# Essays on Dynamic Optimization for Markets and Networks

Yuanling Gan

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2023

# Abstract

Essays on Dynamic Optimization for Markets and Networks

Yuanling Gan

We study dynamic decision-making problems in networks and markets under uncertainty about future payoffs. This problem is difficult in general since 1) Although the current decision (potentially) affects future decisions, the decision-maker does not have exact information on the future payoffs when he/she commits to the current decision; 2) The decision made at one part of the network usually interacts with the decisions made at the other parts of the network, which makes the computation scales very fast with the network size and brings computational challenges in practice. In this thesis, we propose computationally efficient methods to solve dynamic optimization problems on markets and networks, specify a general set of conditions under which the proposed methods give theoretical guarantees on global near-optimality, and further provide numerical studies to verify the performance empirically. The proposed methods/algorithms have a general theme as "local algorithms", meaning that the decision at each node/agent on the network uses only partial information on the network.

In the first part of this thesis, we consider a network model with stochastic uncertainty about future payoffs. The network has a bounded degree, and each node takes a discrete decision at each period, leading to a per-period payoff which is a sum of three parts: node rewards for individual node decisions, temporal interactions between individual node decisions from the current and previous periods, and spatial interactions between decisions from pairs of neighboring nodes. The objective is to maximize the expected total payoffs over a finite horizon. We study a natural

*decentralized* algorithm (whose computational requirement is linear in the network size and planning horizon) and prove that our decentralized algorithm achieves *global* near-optimality when temporal and spatial interactions are not dominant compared to the randomness in node rewards. Decentralized algorithms are parameterized by the locality parameter $L$: An $L$-local algorithm makes its decision at each node $v$ based on current and (simulated) future payoffs only up to $L$ periods ahead, and only in an $L$-radius neighborhood around $v$. Given any permitted error $\epsilon > 0$, we show that our proposed $L$-local algorithm with $L = O(\log(1/\epsilon))$ has an average per-node-per-period optimality gap bounded above by $\epsilon$, in networks where temporal and spatial interactions are not dominant. This constitutes the first theoretical result establishing the global near-optimality of a local algorithm for network dynamic optimization.

In the second part of this thesis, we consider the previous three types of payoff functions under adversarial uncertainty about the future. In general, there are no performance guarantees for arbitrary payoff functions. We consider an additional convexity structure in the individual node payoffs and interaction functions, which helps us leverage the tools in the broad Online Convex Optimization literature. In this work, we study the setting where there is a trade-off between developing future predictions for a longer lookahead horizon, denoted as $k$ versus increasing spatial radius for decentralized computation, denoted as $r$. When deciding individual node decisions at each time, each node has access to predictions of local cost functions for the next $k$ time steps in an $r$-hop neighborhood. Our work proposes a novel online algorithm, Localized Predictive Control (LPC), which generalizes predictive control to multi-agent systems. We show that LPC achieves a competitive ratio of $1 + \widetilde{O}(\rho_T^k) + \widetilde{O}(\rho_S^r)$ in an adversarial setting, where $\rho_T$ and $\rho_S$ are constants in $(0, 1)$ that increase with the relative strength of temporal and spatial interaction costs, respectively. This is the first competitive ratio bound on decentralized predictive control for networked online convex optimization. Further, we show that the dependence on $k$ and $r$ in our results is near-optimal by lower bounding the competitive ratio of any decentralized online algorithm.

In the third part of this work, we consider a general dynamic matching model for online competitive gaming platforms. Players arrive stochastically with a skill attribute, the Elo rating. The

distribution of Elo is known and i.i.d across players. However, the individual's rating is only observed upon arrival. Matching two players with different skills incurs a match cost. The goal is to minimize a weighted combination of waiting costs and matching costs in the system. We investigate a popular heuristic used in industry to trade-off between these two costs, the *Bubble* algorithm. The algorithm places arriving players on the Elo line with a growing bubble around them. When two bubbles touch, the two players get matched. We show that, with the optimal bubble expansion rate, the Bubble algorithm achieves a constant factor ratio against the offline optimal cost when the match cost (resp. waiting cost) is a power of Elo difference (resp. waiting time). We use players' activity logs data from a gaming start-up to validate our approach and further provide guidance on how to tune the Bubble expansion rate in practice.

# Table of Contents

# List of Figures

# List of Tables

## Acknowledgements

This thesis would not have been possible without the support of many people.
First and foremost, I would like to express my gratitude to my advisor, Prof. Kanoria for his invaluable guidance and support throughout the years. He is brilliant and captures the essence of the most technically difficult results. He taught me how to think about math problems with the right intuition and position the research work in a broader context. Besides the academic mentorship, Yash showed me great support and compassion when I went through several difficult periods during my Ph.D. study. I appreciate those Zoom sessions organized by him as virtual office spaces during the Pandemic to keep me connected with my advisor and other Ph.D. students. I also appreciate his encouragement and help when I encounter setbacks during the job-hunting process in this difficult and strange year. I would also like to thank Prof. Will Ma for his guidance and support. I am continuously amazed by his remarkable productivity and success in research work, teaching, poker competitions, gaming startups, and many other professional services. I truly enjoyed his class on Revenue and Supply Chain Management and learned many clever online algorithms. I am grateful for our research conversations and discussions.
I extend my sincere appreciation to my committee members, Prof. Omar Besbes, Prof. Kostas Bimpikis, and Prof. Hongseok Namkoong, for their invaluable suggestions and feedback. I also owe a debt of gratitude to other faculty and staff members at Columbia University, including but not limited to Professor Santiago Balseiro, Carri Chan, Jing Dong, Awi Federgruen, Assaf Zeevi, Daniel Russo, Hongyao Ma, Fanyin Zheng, Elizabeth Elam, Kira Grant, and Winnie Leung. I would also like to thank fellow collaborators Xuan Zhang and Yiheng Lin for their constant

sources of inspiration in both life and research. The past five years have been a journey filled with ups and downs. I am thankful to many friends inside and outside Columbia: Xiaotian Xie, Peijun Xiao, Lei Xiao, Xuan Zhang, Yue Hu, Shangzhou Xia, Mike Li, Muye Wang, Anand Kalvit, Kumar Goutam, Jiaqi Lu, Pengyu Qian, Yun Wen, Wenxin Zhang, Zhen Huang, Yunbei Xu and Yilun Chen who offered their help, believed in me during my moments of doubt and encouraged me through their kindness.

Lastly, I express my heartfelt gratitude to my parents for their unwavering love, support, and unbiased advice. Those Facetime calls with Mom and Dad every evening are the most relaxing and beautiful time of my day. I also extend my appreciation to my boyfriend, Josh Wu, for being my constant companion through thick and thin.

# Chapter 1: Introduction

Dynamic decision-making problems have evolved and become more crucial than ever in the past twenty years due to the amount of available data. Dynamic policies are no longer just a concept but are studied and implemented in many settings, such as dynamic pricing for e-commerce companies, dynamic dispatching for ride-hailing platforms, dynamic matching for online gaming platforms, and many more. A dynamic policy has advantages over static optimization in the sense that this framework incorporates new information as time passes. But a poor design of dynamic policies may bring less value to the table or even perform worse than a calibrated static policy due to the following (and potentially other) reasons:

- A dynamic policy requires much more computation resources compared to a static policy and is even infeasible to implement in practice.
- A dynamic policy oscillates a lot between each decision epoch and hurts the overall performance. A well-known example from (Hazan, 2021) shows a naive dynamic Follow The Leader algorithm can perform much worse compared to the best static policy due to oscillation.
- A dynamic policy is sensitive to future predictions and hard to quantify the performance guarantees when prediction quality varies.

In this thesis, we study dynamic decision-making problems in networks and markets where the objective is to collectively maximize the total payoff across the entire network over a finite horizon. We propose methods and algorithms that avoid the potential drawbacks of dynamic policies and quantify the policy performance theoretically as well as numerically.

## 1.1 Dynamic decision-making on networks under stochastic uncertainty

In the Chapter 2, we start with a stochastic framework to model the uncertainty about future payoffs. The overall objective is an unweighted sum over a finite time horizon with potentially nonstationary per-period payoffs. At each decision epoch, the decision-maker is aware of his/her past decisions, and the current payoff function, but not the future payoff realizations. We assume the future payoff distributions are known. To make the problem more general and interesting, we are not assuming an analytical form of the distributions but only using simulated samples in the decision process. Our goal is to identify under what conditions, local algorithms (which are computationally efficient for large networks) are globally near-optimal.

This setup relates to Markov Decision Problem (MDP) in networks. In the Operations Research field, the most classical books are (Bertsekas, 2011; Bertsekas, 2012) where an MDP is defined by a tuple of the following four elements: state space, action space, transition probabilities, and immediate payoff. For the canonical MDP problem, running the Dynamic Programming algorithm outputs the global optimal solution; however, our setup includes problems with arbitrarily long horizons and large network sizes (which give rise to a very large state space). This makes the Dynamic Programming algorithm suffer the *curse of dimensionality* and infeasible for practical uses. In the Approximated Dynamic Programming literature, the Rollout algorithm is a popular heuristic that essentially truncates the horizon up to some time steps. However, this heuristic still does not handle the complexity coming from the large network size.

In the Chapter 2, we use a similar idea as Rollout but extend it to a network setting. Specifically, we consider "local algorithms" in both temporal and spatial dimensions. We characterize sufficient conditions on the per-period payoff functions such that the temporal and spatial radius needed for a global near-optimal solution (compared to the Bellman optimal) is "small". Note that our model does have a simpler structure than a typical MDP where we assume the transitions are deterministic and the previous action is the current state for each node on the network. The stochasticity comes from the random payoff functions, not the transitions. Our work may serve as a starting point

for developing a theoretical understanding of sufficient conditions for local algorithms achieving global near-optimality in general network Markov Decision Problem settings.

### 1.2 Dynamic decision-making on networks under adversarial uncertainty

In the Chapter 3, we continue our exploration of local algorithms in an adversarial framework, where the future payoffs are unknown and arbitrary. The overall objective is to minimize an unweighted sum over a finite time horizon. In both stochastic and adversarial models, the per-period payoffs consist of three parts: node costs/rewards, temporal interactions, and spatial interactions. In the adversarial framework, we further assume that the node costs are strongly convex, and the interaction functions are convex. Our goal is to identify under what conditions, local algorithms achieve near-optimal competitive ratios compared to the offline optimal benchmark. Moreover, we explore the trade-off between investing more resources in developing predictions in a longer horizon versus increasing (spatial) locality radius for decentralized computation.

This setup relates to Smoothed Online Convex Optimization (SOCO) problem which is first introduced by Lin et al. (2012) to model dynamic power management in data centers. In contrast to classical Online Convex Optimization, SOCO adds temporal interaction cost functions in the model which further encourage nodes/agents to choose an action that is "compatible" with their previous actions. For example, a temporal interaction could be a switching cost that penalizes large deviations from the previous action, to make the trajectory of local actions "smooth". However, the SOCO model is not specific to network settings and hence the theoretical results do not incorporate network structures.

In the Chapter 3, we generalize Model Predictive Control (MPC) algorithm in the SOCO literature to the network setting. We named the corresponding algorithm as Local Predictive Control (LPC). We develop a general method of perturbation analysis to quantify the sensitivity of LPC solutions in terms of future predictions as well as node decisions at other parts of the network. We conclude this section by establishing a near-optimal competitive ratio guarantee for LPC and resource augmentation bounds in terms of trading off between the value of predictions and (spatial)

computation radius.

