\epsilon + \beta > 1$: U2 always follows the observations, $\mathbb{P}(Y_0 = 0) = \bar{\epsilon}\bar{\beta} + \epsilon\bar{\beta} = \bar{\beta}$;

- $\epsilon < \beta$ and $\epsilon + \beta < 1$: U2 always assumes the $\bar{\epsilon} = 1 - \epsilon$ branch ($S_t = S_{t-1} \cdot (-1)^{Y_{t-1}+1}$), $\mathbb{P}(Y_0 = 0) = \epsilon\bar{\beta} + \epsilon\beta = \epsilon$;

- $\epsilon > \beta$ and $\epsilon + \beta > 1$: U2 always assumes the $\epsilon$ branch ($S_t = S_{t-1} \cdot (-1)^{Y_{t-1}}$), $\mathbb{P}(Y_0 = 0) = \bar{\epsilon}\beta + \bar{\epsilon}\bar{\beta} = \bar{\epsilon}$;

- $\epsilon > \beta$ and $\epsilon + \beta < 1$: U2 always flips the observations, $\mathbb{P}(Y_0 = 0) = \bar{\epsilon}\beta + \epsilon\beta = \beta$.

Back to the decision of $U_0$, U1 compares the instantaneous rewards of the three policies $g^{\mathrm{i}}[0,0]$, $g^{\mathrm{i}}[0,1]$, and $g^{\mathrm{i}}[1,0]$, and chooses the highest. The instantaneous reward of $g^{\mathrm{i}}[0,1]$ is $1-\epsilon c$ (it communicates $\epsilon$ of the time), and the instantaneous reward of $g^{\mathrm{i}}[0,1]$ is $1-(1-\epsilon)c$. Also recall the instantaneous reward of $g^{\mathrm{i}}[0,0]$ is $\mathbb{P}(Y_0 = 1)$ under the policy. Let $U_0'$ be the indicator that one of $g^{\mathrm{i}}[0,1]$ and $g^{\mathrm{i}}[1,0]$ is optimal; that is, $U_0' = 1$ if U1 should convey the information of $S_0$ through a protocol, and $U_0' = 0$ if U1 should just leave U2 to guess $S_0$ with maximum likelihood estimation. Then the optimal choice of $U_0'$ is

$$\begin{aligned} U_0' &= \begin{cases} 1, & \text{if } \mathbb{P}(Y_0 = 1) < 1-\epsilon c \text{ or } \mathbb{P}(Y_0 = 1) < 1-(1-\epsilon)c, \\ 0, & \text{if } \mathbb{P}(Y_0 = 1) > 1-\epsilon c \text{ and } \mathbb{P}(Y_0 = 1) > 1-(1-\epsilon)c. \end{cases} \\ &= \begin{cases} 1, & \text{if } \mathbb{P}(Y_0 = 0) < \min\{\epsilon, 1-\epsilon\}c, \\ 0, & \text{if } \mathbb{P}(Y_0 = 0) > \min\{\epsilon, 1-\epsilon\}c. \end{cases} \end{aligned} \tag{D.24}$$

Now we change the indexing back to $t$. Combining (D.24) with the four cases listed above gives (5.29), and combining (5.29) with the definition of $g^{\mathrm{i}}[j,k]$ gives (5.28).

## D.7 Auxiliary Lemmas

**Lemma D.7** (Hoeffding-Azuma inequality)**:** *Let $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_n$ be a filtration, and $X_1, \ldots, X_n$ be real random variables such that $X_i$ is $\mathcal{F}_i$-measurable, $\mathbb{E}[X_i | \mathcal{F}_{i-1}] = 0$, $|X_i| \le b$ for some fixed $b \ge 0$. Furthermore, let $\{y_i\}_{i=1}^n$ be a fixed sequence. Then with probability at least $1-\delta$,*

$$\sum_{i=1}^n y_i X_i \le b \sqrt{2 \left( \sum_{i=1}^n y_i^2 \right) \log(1/\delta)}.$$