### 1.3 Dynamic match-making on online competitive gaming platforms

In Chapter 4, we consider a practical application of dynamic match-making for online gaming platforms. We start with a popular heuristic (i.e., Bubble algorithm) in the gaming industry and develop a more theoretical understanding of the heuristic. The algorithm places arriving players on a scale of their skill attributes and then expands a bubble around each player. When two bubbles touch, the two players get matched. Specifically, we would like to understand under what model objective, this heuristic is near-optimal. Moreover, can the corresponding theoretical model guide the practitioners to adjust the parameters in the bubble algorithm?

In this work, we are mostly interested in gaming startups where the market is relatively thinner compared to the more established gaming companies. As a result, the waiting times are not negligible. We propose a general model which focuses on the foundational trade-off in balancing between waiting times and matching qualities. Specifically, we consider an analytical form of power laws where the waiting cost is the waiting time to some power and the matching cost is the players' skill differences to some other power. In this model, we show that the Bubble algorithm achieves a constant factor competitive ratio compared to the offline optimal benchmark.

In the empirical study of this work, we estimate the aforementioned powers from the players' activity logs dataset and provide insights on tuning the parameter in the Bubble algorithm to achieve improvements in the players' engagement metric.

# Chapter 2: Near-optimality of local algorithms on networks under stochastic uncertainty

## 2.1 Introduction

Many real-world contexts call for dynamic decision-making in networks with uncertainty about the future: At each period, a decision is made at each node in the network and a central planner aims to maximize the total payoff across the network. Examples of such settings include influence maximization in social networks (Tong et al., 2017; Banerjee, Jenamani, and Pratihar, 2020), multi-product pricing on product networks (Candogan, Bimpikis, and Ozdaglar, 2012; Caro and Gallien, 2012), and logistics planning on transportation networks (Devari, Nikolaev, and He, 2017; Fatehi and Wagner, 2021). In all these settings, the resulting payoffs arise from individual decisions at nodes, interactions among neighbors on the network, and temporal interactions between consecutive decisions made at each node.

We outline an illustrative example from a version of the multiproduct pricing problem (Talluri and Ryzin, 2006; Gallego and Topaloglu, 2019). Consider a retailer who dynamically decides which products (or product groups) to put on sale at different times. Here, nodes in the network correspond to products and the edges connect related (e.g., substitutable or complementary) products. The payoff collected in each "period" (e.g, a week) includes several components: "Node rewards" correspond to the revenue from each product at the given price, "edge interactions" capture the change in product revenue if a related product (i.e., substitute or complement) is on discount, and "temporal interactions" capture how the trajectory of past discounts on the product affect the revenue from the product in that period. Future node rewards are uncertain because, e.g., the customer demand for a product in the future is uncertain. The goal is to maximize the expected rewards over a finite time horizon.

For another practical example, consider dynamic content recommendation problem (Yin et al., 2015; Ullah and Lee, 2016) on a social platform such as Facebook, Instagram, and Twitter. Here, the decision network is the social network, with nodes being the users and edges being their connections. The payoff which accrues from recommendations again consists of several components. First, there are "node rewards" which capture how interesting the post is to a given user. Second, there are "edge interactions": if the platform recommends the same post to a pair (or a group) of friends around the same time, this produces an additional reward since shared experiences with friends increase users' engagement. Third, there are "temporal interactions": e.g., repeating similar content decreases users' engagement and hence it is costly to show the post again if it has already been shown to a user. The goal of the platform is to maximize the expected rewards over a certain time horizon. Notably, there is uncertainty about future rewards – for instance, a user might become inactive.

Besides the goal of maximizing payoffs in such contexts, it is desirable to have decision rules that are "simple" in various ways such as computational efficiency, the potential to be computed in a distributed manner, interpretability, and robustness to model misspecification. In a networked optimization setting, an attractive class of algorithms is *decentralized* algorithms which obtain the decisions of individual nodes based solely on information from the "nearby" part of the network (Suomela, 2013). Motivated by the ubiquitousness of dynamic decision-making on networks and the practicality of decentralized algorithms, we aim to answer the following research question:

*Can decentralized algorithms be globally near-optimal in terms of maximizing collective rewards on networks under stochastic uncertainty about the future?*

In this paper, we study a benchmark model of dynamic decision-making on bounded degree graphs with stochastic uncertainty about the future. A decision must be taken at each node, in each period. The global payoff consists of 1) per-period *individual* node rewards, which are random functions over individual decisions; 2) per-period *spatial* interaction payoffs between neighboring nodes, which are (possibly random) functions of pairs of decisions; and 3) *temporal* interaction payoffs for individual node between consecutive periods, which are (possibly random) functions

of pairs of decisions. Importantly, we model the future payoff functions as stochastic, with known distributions. The goal is to maximize the expected reward collected over a finite horizon.

### 2.1.1 Contribution

We establish a first *theoretical* result establishing *global near-optimality* of a local algorithm for network dynamic optimization problems under stochastic uncertainty when temporal and spatial interactions are not too strong relative to the randomness in node rewards and the graph degree. Notably, our decentralized algorithm has a computational requirement *linear* in the network size and the time horizon. In each period, for each node $v$ (the "focal" node), the decision to be taken at $v$ is chosen as follows: Our algorithm simulates future reward realizations up to $L$ periods ahead, for an $L$-radius local neighborhood of $v$, where $L$ denotes the locality parameter determined based on how closely we want to approximate the global optimum. Crucially, $L$ is *independent of the network size and time horizon*. Our algorithm then solves the network optimization problem on the local subgraph and extracts from the solution only the decision at the focal node $v$.

Our work extends the literature on *correlation decay* for static (i.e., single-period) optimization in networks (Gamarnik, Goldberg, and Weber, 2014; Kerimov, 2014) which studies how the effects of decisions at the boundary of a graph propagate towards the focal node. In the previous static optimization setting, the total reward functions consist of node rewards and pairwise spatial interactions. However, when generalizing to the dynamic setting, the value-to-go functions further contain interactions between groups of nodes that are not direct neighbors of each other. Due to such *interaction-at-a-distance*, previous technical machinery developed to establish correlation decay in static networks does not generalize to our dynamic setting. We thus develop a novel machinery to establish correlation decay in dynamic decision-making settings (outlined in 2.4), which handles the interaction-at-a-distance phenomenon.

We now summarize the approach taken in our new analysis for establishing the existence of correlation decay and the global near-optimality of local algorithms. For any node at a given time, we construct a sequence of local (dynamic) optimization problems on the subgraph around this

focal node with increasing locality radius starting from $L$ and then compare the optimal decisions at the focal node (pathwise for the realized reward functions) in two local (dynamic) optimization problems: one with locality radius $H$ and the other with radius $H+1$ for any $H \geq L$. Specifically, we ask: in the problem with radius $H+1$, how large is the optimality gap caused by forcing the focal node decision to be the optimal decision for the problem with radius $H$? We show, via a delicate analysis summarized in the next paragraph, that this optimality gap is exponentially small in $H$. Summing over $H = L, L+1, \ldots, \infty$ yields a bound on the optimality gap caused by using an $L$-local algorithm.

The aforementioned optimality gap from constraining the focal node decision is the sum of the difference in the same-period payoff and the difference in the value-to-go functions. To overcome the issue that the value-to-go functions involve interaction among groups of nodes, we construct a spatial-temporal (ST) graph by making a clone of the original static network for each decision period and connecting consecutive-period clones of each node via temporal interaction edges (see 2.1.3 for the definition of ST graphs). Note that each node in the ST graph corresponds to an individual decision (at a particular node and period). On the ST graph, the total reward can be decomposed to a sum of simple terms, one for each node and one for each edge, eliminating group interactions-at-a-distance. Importantly, uncertainty about the future continues to pose a challenge in the ST graph re-formulation: the future reward functions are unknown at time $t$ while the decision-maker has to commit to a decision at time $t$. Due to this additional challenge in the temporal dimension, our optimization problem on the ST graph is still not the same as a static optimization problem on the ST graph. We work with the dynamic programming framework and prove that the difference in the value-to-go functions (from constraining a single node's decision policy) is upper bounded by the probability of having different node decisions at this single node in the next period under two corresponding local optimization problems, multiplied by a constant proportional to the temporal interaction strength. This yields a "contraction" argument via coupling the future decisions up to $H$ periods and then applying induction from "boundary" nodes of the $H$-neighborhood in the ST graph (see Lemmas 2.4.1 and 2.4.2). Note that using the contraction

argument to show the convergence to the global optimal solution is an important technique in iterative methods for nonlinear problems (Bertsekas and Tsitsiklis, 1989b). However, the key difficulty in employing it is to construct an appropriate action space and associated norm such that there indeed exists such a contraction mapping. In our work, we establish such a contraction mapping for the random decision vectors as summarized above.

To demonstrate our results numerically, we conduct a synthetic experiment on random regular graphs in 2.6. The experiment shows that when temporal and edge interactions don't dominate compared to the randomness in the node rewards and graph degree, our proposed decentralized algorithm achieves a near-optimal payoff with a small local radius $L$.

### 2.1.2 Related work

Our paper contributes to the following related research areas: (1) Correlation decay for decentralized algorithms; (2) Dynamic optimization in networks and (3) Multi-agent reinforcement learning.

*Correlation decay for decentralized algorithms.* Correlation decay is the cornerstone for the success of numerous decentralized algorithms for *static* network optimization in the literature. It is first developed in the domain of statistical physics. The seminal work of Dobrushin (1970) studies graphical models (e.g., a Markov chain is a graphical model on a line graph) on infinite graphs via correlation decay methods, investigating whether the joint distribution – the Gibbs measure – is uniquely determined by the distribution of each random variable conditional on its neighbors. Since then, the concept of correlation decay has expanded to applications beyond statistical physics (Weitz, 2006; Chen, Liu, and Vigoda, 2020; Ding, Sly, and Sun, 2015; Montanari, 2019), including wireless communication (Bandyopadhyay and Gamarnik, 2008; Gamarnik and Katz, 2009; Weitz, 2006), combinatorial optimization (Gamarnik and Goldberg, 2010; Gamarnik, Goldberg, and Weber, 2014), marginal inference on graphical models (Tatikonda and Jordan, 2002), etc. The typical regime for static decision problems under which the correlation decay property holds is when the underlying graph has a bounded degree (i.e., each node interacts with a constant number

of other nodes) (Weitz, 2006; Gamarnik and Goldberg, 2010; Gamarnik, Goldberg, and Weber, 2014). However, in our multi-period network model involving uncertainty about the future, there exists an implicit interaction between *every* pair of nodes because their decisions for future periods are correlated, and hence the underlying interaction graph no longer has a bounded degree, which results in new challenges.

*Dynamic optimization in networks.* Dynamic decision-making in networks under uncertainty about the future has been studied in a variety of contexts including network revenue management (Talluri and Ryzin, 2006), network diffusion models (Leduc, Jackson, and Johari, 2017; Manshadi, Misra, and Rodilitz, 2020; Akbarpour, Malladi, and Saberi, 2018), online matching under stochastic arrivals (Aouad and Saritaç, 2020; Collina et al., 2020; Anderson et al., 2017; Sivan, 2013), and choosing lockdown policies in a commuting network (Fajgelbaum et al., 2020), to name a few. In most of these works the ideas used (e.g., shadow prices) are very different from ours. Some of these previous work (Besbes, Gur, and Zeevi, 2016; Aouad and Saritaç, 2020) make use of local decompositions that rely on their specific problem structures or require convexity in the reward functions (Lin et al., 2022). We adopt a general framework, towards developing a foundational understanding regarding the sufficiency of decentralized algorithms for obtaining near-optimality in dynamic stochastic optimization problems (with no convexity assumptions on the reward functions). Our paper contributes to this literature by providing an important theoretical foundation for decision-making problems on large networks: that is, even though the network interactions may be complicated and may evolve, considering only the local neighborhood around the focal node already gives near-optimal performance when the strength of interactions is not too large.

*Multi-agent reinforcement learning.* In the setting of multi-agent reinforcement learning (MARL), computational issues are central due to large global states and decision spaces (exponential in the number of agents). A promising approach is to exploit local dependency structures (i.e., "agents" only interact with neighboring agents in the network). Lin et al. (2020) consider a setting where the transition of the global state has only local dependencies and the problem objective is to maximize the discounted sum of global rewards. They propose an Actor-Critic method with state aggregation

defined based on the local neighborhood of each agent. Moreover, due to discounting, the Bellman operator with state aggregation defines a contraction mapping of the Q functions. They show the proposed algorithm converges to the *stationary point* of the global objective (note that a stationary point yields a local optimum, at best). In contrast, maximizing the average reward per period (as we consider) is fundamentally harder since the Bellman operator is no longer a contraction (Bertsekas, 2011; Tsitsiklis and Roy, 2002). In a recent advance for such a setting, Qu et al. (2020) establish convergence to an *approximately stationary point* of the Q functions. Still, this result does not guarantee global near-optimality. We contribute to this literature by showing, in a special case where the state transitions are simple (i.e., the current state is the previous decision), *global near-optimality* of a local policy for the average reward problem with non-stationary reward functions over an arbitrarily long horizon. Our work may serve as a starting point for developing a theoretical understanding of sufficient conditions for local algorithms achieving global near-optimality in general network Markov Decision Problem settings.

### 2.1.3 Notation and terminology

We denote our underlying graph as $G = (V, E)$ with node (or vertex) set $V$ and edge set $E$. Given a graph $G$, we denote by $V(G)$ and $E(G)$ its node set and edge set, respectively. For two nodes $u, v \in G$, we let $\text{dist}_G(u, v)$ denote the length of a shortest path between $u$ and $v$. If an edge $uv \in E$ (or equivalently denoted as $\{u, v\}$), we say $u$ is a *neighbor* of $v$. For any node $v \in V$, we denote by $\Gamma(v)$ its set of neighbors: $\Gamma(v) \coloneqq \{u \in V : uv \in E\}$; and denote by $d_G(v) \coloneqq |\Gamma(v)|$ its *degree* in $G$. We let $d_G$ denote the *degree* of graph $G$, which is the maximum degree of all nodes in graph $G$. For a subgraph $M$ of $G$ and a vector $y \coloneqq \{y^v\}_{v \in V}$ indexed by vertices of $G$, we denote by $y^M$ the subvector $\{y^v\}_{v \in V(M)}$. For integer $R \in \mathbb{N}_+$, let $B_G(v, R)$ denote the subgraph induced by all vertices whose distance to $v$ is at most $R$. When the underlying graph is clear from context, we drop the subscript for the above notations.

For a given integer $K \geq 1$, we use $[K]$ to denote the set $\{1, 2, \cdots, K\}$. Given a graph $G = (V, E)$ and time horizon $\mathcal{T}$, the spatial-temporal (ST) graph is constructed by making a

clone of $G$ for each time $t \in [\mathcal{T}]$, and connecting copies of the same node between consecutive times via edges. The nodes of the ST graphs are $\{(v,t) : v \in V, t \in [\mathcal{T}]\}$ and the edges are $\{\{(v_1,t),(v_2,t)\} : v_1v_2 \in E, t \in [\mathcal{T}]\} \cup \{\{(v,t),(v,t+1)\} : v \in V, t \in [\mathcal{T}-1]\}$. The ST graph distance is defined as follows: given two ST nodes $(v_1,t_1)$ and $(v_2,t_2)$, $\text{dist}^{\text{st}}((v_1,t_1),(v_2,t_2)) = \text{dist}(v_1,v_2) + |t_1 - t_2|$.

For a collection of random variables $Y_{[k]}$, $\sigma(Y_{[k]})$ denotes the smallest sigma algebra generated by $Y_{[k]}$.

### 2.2 Model

We consider a dynamic decision network $(G = (V,E), \Phi, \mathcal{T}, \mathcal{A}, x_0)$ with future stochastic uncertainty. Here, $G$ is an underlying undirected graph where individual decisions are made at each node. $\Phi$ denotes the joint stochastic reward functions over the graph $G$ during the planning horizon. We consider a discrete-time model from time 0 to the planning horizon $\mathcal{T}$. We denote by $\mathcal{A}$ the discrete set that the decision of each node must be chosen from. The initial decision vector taken on the network is given and is denoted by $x_0 \in \mathcal{A}^{|V|}$. The global objective is to maximize the collective (undiscounted) payoff from the entire graph over the time horizon. At the time $t$, the per-period reward is the sum of three types of (random) reward functions: In the following, we define each type of (random) reward functions and give an illustrating example when $\mathcal{A} = \{\mathtt{N}, \mathtt{Y}\}$.

- **Node reward**: Each node $v \in V$ earns a random reward $\Phi_t^v(x_t^v) : \mathcal{A} \to \mathbb{R}$, which depends on its decision $x_t^v$ at time $t$. E.g., $\mathcal{A} = \{\mathtt{N}, \mathtt{Y}\}$, $\Phi_t^v(\mathtt{N}) = 0$ and $\Phi_t^v(\mathtt{Y}) = 1 + \epsilon_t^v$ where $\epsilon_t^v \sim N(0,1)$.
- **Temporal interaction reward**: Each node $v \in V$ at each time $t$ is associated with a random reward function $\Phi_{t-1,t}^v(x_{t-1}^v, x_t^v) : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$, which captures how consecutive decisions at node $v$ interact with each other. E.g., $\Phi_{t-1,t}^v(\mathtt{N},\mathtt{N}) = \Phi_{t-1,t}^v(\mathtt{N},\mathtt{N}) = c$, $\Phi_{t-1,t}^v(\mathtt{N},\mathtt{Y}) = \Phi_{t-1,t}^v(\mathtt{Y},\mathtt{N}) = 0$ for some $c > 0$, i.e., there is a reward $c$ for making the same node decision in consecutive periods.
- **Spatial interaction reward**: Each edge $uv \in E$ at each time $t$ is associated with a random reward function $\Phi_t^{u,v}(x_t^u, x_t^v) : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$, which captures how neighboring nodes interact

with each other at time $t$. E.g., $\Phi_t^{uv}(\mathtt{N},\mathtt{N}) = \Phi_t^{uv}(\mathtt{Y},\mathtt{Y}) = c$, $\Phi_t^{uv}(\mathtt{N},\mathtt{Y}) = \Phi_t^{uv}(\mathtt{Y},\mathtt{N}) = 0$ for $c > 0$, i.e., there is a reward $c$ when neighboring nodes make the same decision.

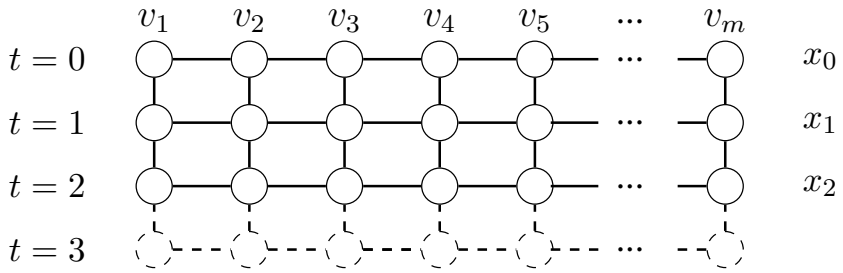


Figure 2.1: Decision dynamics.

Collectively, we let $\Phi \coloneqq \{\Phi_t^v, \Phi_{t-1,t}^v, \Phi_t^{uv}\}_{t\in[\mathcal{T}],v\in V,uv\in E}$ denote joint random reward functions. At each time $t \in [\mathcal{T}]$, node $v \in V$ makes a decision $x_t^v \in \mathcal{A} \coloneqq \{0, 1, \cdots, |\mathcal{A}| - 1\}$. These reward functions are endowed with a probabilistic structure: their function values are assumed to follow known distributions. The random functions $\Phi_t^{\text{node}} \coloneqq \{\Phi_t^v\}_{v\in V}$ and $\Phi_t^{\text{inter}} \coloneqq \{\Phi_{t-1,t}^v, \Phi_t^{uv}\}_{v\in V,uv\in E}$ are realized only at the beginning of time period $t$. We denote the realized reward functions as $\{\phi_t^v\}_{v\in V}$, $\{\phi_{t-1,t}^v\}_{v\in V}$ and $\{\phi_t^{uv}\}_{uv\in E}$. Moreover, we denote the random reward functions, and their realization, at time $t$, collectively, by $\Phi_t$ and $\phi_t$, respectively. Given any subgraph $M$ of $G$, let $\Phi_t^M \coloneqq \{\Phi_t^v, \Phi_{t-1,t}^v, \Phi_t^{uv}\}_{v\in V(M),uv\in E(M)}$ and $\phi_t^M \coloneqq \{\phi_t^v, \phi_{t-1,t}^v, \phi_t^{uv}\}_{v\in V(M),uv\in E(M)}$.

We call $x_t \coloneqq \{x_t^v\}_{v\in V}$ a *decision vector* at time $t$. At each time $t$, a decision vector $x_t$ must be chosen after observing $\phi_t$. We illustrate the dynamics under our model through an example in Figure 2.1 with an example of a dynamic decision network $(G = (V, E), \Phi, \mathcal{T}, \mathcal{A}, x_0)$ with $G$ being a line graph, $V = \{v_1, v_2, \cdots, v_m\}$, $E = \{v_{i-1}v_i : i \in \{2, 3, \cdots, m\}\}$, and $\mathcal{T} = 3$. At the time $t = 2$, with the previous decision vector $x_1$, realized rewards $\phi_1, \phi_2$ (represented by solid lines and circles), and unrealized reward $\Phi_3$ (represented by dotted lines and circles), decision vector $x_2$ needs to be chosen. Given realized reward function $\phi_t$ at time $t$, and decision vectors $x_{t-1}, x_t$, the *single-period reward* collected at period $t$ is

$$f_t(x_t; x_{t-1}, \phi_t) \coloneqq \sum_{v\in V} \phi_{t-1,t}^v(x_{t-1}^v, x_t^v) + \sum_{v\in V} \phi_t^v(x_t^v) + \sum_{uv\in E} \phi_t^{u,v}(x_t^u, x_t^v)\,. \tag{2.1}$$

The overall goal is to construct a dynamic decision-making policy $x_t$, which is adapted to the available information up to time $t$, i.e., $x_t \in \sigma(x_0, x_{[t-1]}, \Phi_{[t]})$, that maximizes the expected collected rewards over a finite horizon: $\mathcal{R} \coloneqq \mathbb{E}_\Phi \left[ \sum_{t=1}^{\mathcal{T}} f_t(x_t; x_{t-1}, \Phi_t) \right]$.