**Lemma D.8** ([126, 56])**:** *Let $\hat{p}(\cdot) \in \mathbb{R}_+^d$ be the empirical over $d$ distinct events from $n$ samples, and let $p(\cdot)$ be the true underlying distribution. Then with probability at least $1-\delta$,*

$$\|\hat{p}(\cdot) - p(\cdot)\|_1 \le \sqrt{\frac{2d \log(T/\delta)}{n}}.$$

**Lemma D.9** (Lemma 4.1 of [57])**:** *For a positive integer* $\tau$*, define* $\alpha_\tau^i = \alpha_i \Pi_{j=i+1}^{\tau}(1-\alpha_j)$ *for* $1 \le i \le \tau$ *and* $\alpha_\tau^0 = \Pi_{j=1}^{\tau}(1-\alpha_j)$ *where* $\alpha_\tau = \frac{H+1}{H+\tau}$*. Then the following hold:*

1. $\frac{1}{\sqrt{\tau}} \le \sum_{i=1}^{\tau} \frac{\alpha_\tau^i}{\sqrt{i}} \le \frac{2}{\sqrt{\tau}}$ *for all* $\tau \ge 1$*.*
2. $\max_{i \in [\tau]} \alpha_\tau^i \le \frac{2H}{\tau}$ *and* $\sum_{i=1}^{\tau} (\alpha_\tau^i)^2 \le \frac{2H}{\tau}$ *for all* $\tau \ge 1$*.*
3. $\sum_{\tau=i}^{\infty} \alpha_\tau^i = 1 + \frac{1}{H}$ *for all* $i \ge 1$*.*

**Lemma D.10:** *Let* $\alpha_n^i$ *be defined as in Lemma D.9. Let* $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_n$ *be a filtration, and* $p_1, \ldots, p_n$ *be distribution over* $\mathcal{S}$ *where* $p_i$ *is deterministic given* $\mathcal{F}_{i-1}$*. Suppose that* $s_i \in \mathcal{S}$ *is drawn from* $p_i$*. Then with probability at least* $1-\delta$*, for all* $V : \mathcal{S} \to [0,1]$*,*

$$\left| \sum_{i=1}^{n} \alpha_n^i V(s_i) - \sum_{i=1}^{n} \alpha_n^i \sum_{s} p_i(s) V(s) \right| \le 3\sqrt{\frac{SH}{n} \log(4n/\delta)}.$$

**Proof:** Consider the following $\epsilon$-cover for the space of $V$: $\mathcal{V} = \{V : \mathcal{S} \to \{0, \epsilon, 2\epsilon, \ldots, 1\}\}$ For every fixed $V \in \mathcal{V}$, we have with probability at least $1-\delta'$,

$$\left| \sum_{i=1}^{n} \alpha_n^i V(s_i) - \sum_{i=1}^{n} \alpha_n^i \sum_{s} p_i(s) V(s) \right| \le \sqrt{2\left(\sum_{i=1}^{n} (\alpha_n^i)^2\right) \log(2/\delta')} \le 2\sqrt{\frac{H}{n} \log(2/\delta')}.$$

By an union bound, the above holds for all $V \in \mathcal{V}$ with probability at least $1 - |\mathcal{V}|\delta'$.

For any $V : \mathcal{S} \to [0,1]$, there is a $\tilde{V} \in \mathcal{V}$ such that $|V(s) - \tilde{V}(s)| \le \frac{\epsilon}{2}$ for all $s$. Thus, with probability $1 - |\mathcal{V}|\delta'$, we have

$$\left| \sum_{i=1}^{n} \alpha_n^i V(s_i) - \sum_{i=1}^{n} \alpha_n^i \sum_{s} p_i(s) V(s) \right| \le 2\sqrt{\frac{H}{n} \log(2/\delta')} + \epsilon.$$