Following the modeling convention in dynamic stochastic optimization (Cao, Zhang, and Poor, 2021; Bent and Van Hentenryck, 2004), we assume that $\Phi_t$ is independent of past reward functions $\{\phi_{[t-1]}\}$. At time $t$, we observe the previous decision vector $x_{t-1}$ and current single-period reward realization, i.e., $\phi_t$. By the principle of optimality, the optimal $x_t(x_{t-1}, \phi_t)$ maximizes the *realized value-to-go* function: $\text{RV}_{t-1}(x_t; x_{t-1}, \phi_t) \coloneqq f_t(x_t; x_{t-1}, \phi_t) + V_t(x_t; \phi_t)$ , where the *expected value-to-go function* $V_t(x_t)$ is recursively defined as

$$V_t(x_t; \phi_t) \coloneqq \mathop{\mathbb{E}}_{\Phi_{t+1}} \Big[ \max_{x_{t+1}} \text{RV}_t(x_{t+1}; x_t, \Phi_{t+1}) \Big], \tag{2.2}$$

with $V_{\mathcal{T}}(x_{\mathcal{T}}) = 0$ at the end of the horizon. We denote by $x^* \coloneqq \{x_t^*\}_{1 \le t \le \mathcal{T}}$ the optimal solution of (2.2) and denote the optimal expected global reward as $\mathcal{R}^* \coloneqq \mathbb{E}_\Phi \Big[ \sum_{t=1}^{\mathcal{T}} f_t(x_t^*; x_{t-1}^*, \Phi_t) \Big]$. For any *adaptive algorithm* which makes decisions $\text{Alg}_t \in \sigma(x_0, x_{[t-1]}, \Phi_{[t]})$ at time $t$, we define the expected total rewards under Alg as $\mathcal{R}(\text{Alg}) \coloneqq \mathbb{E}_\Phi \Big[ \sum_{t=1}^{\mathcal{T}} f_t(\text{Alg}_t; \text{Alg}_{t-1}, \Phi_t) \Big]$, with $\text{Alg}_0 \equiv x_0$.

**Challenges.** Our problem presents two main challenges: 1) Uncertainty about future rewards functions is a source of complexity compared to a single-period problem. In a single-period problem with random rewards, one observes reward realizations at all parts of the network at once and then make the corresponding optimal decision. In our setup, reward realizations are revealed *sequentially* along the temporal dimension. When the reward function is realized at time $t$, we must commit to a decision vector at time $t$. Under each realization $\phi_t$, the optimal decision at time $t$, i.e., $x_t$ may be different and this further impacts the consequent optimization problem at time $t+1$. 2) Our problem can be viewed as a *dynamic programming* problem with state variable at time $t$ to be $x_{t-1}$. The state space is $|\mathcal{A}|^{|V(G)|}$, which is exponential in the number of nodes in the network. Thus, directly solving (2.2) above suffers from the curse of dimensionality and is impractical on large networks. Instead, we explore *decentralized* algorithms which make each decision based on

available information about the nearby part of the network (and only a few periods into the future), and find sufficient conditions under which such algorithms can achieve near-optimal collective reward compared to the global optimal value $\mathcal{R}^*$.

### 2.3 Main Results and Algorithms

We make the following assumptions on the reward functions.

**Assumption 2.3.1.** *For some constants $C_{\text{node}}, g, c_{\text{time}}, c_{\text{edge}} \in (0, \infty)$, the distributions of reward functions $\{\Phi_t\}_{t\in\mathcal{T}}$ satisfy the following:*

- **(Node rewards are bounded)** *For every $v \in V$ and $t \in [\mathcal{T}]$, $\sup_{a\in\mathcal{A}} |\Phi_t^v(a)| \le C_{\text{node}}$.*
- **(Reward functions are independent across time)** *For every $t \in [\mathcal{T}]$, $\Phi_t$ are independent of past decisions $x_{[t-1]}$ and past reward realizations $\phi_{[t-1]}$.*
- **(Each node reward function is sufficiently random after conditioning on interaction functions and other node rewards)** *For every $v \in V$, $t \in [\mathcal{T}]$, and decisions $a \neq a' \in \mathcal{A}$,*

$$\mathbb{P}(\Phi_t^v(a) - \Phi_t^v(a') \in [b_1, b_2) \mid \Phi_t^{\text{inter}}, \{\Phi_t^u\}_{u\neq v}) \le g(b_2 - b_1), \text{for any } b_1 < b_2.$$

- **(Interactions are not too strong)** *With probability 1, for every $v \in V$, $uv \in E$, $t \in [\mathcal{T}]$ and decisions $a \neq a' \in \mathcal{A}$, the temporal interaction $\Phi_{t-1,t}^v(a, a') \in [-c_{\text{time}}, c_{\text{time}}]$, and the edge interaction $\Phi_t^{u,v}(a, a') \in [-c_{\text{edge}}, c_{\text{edge}}]$. Moreover,*

$$\rho := 4g(dc_{\text{edge}} + 2c_{\text{time}}) \;\le\; \frac{1}{2(d+2)}\,.$$

We now provide intuitive interpretations for these assumptions. 1) The first condition provides a uniform bound for the change in the global reward when a single node switches its single-period decision. This assumption ensures that there is no single node whose decision at a certain period

has a dominant impact on the global reward. It is a standard assumption in the dynamic programming and reinforcement learning literature (Tsitsiklis and Van Roy, 1996). 2) The second condition demands independence of the rewards across time. E.g., considering the reward functions described in Section 2 where $\mathcal{A} = \{\mathtt{N}, \mathtt{Y}\}$, $\Phi_t^v(\mathtt{N}) = 0$, $\Phi_t^v(\mathtt{Y}) = 1 + \epsilon_t^v$ where $\epsilon_t^v \sim N(0,1)$, this assumption requires the joint distributions of $\{\epsilon_t^v : v \in V\}_{t\geq 1}$ to be independent across periods $t$. Independence across time periods is a classic assumption in the dynamic (stochastic) optimization setting (Bertsekas, 1995), and serves to simplify the technical development in our paper. 3) The third condition guarantees sufficient randomness in the single-node reward function at each period. For example, considering the same reward functions, we require $\{\epsilon_t^v : v \in V\}$ to satisfy that for each $v \in V$, $t \in [\mathcal{T}]$ and $b_1 < b_2$, $\mathbb{P}(\epsilon_t^v \in [b_1, b_2) \mid \epsilon_t^u : u \neq v) \leq g(b_2 - b_1)$. Note that this requirement allows reward functions at a given period to have arbitrary dependence across nodes. 4) The last condition requires that interactions are not too strong compared to the randomness in node rewards. This condition (along with the sufficient randomness condition) is crucial for the *correlation decay* property to emerge. In situations where there are long-range correlations, decentralized algorithms are in general not near-optimal (Weber, 2010; Gamarnik, 2014). Previous work (Gamarnik, Goldberg, and Weber, 2014) studies a *static* version of our model and assumes the magnitude of interactions is bounded by $c_{\text{edge}} = O(\frac{1}{d^2})$. Our assumption that $dc_{\text{edge}} + 2c_{\text{time}} = O(\frac{1}{d})$ matches their requirement in the static case. That is, we do not require a stronger assumption on the magnitude of interactions to accommodate the dynamic setting. Admittedly, this requirement becomes more stringent as graph degree $d$ increases. However, dependence on $d$ is unavoidable: In Section 2.10, we explicitly construct (static) decision networks with $c_{\text{edge}} = \Theta(1/d)$ which exhibit long-range correlations and show that local algorithms can perform poorly on such networks.

### 2.3.1 Main theorem

**Definition 2.3.1.** *An algorithm for the dynamic decision network $(G, \Phi, \mathcal{T}, \mathcal{A}, x_0)$ is said to be an* $L$-local algorithm *if: for the decision of any node $v$ at time $t$ the followings hold 1) The decision only relies on the local information up to its $L$-radius neighborhood in the graph $G$, i.e.,*

$x_t^v \in \sigma(x_{t-1}^{B(v,L)}, \Phi_t^{B(v,L)})$*; 2) The decision only relies on* $L$*-period-lookahead* simulations *of future reward functions in this neighborhood, i.e.,* $\{\Phi_\tau^{B(v,L)} : t+1 \le \tau \le t+L\}$.

**Definition 2.3.2.** *Consider a dynamic decision network* $(G, \Phi, \mathcal{T}, \mathcal{A}, x_0)$*. An algorithm* Alg *is an* $\epsilon$(-additive)-approximation *algorithm if* $\mathcal{R}^* - \mathcal{R}(\text{Alg}) \le |V|\mathcal{T}\epsilon$*, where* $\mathcal{R}^*$ *is the global optimal payoff, and* $\mathcal{R}(\text{Alg})$ *is the payoff collected by* Alg.

Note that there is a $|V|\mathcal{T}$ factor in the loss permitted because the total reward scales up linearly with the number of nodes $|V|$ times the time horizon $\mathcal{T}$; in other words, we permit an average per-node-per-period loss of up to $\epsilon$.

We next introduce a model parameter $C$, which is the largest possible change in total rewards when one node changes its decision at one time. For any $a, a' \in \mathcal{A}$, changing from $x_t^v = a$ to $x_t^v = a'$ can cause at most $2 \cdot C_{\text{node}}$ difference in the node reward, $d \cdot 2c_{\text{edge}}$ difference in the edge rewards, and $2 \cdot 2c_{\text{time}}$ difference in the temporal rewards. Hence, we define the constant

$$C \coloneqq 2C_{\text{node}} + 2dc_{\text{edge}} + 4c_{\text{time}}. \tag{2.3}$$

**Theorem 2.3.1.** *Consider a dynamic decision network* $(G, \Phi, \mathcal{T}, \mathcal{A}, x_0)$ *where underlying graph* $G$ *has degree* $d \ge 2$*. Suppose the reward functions* $\Phi$ *satisfy Assumption 2.3.1. Then, given any* $\epsilon > 0$*, for* $L \triangleq \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$*, we can construct an* $L$*-local algorithm for the dynamic decision network problem that is an* $\epsilon$*-approximation algorithm.*

The main contribution of 2.3.1 is *theoretical*, which establishes the *global near-optimality* property of the class of *decentralized* algorithms in the network dynamic optimization setting under Assumption 2.3.1. The constructed $L$-local algorithm (presented in 1) is a simple and intuitive illustration of how one may use 2.3.1 to develop efficient decentralized algorithms: the computational requirement of 1 is $O(|V|\mathcal{T}e^{\text{poly}(\frac{1}{\epsilon})})$, where the dependence on model parameters $d, g, C$ and $|\mathcal{A}|$ is suppressed in the $O(\cdot)$ notation.

The proof of 2.3.1 and the details on the computational requirement of an $L$-local algorithm are presented in Section 2.4 and 2.5, respectively.

### 2.3.2 Local Algorithm

In this section, we present our local algorithm. Given $t \in [\mathcal{T}]$, the global decision problem is

$$\max_{x_t} \; f_t(x_t; x_{t-1}, \phi_t) + V_t(x_t). \tag{2.4}$$

The algorithm, outlined in Algorithm 1, determines the decision of each node by solving a *decentralized* version of (2.4). For each node $v$, the local algorithm utilizes all available reward information from its local neighborhood $B(v, L)$ and fixes the decision of each boundary node $u \in B(v, L) \setminus B(v, L-1)$ as a default decision 0. We define

$$f_t^L(x_t^{B(v,L)}; x_{t-1}^{B(v,L)}, \phi_t^{B(v,L)}) \coloneqq \sum_{u \in B(v,L)} (\phi_t^u(x_t^u) + \phi_{t-1,t}^u(x_{t-1}^u, x_t^u)) + \sum_{uu' \in B(v,L)} \phi_t^{u,u'}(x_t^u, x_t^{u'}) \tag{2.5}$$

as the single-period payoff on $B(v, L)$ and

$$V_t^L(x_t^{B(v,L)}) \coloneqq \mathop{\mathbb{E}}_{\Phi_{t+1}} [\max_{x_{t+1}^{B(v,L)}} f_{t+1}^L(x_{t+1}^{B(v,L)}; x_t^{B(v,L)}, \phi_{t+1}^{B(v,L)}) + V_{t+1}^L(x_{t+1}^{B(v,L)})] \tag{2.6}$$

with terminal condition $V^L_{\min\{t+L,\mathcal{T}\}}(\cdot) = 0$ as the expected value-to-go function up to $L$-step look-ahead on $B(v, L)$. In addition, we denote by $\widehat{V}_t^{L,n}(x_t^{B(v,L)})$ the sample average estimate of $V_t^L(x_t^{B(v,L)})$ by simulating independent samples of $\Phi_{t+1}^{B(v,L)}$, $\Phi_{t+2}^{B(v,L)}$, $\cdots$, $\Phi_{\min(\mathcal{T},t+L)}^{B(v,L)}$. In the remainder of the paper, for simplicity of notation, we omit to make the dependency on $v$ and $B(v, L)$ explicit when the focal node $v$ for the decentralized algorithm is clear from the context.

We illustrate the algorithm output $\text{Alg}_t^v$ at time step $t$ for a given focal node $v \in V$ with $L = 1$ on the underlying line graph $G$ in Figure 2.2, where the black circles, red lines, and blue lines denote the individual node reward functions, spatial interaction functions, and temporal interaction functions, respectively. The reward functions, either their realizations or simulations, are marked by solid lines when solving Equation (2.7).

**Algorithm 1** Obtain a near-optimal solution to the decision problem (2.4) at time $t$.

**Input:** decision network $(G, \Phi, \mathcal{T}, \mathcal{A}, \text{Alg}_{t-1})$, realized reward function $\phi_t$, precision level $\epsilon$.
**Output:** a near-optimal solution $\text{Alg}_t$ for the problem in (2.4).

1: set the locality parameter $L = \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$ and sample size $n = O((\frac{1}{\epsilon})^{2\log_2 d})$
2: **for all** $v \in V$ **do**
3: restrict to subgraph $B(v, L)$
4: let $x_{t-1} = \text{Alg}_{t-1}$ and $y_t \in \mathcal{A}^{B(v,L)}$ be an optimal solution to the following problem

$$\begin{aligned} \widehat{\text{RV}}_{t-1}^{L,n}(x_{t-1}; \phi_t) \coloneqq \max_{x_t \in \mathcal{A}^{B(v,L)}} \quad & f_t^L(x_t; x_{t-1}, \phi_t) + \widehat{V}_t^{L,n}(x_t) \\ \text{s.t.} \quad & x_t^u = 0, \quad \text{if dist}(v, u) = L \end{aligned} \tag{2.7}$$

where $\widehat{V}_t^{L,n}(\cdot)$ is an estimate of $V_t(\cdot)$ defined recursively in **function** $\widehat{V}_\tau^{L,n}(\cdot)$ for $t \le \tau \le t + L$
5: set $\text{Alg}_t^v = y_t^v$
6: **end for**

---

1: **function** $\widehat{V}_\tau^{L,n}(x_\tau)$ ▷ **Input**: $L$, $n$, $v$, $\tau$.
2: **if** $\tau = \min\{t + L, \mathcal{T}\}$ **then**
3: set $\widehat{V}_\tau^{L,n}(x_\tau) = 0$ for any $x_\tau \in \mathcal{A}^{B(v,L)}$
4: **else**
5: sample $\{\phi_{\tau+1}^{(s)}\}_{s\in[n]}$ independently from $\Phi_{\tau+1}$
6: for any $x_\tau \in \mathcal{A}^{B(v,L)}$, compute $\widehat{V}_\tau^{L,n}(x_\tau) \coloneqq \frac{1}{n}\sum_{s=1}^n \widehat{\text{RV}}_\tau^{L,n}(x_\tau; \phi_{\tau+1}^{(s)})$ where the summand $\widehat{\text{RV}}_\tau^{L,n}(x_\tau; \phi_{\tau+1}^{(s)})$ is defined as in Equation (2.7) with $t = \tau + 1$.
7: **end if**
8: **end function**

---

## 2.4 Proof Outline of Theorem 2.3.1

In this section, we outline the steps to establish the $\epsilon$-approximation result in Theorem 2.3.1. We show that with high probability, Algorithm 1 outputs the globally optimal decision at a given focal node. Our analysis proceeds in two steps. The main technical contribution is the first step, where we construct a sequence of local dynamic optimization problems with increasing local radius. We use the term *locality loss* to refer to the probability of making a suboptimal decision at the focal node due to fixing the decision boundary of its $L$-local neighborhood suboptimally, e.g., to a default decision $0$. This is the loss which is the unavoidable consequence of local decision-making. We present our argument bounding the locality loss in Section 2.4.1. The second step is to

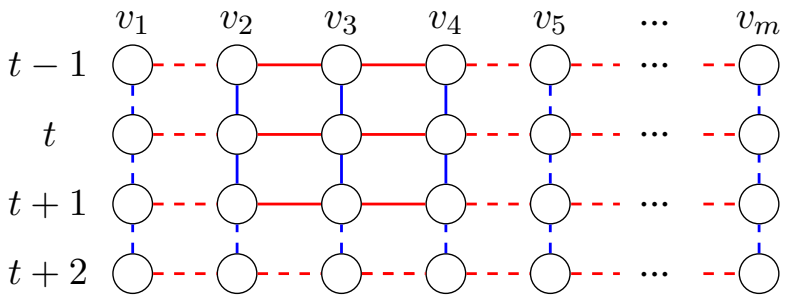


Figure 2.2: Illustration of Algorithm 1

bound the probability, termed the *sampling loss*, of making a suboptimal decision at the focal node as a result of using approximate (local) value-to-go functions estimated from sample averages. The second step relies on standard techniques such as Hoeffding's inequality, which we present in Section 2.4.2.

### 2.4.1 Bounding the Locality Loss

In this subsection, we bound the loss that is unavoidable from local decision-making, even if one is able to perfectly estimate the local value-to-go functions. We define a sequence of decentralized policies $\{\pi(H)\}_{H \geq L}$, indexed by the locality parameter $H$. Note that a policy defines a mapping from available information so far to decision vectors. We use $\pi_t^v(H)$ to denote the decision of node $v$ at time $t$ under policy $\pi(H)$, and collectively, we use $\pi_t(H)$ to denote the decision vector at time $t$.

When solving for the decision at a focal node $v \in V$ and time $t \in [\mathcal{T}]$, the policy $\pi(H)$ focuses on the subgraph $B(v, H) \subset G$. It assumes nodes outside of $B(v, H)$ take a default decision 0 at all times. Along the temporal dimension, the policy $\pi(H)$ computes $V_t^H(\cdot)$, an estimate of the (local) value-to-go function, via an *$H$-step look-ahead* with the terminal condition $V^H_{\min(\mathcal{T}, t+H)}(x) = 0$ for all decision vectors $x \in \mathcal{A}^{B(v,H)}$. Formally, for a given focal node $v$ at time $t$, $\pi(H)$ solves the following:

$$\begin{aligned} \text{RV}_{t-1}^H(\pi_{t-1}(H); \phi_t) := \max_{x_t \in \mathcal{A}^{B(v,H)}} \quad & f_t^H(x_t; \pi_{t-1}(H), \phi_t) + V_t^H(x_t) \\ \text{s.t.} \quad & x_t^u = 0 \quad \text{if } \text{dist}(v, u) = H\,. \end{aligned} \tag{2.8}$$

where the $H$-step look-ahead value-to-go $V_t^H(x_t;\phi_t)$ in the objective is defined recursively via

$$V_\tau^H(x_\tau;\phi_\tau) \coloneqq \mathop{\mathbb{E}}_{\Phi_{\tau+1}}[\mathrm{RV}_\tau^H(x_\tau;\Phi_{\tau+1})], \tag{2.9}$$

for $t \le \tau < \min\{t+H, \mathcal{T}\}$.

Recall that $C \coloneqq 2C_{\text{node}} + 2dc_{\text{edge}} + 4c_{\text{time}}$ and $\{x_t^*\}_{t\in[\mathcal{T}]}$ denotes the global optimal decision. In this subsection, the probability space is over the joint reward distributions $\Phi = (\Phi_1, \cdots, \Phi_{\mathcal{T}})$. We write $\mathbb{P}$ as a shorthand for $\mathbb{P}_\Phi$.

**Proposition 2.4.1.** *Given any $\epsilon > 0$, with $L = \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$, we have for any $v \in V$ and $t \in [\mathcal{T}]$,*

$$\mathbb{P}(\pi_t^v(L) \neq (x_t^v)^*) \le \epsilon/(2C).$$

Proposition 2.4.1 establishes that the probability of policy $\pi(L)$ making a suboptimal decision at the focal node $v$ is exponentially small in the locality parameter $L$. This is a strong statement that upper bounds the marginal probability of different decisions at time $t$ at node $v$, under the $L$-local policy and the global optimal policy, without assuming the decisions at previous time steps are equal. In the remaining subsection, we present two important lemmas that prove Proposition 2.4.1. We consider a fixed $t \in [\mathcal{T}]$, a fixed focal node $v \in V$, and a fixed value $H$ and then compare the focal node decisions at time $t$ obtained under two local policies $\pi(H)$ and $\pi(H+1)$ (recall that under policy $\pi(H)$ (resp. $\pi(H+1)$), the locality parameter is fixed to $H$ (resp. $H+1$) throughout, i.e., from $t = 0$ onwards.). Formally, we compare the solutions of the optimization problems in (2.8) when setting the locality parameter as $H$ and $H+1$. For $t \le \tau \le t+H$, we let $w_\tau$ denote the optimal solution of (2.8) at time $\tau$ when the locality parameter is $H$; and similarly, we let $z_\tau$ denote the optimal solution of (2.8) at time $\tau$ when the locality parameter is $H+1$. Moreover, we use $w_{t-1}$ and $z_{t-1}$ as shorthand notations for the decision vectors $\pi_{t-1}(H)$ and $\pi_{t-1}(H+1)$, respectively. For convenience, we extend the definition of $z_\tau^u$ (resp. $w_\tau^u$) to the entire network by setting $z_\tau^u = 0$ for $u \in V \setminus B(v, H+1)$ (resp. $w_\tau^u = 0$ for $u \in V \setminus B(v, H)$), and this does not change our original optimization problem in Equation (2.8). Note that when $\tau > t$, $\{w_\tau\}_{t+1\le\tau\le t+H}$ and $\{z_\tau\}_{t+1\le\tau\le t+H}$

are the optimal "tentative" decisions from time $t+1$ to time $t+H$ under $\pi(H)$ and $\pi(H+1)$. That is, at time $t$, $\pi(H)$ (resp. $\pi(H+1)$) only executes $w_t$ (resp. $z_t$) and discards the other decision vectors $\{w_\tau\}_{t+1\le\tau\le t+H}$ (resp. $\{z_\tau\}_{t+1\le\tau\le t+H}$). Moreover, for $t+1\le\tau\le t+H$, note that $z_\tau$ and $w_\tau$ are random vectors which are measurable with respect to realized random rewards up to time $t$, i.e., $\sigma(\Phi_{[t]})$. Recall that $\rho \coloneqq 4g(dc_{\text{edge}}+2c_{\text{time}})$ and $\Gamma(v)$ denotes the neighbors of $v$.