Picking $\epsilon = \frac{1}{n}$ (which implies $|\mathcal{V}| = \left(\frac{1}{\epsilon} + 1\right)^S \le \left(\frac{2}{\epsilon}\right)^S = (2n)^S$), and $\delta' = \frac{\delta}{|\mathcal{V}|}$, the right-hand side above can be upper bounded by

$$2\sqrt{\frac{H}{n} \log(2(2n)^S/\delta)} + \frac{1}{n} \le 3\sqrt{\frac{SH}{n} \log(4n/\delta)}.$$

■

# BIBLIOGRAPHY

# BIBLIOGRAPHY

[1] Y. Abbasi-Yadkori, P. L. Bartlett, and C. Szepesvári, “Online learning in Markov decision processes with adversarially chosen transition probability distributions,” *arXiv preprint arXiv:1303.3055*, 2013.

[2] R. Agrawal, “Sample mean based index policies by o(log n) regret for the multi-armed bandit problem,” *Advances in Applied Probability*, vol. 27, no. 4, pp. 1054–1078, 1995.

[3] R. Arora, T. V. Marinov, and M. Mohri, “Corralling stochastic bandit algorithms,” in *International Conference on Artificial Intelligence and Statistics.* PMLR, 2021, pp. 2116–2124.

[4] H. Attouch and D. Azé, “Approximation and regularization of arbitrary functions in Hilbert spaces by the Lasry-Lions method,” *Annales de l’IHP Analyse non linéaire.*, vol. 10, no. 3, pp. 289–312, 1993.

[5] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multi-armed bandit problem,” *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[6] R. J. Aumann, “Agreeing to disagree,” *The Annals of Statistics*, vol. 4, no. 6, pp. 1236–1239, 1976.

[7] M. G. Azar, I. Osband, and R. Munos, “Minimax regret bounds for reinforcement learning,” in *Proc. 34th International Conference on Machine Learning (ICML 2017)*, 2017.

[8] K. Azizzadenesheli, A. Lazaric, and A. Anandkumar, “Reinforcement learning in rich-observation MDPs using spectral methods,” arXiv preprint arXiv:1611.03907, 2018.

[9] Y. Bai, C. Jin, H. Wang, and C. Xiong, “Sample-efficient learning of Stackelberg equilibria in general-sum games,” arXiv preprint arXiv:2102.11494, 2021.

[10] N. Bard *et al.*, “The hanabi challenge: A new frontier for AI research,” *Artificial Intelligence*, vol. 280, 2020, 103216.

[11] D. Bauso, L. Giarré, and R. Pesenti, “Non-linear protocols for optimal distributed consensus in networks of dynamic agents,” *Systems and Control Letters*, vol. 55, no. 11, pp. 918–928, 2006.

[12] A. Beck and L. Tetruashvili, “On the convergence of block coordinate descent type methods,” *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[13] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, “Balancing communication and computation in distributed optimization,” *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3141–3155, 2018.

[14] A. S. Berahas, R. Bollapragada, and E. Wei, “On the convergence of nested decentralized gradient methods with multiple consensus and gradient steps,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 4192–4203, 2021.

[15] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, “The complexity of decentralized control of Markov decision processes,” *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.

[16] D. Bertsekas, “Multiagent rollout algorithms and reinforcement learning,” arXiv preprint arXiv:1910.00120, 2019.

[17] P. Bianchi and J. Jakubowicz, “Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization,” *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2013.

[18] I. Bistritz, T. Z. Baharav, A. Leshem, and N. Bambos, “One for all and all for one: Distributed learning of fair allocations with multi-player bandits,” *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 2, pp. 584–598, 2021.

[19] I. Bistritz and A. Leshem, “Game of thrones: Fully distributed learning for multiplayer bandits,” *Mathematics of Operations Research*, vol. 46, no. 1, pp. 159–178, 2021.

[20] V. S. Borkar, “Stochastic approximation with two time scales,” *Systems and Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.

[21] ——, *Stochastic approximation: a dynamical systems viewpoint.* New Delhi, India, and Cambridge, UK: Hindustan Book Agency, and Cambridge University Press, 2008.