**Lemma 2.4.1.** *For time $t\le\tau\le t+H$, and $u\in V$,*

$$\mathbb{P}(w_\tau^u\neq z_\tau^u, w_\tau^{\Gamma(u)}=z_\tau^{\Gamma(u)})\le(\mathbb{P}(w_{\tau-1}^u\neq z_{\tau-1}^u)+\mathbb{P}(w_{\tau+1}^u\neq z_{\tau+1}^u))\rho.$$

We first look at a special case to get some intuitive understanding of the above lemma: if both $\mathbb{P}(w_{\tau-1}^u \neq z_{\tau-1}^u)$ and $\mathbb{P}(w_{\tau+1}^u \neq z_{\tau+1}^u)$ are equal to zero, then Lemma 2.4.1 implies $\mathbb{P}(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)}) = 0$. This reflects the fact that given an ST node $(\tau,u)$ in the ST graph constructed from $G$, if all immediate neighbors (i.e., spatial neighbors $\Gamma(u)$, temporal neighbors $(\tau-1,u)$ and $(\tau+1,u)$) take the same decisions under $\pi(H)$ and $\pi(H+1)$, then by principle of optimality, ST node $(\tau,u)$ take the same optimal decision under the above two policies. The lemma constitutes the key element of our analysis where we overcome the challenge of analyzing dynamics with uncertainty about the future. Note that $w_{\tau+1}$ and $z_{\tau+1}$ are as yet unrealized when $w_\tau$ and $z_\tau$ are determined. Instead of bounding the probability of the focal node taking a suboptimal decision when $k$-hop neighbors ($2\le k\le H$) in the (static) spatial graph take suboptimal decisions, we bound this probability in the ST graph since the interactions among nodes in the ST graph are easier to track. In the ST graph, a node makes a suboptimal decision only if a spatial or temporal neighbor is fixed suboptimally. The rigorous proof of Lemma 2.4.1 is more involved. It argues that the event $(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)})$ happens only if the value difference of single-node reward function due to taking different actions, i.e., $\Phi_\tau^u(w_\tau^u)-\Phi_\tau^u(z_\tau^u)$ falls in a small interval whose length is proportional to $c_{\text{time}}(\mathbb{I}\{w_{\tau-1}^u \neq z_{\tau-1}^u\}+\mathbb{E}_{\Phi_{\tau+1}}[\mathbb{I}\{w_{\tau+1}^u \neq z_{\tau+1}^u\}])$. By the third condition in Assumption 1, the probability of the event $(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)})$ is bounded above by quantity proportional to $g\cdot c_{\text{time}}(\mathbb{I}\{w_{\tau-1}^u \neq z_{\tau-1}^u\}+\mathbb{E}_{\Phi_{\tau+1}}[\mathbb{I}\{w_{\tau+1}^u \neq z_{\tau+1}^u\}])$, which further

leads to the inequality in Lemma 2.4.1. We present the details in the Section 2.8.1.

After obtaining Lemma 2.4.1, we use induction on the ST graph distance to node $(v, t)$ to upper bound the probability of making different node decisions under $\pi(H)$ and $\pi(H+1)$. We illustrate our proof ideas in Figure 2.3 and defer the proof of Lemma 2.4.2 to Section 2.8.2. In Figure 2.3: suppose we have proved Equation (2.11) for ST nodes $(q, t')$ with $\text{dist}^{\text{st}}((v,t),(q,t')) \geq 3$ (drawn as circles) and in the induction step, consider each ST node $(u, \tau)$ with $\text{dist}^{\text{st}}((v,t),(u,\tau)) \leq 2$ (drawn as diamonds). By induction hypothesis, each of the $\leq d+2$ ST neighbors of $(u, \tau)$ has probability $\leq \xi^{H+1-3}\rho = \xi^3\rho$ taking different solutions under $\pi(H)$ and $\pi(H+1)$. By Lemma 2.4.1, $\mathbb{P}(z_\tau^u \neq w_\tau^u) = \mathbb{P}(z_\tau^{\Gamma(u)} \neq w_\tau^{\Gamma(u)})\mathbb{P}(z_\tau^u \neq w_\tau^u | z_\tau^{\Gamma(u)} \neq w_\tau^{\Gamma(u)}) + \mathbb{P}(z_\tau^{\Gamma(u)} = w_\tau^{\Gamma(u)})\mathbb{P}(z_\tau^u \neq w_\tau^u | z_\tau^{\Gamma(u)} = w_\tau^{\Gamma(u)}) \leq d \cdot \xi^3\rho \cdot \mathbb{P}(z_\tau^u \neq w_\tau^u | z_\tau^{\Gamma(u)} \neq w_\tau^{\Gamma(u)}) + 1 \cdot (\xi^3\rho + \xi^3\rho) \cdot \rho \leq \xi^4\rho$, where the last inequality holds since $\mathbb{P}(z_\tau^u \neq w_\tau^u | z_\tau^{\Gamma(u)} \neq w_\tau^{\Gamma(u)}) \leq \rho$ (proved in Claim 1 in Section 2.8.2).

For ease of notation, we define another model parameter that is less than or equal to $1/2$ by Assumption 2.3.1

$$\xi := (d+2)\rho. \tag{2.10}$$

**Lemma 2.4.2.** *For $t \leq \tau \leq t+H$ and $u \in B(v, H)$, we have*

$$\mathbb{P}_\Phi(w_\tau^u \neq z_\tau^u) \leq \xi^{H+1-\text{dist}^{\text{st}}((v,t),(u,\tau))}\rho. \tag{2.11}$$

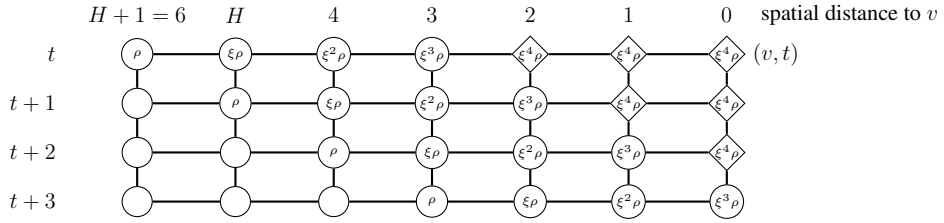


Figure 2.3: Pictorial example of the induction argument of Lemma 2.4.2

Proof of Proposition 2.4.1. We first use Lemma 2.4.2 for $(u, \tau) = (v, t)$ and obtain

$$\mathbb{P}(\pi_t^v(H) \neq \pi_t^v(H+1)) \leq \xi^{H+1}\rho.$$

Then, observe that when the locality parameter $H = +\infty$, we obtain the optimal node decision $(x_t^v)^*$. Finally, we use a union bound over all locality parameters $H$ which is greater than or equal to $L$.

$$\mathbb{P}(\pi_t^v(L) \neq \pi_t^v(+\infty)) \leq \sum_{H \geq L} \mathbb{P}(\pi_t^v(H) \neq \pi_t^v(H+1)) \leq \sum_{H \geq L} \xi^{H+1}\rho \leq 2\xi^{L+1}\rho \leq \epsilon/(2C)$$

since $\xi \leq 1/2$ by Assumption 2.3.1, $L = \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$ and $\rho < 1$.

### 2.4.2 Bounding the Sampling Loss

In this section, we aim to bound the loss in the total rewards due to approximating the expected (local) value-to-go function using sample averages. The main result of this subsection is given in Proposition 2.4.2, which states that Algorithm 1 obtains the same solution as the local policy $\pi(L)$ with high probability.

**Proposition 2.4.2.** *Under the conditions in the Theorem 2.3.1, given any $\epsilon > 0$, there exists a function $N = N(\epsilon, d, g, C) = O((\frac{4C}{\epsilon})^{2\log_2 d} g^2 C^4) < \infty$ such that if sample size $n \geq N$, then for any $v \in V$, $t \geq 0$,*

$$\mathbb{P}(\pi_t^v(L) \neq \text{Alg}_t^v) \leq \epsilon/(2C).$$

Similar to the idea used in the proof of Lemma 2.4.1, the event $\pi_t^v(L) \neq \text{Alg}_t^v$ happens only if the difference in the value of the single-node reward function under different actions (i.e., $\Phi_t^v(\pi_t^v(L)) - \Phi_t^v(\text{Alg}_t^v)$) falls in a small interval whose length is proportional to $\max_{x \in \mathcal{A}^{B(v,L)}}(V_t^L(x) - \widehat{V}_t^{L,n}(x)) + \max_{x \in \mathcal{A}^{B(v,L)}}(\widehat{V}_t^{L,n}(x) - V_t^L(x))$. We derive a recursive bound for the latter quantity in terms of value-to-go functions in the next period; in particular, we show that $\mathbb{E}_\Phi[\max_{x \in \mathcal{A}^{B(v,L)}}(V_t^L(x) - \widehat{V}_t^{L,n}(x))] + \mathbb{E}_\Phi[\max_{x \in \mathcal{A}^{B(v,L)}}(\widehat{V}_t^{L,n}(x)) - V_t^L(x)]$ is bounded above by $\mathbb{E}_\Phi[\max_{x \in \mathcal{A}^{B(v,L)}}(V_{t+1}^L(x) - \widehat{V}_{t+1}^{L,n}(x))] + \mathbb{E}_\Phi[\max_{x \in \mathcal{A}^{B(v,L)}}(\widehat{V}_{t+1}^{L,n}(x)) - V_{t+1}^L(x)]$ plus a small error. Since the terminal condition of the sample averages and expected (local) value-to-go functions are known, i.e., we have $\widehat{V}_{t+L}^{L,n}(x) = V_{t+L}^L(x) = 0$ for any $x \in \mathcal{A}^{B(v,L)}$, we are able to bound both $\mathbb{E}_\Phi[\max_{x \in \mathcal{A}^{B(v,L)}}(V_t^L(x) - \widehat{V}_t^{L,n}(x))]$ and $\mathbb{E}_\Phi[\max_{x \in \mathcal{A}^{B(v,L)}}(\widehat{V}_t^{L,n}(x)) - V_t^L(x)]$. The proof of Proposition 2.4.2 is in Sec-

tion 2.8.3.

### 2.4.3 Bounding the total loss

We now show Algorithm 1 achieves the global near-optimal reward by establishing Theorem 2.3.1.

Proof of Theorem 2.3.1. As a result of Proposition 2.4.1 and Proposition 2.4.2, we have that for all $v \in V$, and $t \in [\mathcal{T}]$,

$$\mathbb{P}(\text{Alg}_t^v \neq (x_t^v)^*) \leq \mathbb{P}(\pi_t^v(L) \neq (x_t^v)^*) + \mathbb{P}(\text{Alg}_t^v \neq \pi_t^v(L)) \leq \epsilon/C.$$

Recall the definition of $C$ in Equation (2.3). Since the largest possible change in total rewards when switching from one node action to another is upper bounded by $C$,

$$\begin{aligned} |\mathcal{R}(ALG) - \mathcal{R}^*| &\leq \sum_{v \in V, t \in [\mathcal{T}]} \mathbb{P}(\text{Alg}_t^v \neq (x_t^v)^*) C \\ &\leq \epsilon \cdot |V| \mathcal{T}. \end{aligned}$$

## 2.5 Computation Efficiency

In this section, we investigate the computation requirement of Algorithm 1. In terms of sampling, Algorithm 1 needs to simulate $n = O((\frac{1}{\epsilon})^{2\log_2 d})$ samples from each of the reward functions $\{\Phi_{t+1}, \Phi_{t+2}, \cdots, \Phi_{t+L}\}$. We illustrate these sample paths in Figure 2.4. Each node in this tree is a random vector consisting of reward functions in $B(v, L)$. The following proposition shows the computation requirement of Algorithm 1 for deciding the action of node $v$ at each time step $t$.

**Proposition 2.5.1.** *The computational requirement of Algorithm 1 is $O(|V|\mathcal{T}\text{e}^{\text{poly}(\frac{1}{\epsilon})})$, where the model parameters $d, g, C$ and $|\mathcal{A}|$ are constants in the $O(\cdot)$ notation.*

Algorithm 1 is decentralized and can compute decisions of each node $v \in V(G)$ at time $t$ in parallel. The overall computation scales linearly with the number of nodes in the graph $G$

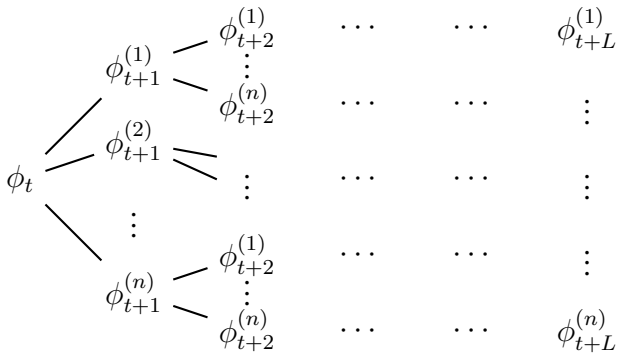

Figure 2.4: Illustration of approximating for value-to-go estimation via sample averages.

and the length of horizon $\mathcal{T}$. Without assuming any problem-specific structure, we can solve the *local optimization problem* presented in the (2.7) by enumerating over node actions, which takes $O(L|\mathcal{A}|^{2d^L}n) = O(\mathrm{e}^{\mathrm{poly}(\frac{1}{\epsilon})})$ computation. We present the proof details in Section 2.9.

In terms of practical implementation, we note that under our framework, one may tailor the solution method for the $L$-local problem based on problem-specific structure to reduce the computational requirement. In Section 2.6, we provide an example where we combine mixed integer programming and decentralized computing for solving a discrete optimization problem on a large network. Another example is related to deep reinforcement learning problems, where the optimal decisions can often be trained through a deep neural network with the input as states and the output as actions. However, when the state/action space is large, training a densely connected neural net requires a huge number of parameters and runs the risk of overfitting. When Assumption 1 is satisfied, Theorem 2.3.1 suggests, e.g., the use of convolutional layers instead of densely connected layers to reduce the number of training parameters, while preserving the ability of the neural network to express a near optimal solution.

## 2.6 Numerical Experiments

To test the presence (or absence) of correlation decay and the consequent success (or failure) of local algorithms while varying interaction strength, we conducted a numerical experiment. We summarize the simulation environment and main findings first and defer the details to Section 2.11.

Since our paper focuses on developing a theoretical understanding of where local algorithms can (provably) perform well, this section serves to empirically confirm our main insight, and does not attempt to develop new algorithms. Hence, the exact method through which one solves the local optimization problem is not of relevance, and in this section, we opt for solving optimization problems via *Mixed Integer Program* (MIP) based on our problem structure.

In this experiment, we first generate multiple dynamic decision networks, parameterized by the same interaction strength $c$ in both the temporal and spatial dimensions. These decision networks share all other components (e.g., a random 3-regular graph as the underlying graph, binary action set $\mathcal{A} = \{0, 1\}$, uniform distribution on $[-1, 1]$ as the single node reward distribution when taking action 1) so that the differences in performance can be solely attributed to the interaction strength $c$. For each decision network, multiple instances are generated by sampling the first-period node rewards (allowing us to bootstrap confidence intervals for the performance). Then, for each instance, we compute several solutions: one being the solution to the global optimization problem, and the others obtained by our local algorithms with different choices of the locality parameter. We formulate each global or local network optimization problem as a MIP and solve it through Gurobi (Gurobi Optimization, LLC, 2022). Lastly, for each solution, we compute its *relative payoff*, which is the ratio between the total payoff under the local solution to that under the global optimal solution.

We summarize the results in Figure 2.5, with one plot showing simply the relative payoffs (the higher, the better) and the other showing the relative payoff gaps ($1 -$ relative payoff) in log scale (the lower, the better). The experiment controls the sampling loss and thus, the loss in total reward is solely due to the locality loss. The vertical line on each data point represents its 95% confidence interval. The global optimal solutions correspond to the local solutions when the locality parameter equals 12, which is the *diameter* of the underlying graph. Our experimental results corroborate our theoretical finding in Theorem 2.3.1; when the interaction strength is small (or even medium-sized), the error in payoffs is seen to decay exponentially in the locality parameter. This is especially prominent in the second plot in Figure 2.5. We also observe that as the interaction

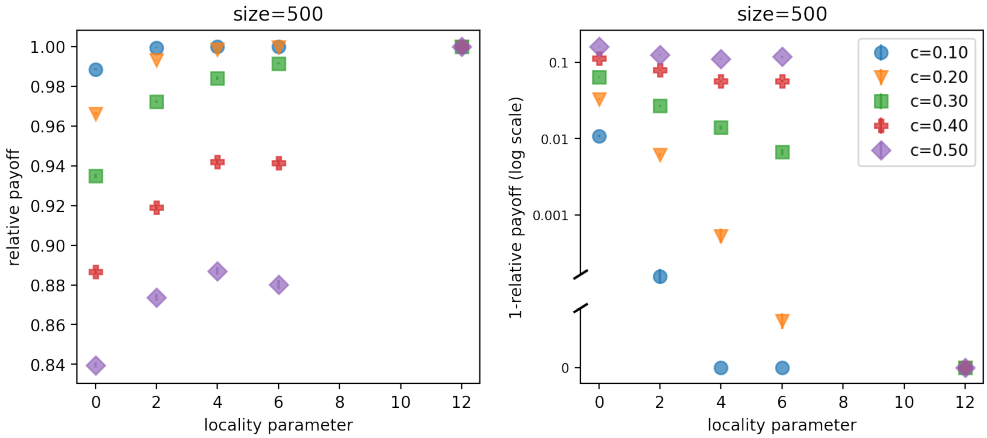


Figure 2.5: Payoffs from solutions under local algorithms and the global optimal solution.

strength gets larger, the optimality gap of the local algorithms increases: for a fixed locality parameter, the larger the interaction strength is, the smaller the relative payoff is. In addition, we observe that when the interaction strength is large $c \geq 0.4$, the optimality gap ceases to improve (and remains non-trivial) for locality parameter values larger than $4$. This suggests that in the setup of our experiment, long-range correlations arise for interaction strength satisfies $c \geq 0.4$, hurting the performance of local decision-making.

## 2.7 Concluding Remarks

We introduced a benchmark model of the dynamic optimization problem in networks where the global payoff includes spatial interactions and temporal interactions as well as individual node rewards. At each time step, a decision vector has to be chosen for all parts of the network under stochastic uncertainty about future reward functions. We establish that under small to moderate interaction strengths, decentralized algorithms – which make dynamic decisions only using information about the nearby part of the network – achieve *global near-optimality*. This theoretical insight is of importance for network applications, where decentralized algorithms can provide significant computational benefits, and moreover, do so (as we show) without sacrificing performance.

## 2.8 Proof details in Section 2.4

In this section, we present the details omitted in the proof outline.

### 2.8.1 Proof of Lemma 2.4.1.

For any node $u$ with $\text{dist}(v,u) > H+1$, we have defined $z_\tau^u = w_\tau^u = 0$. Thus, we consider node $u$ such that $\text{dist}(v,u) \le H+1$. Define the local version of $f_\tau(\cdot)$ function concerning node $u$,

$$f_\tau^u(x^u; x_\tau^{\Gamma(u)}, x_{\tau-1}^u, \Phi_\tau) \coloneqq \Phi_{\tau-1,\tau}^u(x_{\tau-1}^u, x^u) + \Phi_\tau^u(x^u) + \sum_{q\in\Gamma(u)} \Phi_\tau^{u,q}(x_\tau^u, x_\tau^q).$$

Let $z_\tau^{-u}$ (resp., $w_\tau^{-u}$) denote the actions at all nodes other than $u$ under the vector $z_\tau$ (resp., $w_\tau$). Since $z_\tau^u$ is the optimal solution at time $\tau$ when restricting to the subgraph $B(v, H+1)$ and using $H+1$-step look-ahead, taking $z_\tau^u$ at node $u$ at time $\tau$ outputs the maximum (local) value-to-go on the subgraph,

$$f_\tau^{H+1}(z_\tau^u, z_\tau^{-u}; z_{\tau-1}, \Phi_\tau) + V_\tau^{H+1}(z_\tau^u, z_\tau^{-u}) \ge f_\tau^{H+1}(w_\tau^u, z_\tau^{-u}; z_{\tau-1}, \Phi_\tau) + V_\tau^{H+1}(w_\tau^u, z_\tau^{-u}).$$

After rearranging the terms in the above inequality,

$$f_\tau^u(w_\tau^u; z_\tau^{\Gamma(u)}, z_{\tau-1}^u, \Phi_\tau) - f_\tau^u(z_\tau^u; z_\tau^{\Gamma(u)}, z_{\tau-1}^u, \Phi_\tau) \le V_\tau^{H+1}(z_\tau^u, z_\tau^{-u}) - V_\tau^{H+1}(w_\tau^u, z_\tau^{-u}). \quad (2.12)$$

Similarly, by optimality of $w_\tau^u$ when restricting to the subgraph $B(v,H)$ and using $H$-step look-ahead,

$$f_\tau^H(w_\tau^u, w_\tau^{-u}; w_{\tau-1}, \Phi_\tau) + V_\tau^H(w_\tau^u, w_\tau^{-u}) \ge f_\tau^H(z_\tau^u, w_\tau^{-u}; w_{\tau-1}, \Phi_\tau) + V_\tau^H(z_\tau^u, w_\tau^{-u}).$$

Hence, we define the following positive random variable which is effectively the optimality gap

between switching from action $w_\tau^u$ to $z_\tau^u$ under policy $\pi(H)$,

$$\begin{aligned}\Delta_\tau^u &\coloneqq \left[f_\tau^H(w_\tau^u, w_\tau^{-u}; w_{\tau-1}, \Phi_\tau) + V_\tau^H(w_\tau^u, w_\tau^{-u})\right] - \left[f_\tau^H(z_\tau^u, w_\tau^{-u}; w_{\tau-1}, \Phi_\tau) + V_\tau^H(z_\tau^u, w_\tau^{-u})\right] \\ &= \left[f_\tau^u(w_\tau^u; w_\tau^{\Gamma(u)}, w_{\tau-1}^u, \Phi_\tau) - f_\tau^u(z_\tau^u, w_\tau^{\Gamma(u)}; w_{\tau-1}^u, \Phi_\tau)\right] + \left[V_\tau^H(w_\tau^u, w_\tau^{-u}) - V_\tau^H(z_\tau^u, w_\tau^{-u})\right]\end{aligned}$$

Let $A_\tau$ denote the event such that $w_\tau^u \neq z_\tau^u$ and $w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)}$. Then,

$$\begin{aligned}\Delta_\tau^u \mathbb{I}\{A_\tau\} &= \left(f_\tau^u(w_\tau^u; w_\tau^{\Gamma(u)}, w_{\tau-1}^u, \Phi_\tau) - f_\tau^u(z_\tau^u, w_\tau^{\Gamma(u)}; w_{\tau-1}^u, \Phi_\tau) + V_\tau^H(w_\tau^u, w_\tau^{-u}) - V_\tau^H(z_\tau^u, w_\tau^{-u})\right) \mathbb{I}\{A_\tau\} \\ &= \Bigg(f_\tau^u(w_\tau^u; z_\tau^{\Gamma(u)}, z_{\tau-1}^u, \Phi_\tau) - f_\tau^u(z_\tau^u; z_\tau^{\Gamma(u)}, z_{\tau-1}^u, \Phi_\tau) \\ &\quad + \Phi_{\tau-1,\tau}^u(w_{\tau-1}^u, w_\tau^u) - \Phi_{\tau-1,\tau}^u(z_{\tau-1}^u, w_\tau^u) - \Phi_{\tau-1,\tau}^u(w_{\tau-1}^u, z_\tau^u) + \Phi_{\tau-1,\tau}^u(z_{\tau-1}^u, z_\tau^u) \\ &\quad + V_\tau^H(w_\tau^u, w_\tau^{-u}) - V_\tau^H(z_\tau^u, w_\tau^{-u})\Bigg)\mathbb{I}\{A_\tau\} \\ &\leq 4c_{\text{time}}\mathbb{I}\{z_{\tau-1}^u \neq w_{\tau-1}^u\} + \Bigg(f_\tau^u(w_\tau^u; z_\tau^{\Gamma(u)}, z_{\tau-1}^u, \Phi_\tau) - f_\tau^u(z_\tau^u; z_\tau^{\Gamma(u)}, z_{\tau-1}^u, \Phi_\tau) \\ &\qquad + V_\tau^H(w_\tau^u, w_\tau^{-u}) - V_\tau^H(z_\tau^u, w_\tau^{-u})\Bigg)\mathbb{I}\{A_\tau\}.\end{aligned}$$

Then, by (2.12), we further have

$$\begin{aligned}\Delta_\tau^u \mathbb{I}\{A_\tau\} \leq & 4c_{\text{time}}\mathbb{I}\{z_{\tau-1}^u \neq w_{\tau-1}^u\} + \\ & \underbrace{\left(V_\tau^{H+1}(z_\tau^u, z_\tau^{-u}) - V_\tau^{H+1}(w_\tau^u, z_\tau^{-u}) + V_\tau^H(w_\tau^u, w_\tau^{-u}) - V_\tau^H(z_\tau^u, w_\tau^{-u})\right)\mathbb{I}\{A_\tau\}}_{(\natural)}\end{aligned} \tag{2.13}$$

Next, we expand out the expressions for the (local) value-to-go functions $V_\tau^H$, $V_\tau^{H+1}$ in ($\natural$).