[22] J. Borwein and Q. Zhu, *Techniques of Variational Analysis.* New York: Springer-Verlag, 2005.

[23] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing Markov chain on a graph,” *SIAM Review*, vol. 46, no. 4, pp. 667–689, 2004.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, UK: Cambridge University Press, 2004.

[26] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019.

[27] S. Bubeck, T. Budzinski, and M. Sellke, "Cooperative and stochastic multi-player multi-armed bandit: Optimal regret with neither communication nor collisions," in *Conference on Learning Theory.* PMLR, 2021, pp. 821–822.

[28] M. E. Chamie, G. Neglia, and K. Avrachenkov, "Distributed weight selection in consensus protocols by schatten norm minimization," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1350–1355, 2015.

[29] S. Chandak, V. S. Borkar, and P. Dodhia, "Concentration of contractive stochastic approximation and reinforcement learning," arXiv preprint arXiv:2106.14308, 2021.

[30] W. Chang, M. Jafarnia-Jahromi, and R. Jain, "Online learning for cooperative multi-player multi-armed bandits," arXiv preprint arXiv:2109.03818, 2021.

[31] R. Chawla, A. Sankararaman, A. Ganesh, and S. Shakkottai, "The gossiping insert-eliminate algorithm for multi-agent bandits," in *International Conference on Artificial Intelligence and Statistic*, 2020.

[32] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, vol. 155, no. 1-2, pp. 57–79, 2016.

[33] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent system," in *AAAI Conference on Artificial Intelligence*, 1998.

[34] A. d'Aspremont, D. Scieur, and A. Taylor, "Acceleration methods," arXiv preprint arXiv:2101.09545, 2021.

[35] C. A. S. de Witt, J. N. Foerster, G. Farquhar, P. H. S. Torr, W. Böhmer, and S. Whiteson, "Multi-agent common knowledge reinforcement learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[36] F. Derakhshan and S. Yousefi, "A review on the applications of multiagent systems in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, 2019.

[37] J. Dibangoye and O. Buffet, "Learning to act in decentralized partially observable MDPs," in *International Conference on Machine Learning*, 2018.

[38] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[39] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[40] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1874–1889, 2015.

[41] J. Farrell and M. Rabin, "Cheap talk," *Journal of Economic Perspectives*, vol. 10, no. 3, pp. 103–118, 1996.

[42] J. N. Foerster, H. F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling, "Bayesian action decoder for deep multi-agent reinforcement learning," in *Proc. 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[43] H.-O. Georgii, *Gibbs Measures and Phase Transitions.* Berlin, Germany: de Gruyter, 2011.

[44] J. C. Geromel, J. Bernussou, and P. L. D. Peres, "Decentralized control through parameter space optimization," *Automatica*, vol. 30, no. 10, pp. 1565–1578, 1994.

[45] M. Ghorbanzadeh, A. Abdelhadi, and C. Clancy, "Distributed resource allocation," in *Cellular Communications Systems in Congested Environments.* Springer, 2017, pp. 61–91.

[46] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics," *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.

[47] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," arXiv preprint arXiv:1809.01999, 2019.

[48] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI Conference on Artificial Intelligence*, 2004.

[49] N. Hidaka, T. Miyamoto, and H. Sugihara, "An ADMM approach to distributed optimization in distribution system considering fairness among consumers," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.

[50] Y.-C. Ho, "Team decision theory and information structures," *Proceedings of the IEEE*, vol. 68, no. 6, pp. 644–654, 1980.

[51] J. Huang, V. Subramanian, R. Agrawal, and R. Berry, "Downlink scheduling and resource allocation for OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 288–296, 2009.

[52] J. Huang, V. Subramanian, R. Berry, and R. Agrawal, "Scheduling and resource allocation in OFDMA wireless systems," in *Orthogonal Frequency Division Multiple Access Fundamentals and Applications.* Boca Raton: Auerbach Publications, 2009, ch. 6, pp. 131–163.

[53] D. Huo, "Clarification on the wrap-around hexagon network structure," in *IEEE Technical Report C802.20-05/15*, 2005.

[54] M. Jafarnia-Jahromi, R. Jain, and A. Nayyar, "Online learning for unknown partially observable MDPs," arXiv preprint arXiv:2102.12661, 2021.

[55] D. Jakovetić, J. Xavier, and J. M. F. Moura, "Convergence rate analysis of distributed gradient methods for smooth optimization," in *20th Telecomm Forum*, 2012.

[56] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning." *Journal of Machine Learning Research*, vol. 11, no. 4, 2010.

[57] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient," arXiv preprint arXiv:1807.03765, 2018.

[58] T. B. K. Zhang, Z. Yang, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," arXiv preprint arXiv:1911.10635, 2019.

[59] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[60] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *In Proc. 19th International Conference on Machine Learning.* Citeseer, 2002.

[61] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multiplayer multiarmed bandits," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2331–2345, 2014.

[62] H. Kao, C.-Y. Wei, and V. Subramanian, "Decentralized cooperative reinforcement learning with hierarchical information structure," arXiv preprint arXiv:2111.00781, 2021.

[63] A. D. Kara and S. Yuksel, "Near optimality of finite memory feedback policies in partially observed Markov decision processes," arXiv preprint arXiv:2010.07452, 2020.

[64] D. Koller and B. Milch, "Multi-agent influence diagrams for representing and solving games," *Games and Economic Behavior*, vol. 45, no. 1, pp. 181–221, 2003.

[65] M. Kozlov, S. Tarasov, and L. Khachiyan, "The polynomial solvability of convex quadratic programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 5, pp. 223–228, 1980.

[66] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control.* SIAM, 2015.

[67] A. Lalitha and A. Goldsmith, "Bayesian algorithms for decentralized stochastic bandits," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 2, pp. 564–583, 2021.

[68] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision making in multi-agent multi-armed bandits," arXiv preprint arXiv:2003.01312, 2020.

[69] J.-M. Lasry and P.-L. Lions, "A remark on regularization in Hilbert spaces," *Israel J. Math.*, vol. 55, no. 3, pp. 257–266, 1986.

[70] J. M. Lee, *Introduction to Riemannian manifolds.* Springer, Cham, 2018.

[71] B. Lemmens and R. Nussbaum, *Nonlinear Perron–Frobenius theory.* Cambridge, UK: Cambridge University Press, 2012.

[72] S. Leonardos, W. Overman, I. Panageas, and G. Piliouras, "Global convergence of multi-agent policy gradient in Markov potential games," *arXiv preprint arXiv:2106.01969*, 2021.

[73] A. Lerer, H. Hu, J. Foerster, and N. Brown, "Improving policies via search in cooperative partially observable games," arXiv preprint arXiv:1912.02318, 2019.

[74] B. Li, S. Cen, Y. Chen, and Y. Chi, "Communication-efficient distributed optimization in networks with gradient tracking and variance reduction," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[75] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[76] X. Li and A. Scaglione, "Convergence and applications of a gossip based gauss Newton algorithm," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5231–5246, 2013.

[77] X. Lin and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[78] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th International Conference on Machine Learning (ICML 1994)*, 1994.

[79] F. Liu, J. Lee, and N. Shroff, “A change-detection based framework for piecewise-stationary multi-armed bandit problem,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[80] P. D. Lorenzo and G. Scutari, “Next: In-network nonconvex optimization,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[81] U. Madhushani and N. E. Leonard, “A dynamic observation strategy for multi-agent multi-armed bandit problem,” in *2020 European Control Conference (ECC)*, 2020.

[82] A. Mahajan and M. Mannan, “Decentralized stochastic control,” *Annals of Operations Research*, vol. 241, no. 1, pp. 109–126, 2016.

[83] F. Mansoori and E. Wei, “Superlinearly convergent asynchronous distributed network Newton method,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.

[84] W. Mao, K. Zhang, E. Miehling, and T. Başar, “Information state embedding in partially observable cooperative multi-agent reinforcement learning,” in *2020 IEEE 59th Annual Conference on Decision and Control (CDC)*, 2020.

[85] A. S. Mathkar and V. S. Borkar, “Nonlinear gossip,” *SIAM Journal on Control and Optimization*, vol. 54, no. 3, pp. 1535–1557, 2016.

[86] P. Naghizadeh, M. Gorlatova, A. S. Lan, and M. Chiang, “Hurts to be too early: Benefits and drawbacks of communication in multi-agent learning,” in *2019 IEEE INFOCOM*, 2019.

[87] A. Nayyar, A. Mahajan, and D. Teneketzis, “Optimal control strategies in delayed sharing information structures,” *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1606–1620, 2011.

[88] ——, “Decentralized stochastic control with partial history sharing: A common information approach,” *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1644–1658, 2013.

[89] ——, “The common-information approach to decentralized stochastic control,” in *Information and Control in Networks.* Springer, 2014, pp. 123–156.

[90] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, “On distributed averaging algorithms and quantization effects,” *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, 2009.

[91] A. Nedić and A. Ozdaglar, “Distributed subgradient methods for multiagent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[92] A. Nedić and V. Subramanian, "Approximately optimal utility maximization," in *2009 IEEE Information Theory Workshop on Networking and Information Theory*, Volos, Greece, 2009.

[93] B. Ning, G. Sun, J. Li, A. Zhang, W. Hao, and S. Yang, "Resource allocation in multi-user cognitive radio network with Stackelberg game," *IEEE Access*, 2020.

[94] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed optimization for smart cyber-physical networks," arXiv preprint arXiv:1906.10760, 2019.

[95] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs.* Springer, 2016.

[96] A. Olshevsky and J. N. Tsitsiklis, "Convergence rates in distributed consensus and averaging," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.

[97] Y. Ouyang, H. Tavafoghi, and D. Teneketzis, "Dynamic games with asymmetric information: Common information based perfect bayesian equilibria and sequential decomposition," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 222–237, 2017.

[98] Y. Ouyang and D. Teneketzis, "A common information-based multiple access protocol achieving full throughput," in *2015 IEEE International Symposium on Information Theory*, 2015.

[99] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.

[100] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[101] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *Journal of AI Research*, vol. 16, pp. 389–423, 2002.

[102] Z. Rabinovich, C. V. Goldman, and J. S. Rosenschein, "The complexity of multiagent systems: The price of silence," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003.

[103] G. Radanovic, R. Devidze, D. Parkes, and A. Singla, "Learning to collaborate in Markov decision processes," in *International Conference on Machine Learning.* PMLR, 2019, pp. 5261–5270.

[104] R. Radner, "Team decision problems," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 857–881, 1962.

[105] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[106] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 514–529, 2018.

[107] R. T. Rockafellar and R. Wets, *Variational Analysis.* Berlin, Germany: Springer-Verlag, 1998.

[108] A. Sankararaman, A. Ganesh, and S. Shakkottai, "Social learning in multi agent multi armed bandits," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 3, pp. 1–35, 2019.

[109] S. M. Shah and V. S. Borkar, "Distributed stochastic approximation with local projections," *SIAM Journal on Optimization*, vol. 28, no. 4, pp. 3375–3401, 2018.

[110] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Multi-armed bandits in multi-agent networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[111] S. Sokota, E. Lockhart, F. Timbers, E. Davoodi, R. D'Orazio, N. Burch, M. Schmid, M. Bowling, and M. Lanctot, "Solving common-payoff games with approximate policy iteration," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[112] J. Subramanian and A. Mahajan, "Approximate information state for partially observed systems," in *2019 IEEE 58th Annual Conference on Decision and Control (CDC)*, 2019.

[113] J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan, "Approximate information state for approximate planning and reinforcement learning in partially observed systems," arXiv preprint arXiv:2010.08843, 2020.

[114] Y. Sun, A. Daneshmand, and G. Scutari, "Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation," arXiv preprint arXiv:1905.02637, 2019.

[115] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge: MIT Press, 2018.

[116] D. Tang, H. Tavafoghi, V. Subramanian, A. Nayyar, and D. Teneketzis, "Dynamic games among teams with delayed intra-team information sharing," arXiv preprint arXiv:2102.11920, 2019.

[117] H. Tavafoghi, Y. Ouyang, and D. Teneketzis, “A sufficient information approach to decentralized decision making,” in *2018 IEEE 57th Annual Conference on Decision and Control (CDC)*, 2018.

[118] ——, “A unified approach to dynamic decision problems with asymmetric information-part ii: Strategic agents,” arXiv preprint arXiv:1812.01132, 2018.

[119] ——, “A unified approach to dynamic decision problems with asymmetric information: Non-strategic agents,” *IEEE Transactions on Automatic Control*, 2021.

[120] Y. Tian, Y. Wang, T. Yu, and S. Sra, “Online learning in unknown Markov games,” in *International Conference on Machine Learning.* PMLR, 2021, pp. 10 279–10 288.

[121] V. M. Tikhomirov, “On the notion of mean,” in *Selected Works of AN Kolmogorov.* Springer, 1991, pp. 144–146.

[122] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Controll*, vol. 31, no. 9, pp. 803–812, 1986.

[123] D. Vial, S. Shakkottai, and R. Srikant, “Robust multi-agent multi-armed bandits,” in *Proceedings of ACM Mobihoc*, 2021.

[124] J. Wang and G. Joshi, “Cooperative SGD: A unified framework for the design and analysis of communication-efficient sgd algorithms,” arXiv preprint arXiv:1808.07576, 2018.

[125] ——, “Adaptive communication strategies to achieve the best error-runtime trade-off in local-update sgd,” in *Systems and Machine Learning (SysML) Conference*, 2019.

[126] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, “Inequalities for the L1 deviation of the empirical distribution,” *Hewlett-Packard Labs, Tech. Rep*, 2003.

[127] H. S. Witsenhausen, “A standard form for sequential stochastic control,” *Mathematical Systems Theory*, vol. 7, no. 1, pp. 5–11, 1973.

[128] ——, “A standard form for sequential stochastic control,” *Mathematical Systems Theory*, vol. 7, no. 1, pp. 5–11, 1973.

[129] J. Wolfowitz, “Products of indecomposable, aperiodic, stochastic matrices,” *Proceedings of the American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963.

[130] Y. Xiao and J. Hu, “Distributed solutions of convex feasibility problems with sparsely coupled constraints,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.

[131] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

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

[133] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine, “Learning invariant representations for reinforcement learning without reconstruction,” arXiv preprint arXiv:2006.10742, 2021.

[134] J. Zhang, C. A. Uribe, A. Mokhtari, and A. Jadbabaie, “Achieving acceleration in distributed optimization via direct discretization of the heavy-ball ode,” in *2019 American Control Conference (ACC)*, 2019.

[135] K. Zhang, Z. Yang, and T. Başar, “Decentralized multi-agent reinforcement learning with networked agents: Recent advances,” arXiv preprint arXiv:1912.03821, 2019.

[136] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, “Fully decentralized multi-agent reinforcement learning with networked agents,” in *Proc. 35th International Conference on Machine Learning (ICML 2018)*, 2018.

[137] Z. Zhang, Y. Zhou, and X. Ji, “Almost optimal model-free reinforcement learning via reference-advantage decomposition,” arXiv preprint arXiv:2004.10019, 2020.

[138] M. Zhu and S. Martinez, “On the convergence time of asynchronous distributed quantized averaging algorithms,” *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 386–390, 2010.

[139] G. Zoutendijk, *Methods of feasible directions: a study in linear and non-linear programming.* New York: Elsevier, 1960.