Define the following functions over decision vectors at time $\tau + 1$:

$$g_\tau(x) = g_\tau(x; \Phi_{\tau+1}) \coloneqq f_\tau^{H+1}(x; z_\tau^u, z_\tau^{-u}, \Phi_{\tau+1}) + V_{\tau+1}^{H+1}(x),$$
$$h_\tau(x) = h_\tau(x; \Phi_{\tau+1})) \coloneqq f_\tau^{H}(x; w_\tau^u, w_\tau^{-u}, \Phi_{\tau+1}) + V_{\tau+1}^{H}(x),$$
$$\delta_\tau(x) = \delta_\tau(x; \Phi_{\tau+1})) \coloneqq \Phi_{\tau,\tau+1}^u(w_\tau^u, x^u) - \Phi_{\tau,\tau+1}^u(z_\tau^u, x^u).$$

where we omit their dependency on $\Phi_{\tau+1}$ to simplify the notations. Then, we have

$$\text{RV}_\tau^{H+1}(z_\tau^u, z_\tau^{-u}; \Phi_{\tau+1}) = \max_{x \in \mathcal{A}^{B(v,H+1)}} g_\tau(x),$$
$$\text{RV}_\tau^{H+1}(w_\tau^u, z_\tau^{-u}; \Phi_{\tau+1}) = \max_{x \in \mathcal{A}^{B(v,H+1)}} g_\tau(x) + \delta_\tau(x)$$

and

$$\text{RV}_\tau^{H}(w_\tau^u, w_\tau^{-u}; \Phi_{\tau+1}) = \max_{x \in \mathcal{A}^{B(v,H)}} h_\tau(x),$$
$$\text{RV}_\tau^{H}(z_\tau^u, w_\tau^{-u}; \Phi_{\tau+1}) = \max_{x \in \mathcal{A}^{B(v,H)}} h_\tau(x) - \delta_\tau(x).$$

We again similarly omit the dependency on $\Phi_{\tau+1}$ to simplify the notations and note that $z_{\tau+1}$ is an optimal solution for $\max_{x \in \mathcal{A}^{B(v,H+1)}} g_\tau(x)$ and $w_{\tau+1}$ is an optimal solution for $\max_{x \in \mathcal{A}^{B(v,H)}} h_\tau(x)$.

Hence,

$$
\begin{aligned}
(\natural) &= \mathop{\mathbb{E}}_{\Phi_{\tau+1}}[\mathrm{RV}_\tau^{H+1}(z_\tau^u, z_\tau^{-u}) - \mathrm{RV}_\tau^{H+1}(w_\tau^u, z_\tau^{-u}) + \mathrm{RV}_\tau^{H}(w_\tau^v, w_\tau^{-u}) - \mathrm{RV}_\tau^{H}(z_\tau^u, w_\tau^{-u})]\mathbb{I}\{A_\tau\} \\
&\le \mathop{\mathbb{E}}_{\Phi_{\tau+1}}\left[g_\tau(z_{\tau+1}) - (g_\tau(z_{\tau+1}) + \delta_\tau(z_{\tau+1})) + h_\tau(w_{\tau+1}) - (h_\tau(w_{\tau+1}) - \delta_\tau(w_{\tau+1}))\right]\mathbb{I}\{A_\tau\} \\
&= \mathop{\mathbb{E}}_{\Phi_{t+1}}\left[\delta_\tau(w_{\tau+1}) - \delta_\tau(z_{\tau+1})\right]\mathbb{I}\{A_\tau\} \\
&= \mathop{\mathbb{E}}_{\Phi_{t+1}}\left[\Phi^u_{\tau,\tau+1}(w^u_\tau, w^u_{\tau+1}) - \Phi^u_{\tau,\tau+1}(z^u_\tau, w^u_{\tau+1}) - \Phi^u_{\tau,\tau+1}(w^u_\tau, z^u_{\tau+1}) + \Phi^u_{\tau,\tau+1}(z^u_\tau, z^u_{\tau+1})\right]\mathbb{I}\{A_\tau\} \\
&\le 4c_{\text{time}} \mathop{\mathbb{E}}_{\Phi_{\tau+1}}[\mathbb{I}\{w^u_{\tau+1} \neq z^u_{\tau+1}\}]\mathbb{I}\{A_\tau\},
\end{aligned}
$$

where the last inequality is since when $z^u_{\tau+1} = w^u_{\tau+1}$, the four terms on the RHS cancel out. Hence,

$$
\Delta^u_\tau \mathbb{I}\{A_\tau\} \le 4c_{\text{time}}(\mathbb{I}\{z^u_{\tau-1} \neq w^u_{\tau-1}\} + \mathop{\mathbb{E}}_{\Phi_{\tau+1}}[\mathbb{I}\{w^u_{\tau+1} \neq z^u_{\tau+1}\}]\mathbb{I}\{A_\tau\}).
$$

Finally, we have

$$
\begin{aligned}
\mathbb{P}_\Phi(w^u_\tau \neq z^u_\tau, w^{\Gamma(u)}_\tau = z^{\Gamma(u)}_\tau) &\le \mathbb{P}(0 \le \Delta^u_\tau \mathbb{I}\{A_\tau\} \le 4c_{\text{time}}(\mathbb{I}\{z^u_{\tau-1} \neq w^u_{\tau-1}\} + \mathop{\mathbb{E}}_{\Phi_{\tau+1}}[\mathbb{I}\{w^u_{\tau+1} \neq z^u_{\tau+1}\}])) \\
&= \mathbb{P}(w^u_{\tau-1} \neq z^u_{\tau-1})\mathbb{P}(0 \le \Delta^u_\tau \mathbb{I}\{A_\tau\} \le 4c_{\text{time}}(1 + \mathbb{P}(w^u_{\tau+1} \neq z^u_{\tau+1} | w^u_{\tau-1} \neq z^u_{\tau-1}))|w_{\tau-1} \neq z_{\tau-1}) \\
&\quad + \mathbb{P}(w^u_{\tau-1} = z^u_{\tau-1})\mathbb{P}(0 \le \Delta^u_\tau \mathbb{I}\{A_\tau\} \le 4c_{\text{time}}\mathbb{P}(w^u_{\tau+1} \neq z^u_{\tau+1} | w^u_{\tau-1} = z^u_{\tau-1})|w_{\tau-1} = z_{\tau-1}) \\
&\le \mathbb{P}(w^u_{\tau-1} \neq z^u_{\tau-1}) \cdot g \cdot 4c_{\text{time}}(1 + \mathbb{P}(w^u_{\tau+1} \neq z^u_{\tau+1} | w^u_{\tau-1} \neq z^u_{\tau-1})) \\
&\quad + \mathbb{P}(w^u_{\tau-1} = z^u_{\tau-1}) \cdot g \cdot 4c_{\text{time}}(\mathbb{P}(w^u_{\tau+1} \neq z^u_{\tau+1} | w^u_{\tau-1} = z^u_{\tau-1})) \\
&= 4gc_{\text{time}}(\mathbb{P}(w^u_{\tau-1} \neq z^u_{\tau-1}) + \mathbb{P}(w^u_{\tau+1} \neq z^u_{\tau+1})) \\
&\le (\mathbb{P}(w^u_{\tau-1} \neq z^u_{\tau-1}) + \mathbb{P}(w^u_{\tau+1} \neq z^u_{\tau+1}))\rho
\end{aligned}
$$

where the second last inequality is based on the following observation: conditional on previous decisions, previous reward functions, current interactions and node reward functions at other nodes except $u$, $\Delta^u_\tau \mathbb{I}\{A_\tau\} \in [0, s]$ for any $s \ge 0$ if and only if $\Phi^v_\tau(w^u_\tau) - \Phi^v_\tau(z^u_\tau)$ is within some length $s$

interval. Moreover, the probability of the above event is upper bounded by $g$ multiplied by $s$ due to the third condition in Assumption 2.3.1: for any $a \neq a' \in \mathcal{A}$, $b_1 < b_2$,

$$\mathbb{P}(\Phi_\tau^v(a) - \Phi_\tau^v(a') \in [b_1, b_2) \mid \Phi_\tau^{\text{inter}}, \{\Phi_\tau^u\}_{u \neq v}) \leq g(b_2 - b_1).$$

### 2.8.2 Proof of Lemma 2.4.2

We define a new distance metric which is more suitable for the dynamic optimization problem we are interested in. Denote node $v \in V$ at time $t \in [\mathcal{T}]$ as the pair $(v, t)$, which we henceforth call a *ST node*. Define the *ST distance* between two ST nodes $(v_1, t_1)$ and $(v_2, t_2)$ as

$$\text{dist}^{\text{st}}((v_1, t_1), (v_2, t_2)) = \text{dist}(v_1, v_2) + |t_1 - t_2|.$$

In particular, if $t_1 = t_2$, then $\text{dist}^{\text{st}}((v_1, t_1), (v_2, t_2)) = \text{dist}(v_1, v_2)$. We also define another parameter:

$$\xi \coloneqq (d + 2)\rho. \tag{2.14}$$

Note that $\xi \leq \frac{1}{2}$ under our assumption that $\rho \coloneqq 4g(dc_{\text{edge}} + 2c_{\text{time}}) \leq \frac{1}{2(d+2)}$. Recall that $\Gamma(u)$ denotes the set of neighbors of $u$ in $G$.

Before proving Lemma 2.4.2, we present the following claim.

**Claim 1.** Under the same setting as in Lemma 2.4.2, for $t \leq \tau \leq t + H$ and $u \in B(v, H + 1)$, we have

$$\mathbb{P}_\Phi(w_\tau^u \neq z_\tau^u | w_\tau^{\Gamma(u)}, z_\tau^{\Gamma(u)}) \leq \rho.$$

*Proof of Claim 1.* Let $E_\tau$ denote the event $w_\tau^u \neq z_\tau^u$ given $w_\tau^{\Gamma(u)}, z_\tau^{\Gamma(u)}$. Then, we define the following positive random variable as in the proof of Lemma 2.4.1 which is effectively the optimality gap between switching from action $w_\tau^u$ to $z_\tau^u$ under policy $\pi(H)$,

$$\Delta_\tau^u \mathbb{I}\{E_\tau\} \coloneqq \left[f_\tau^H(w_\tau^u, w_\tau^{-u}; w_{\tau-1}, \Phi_\tau) + V_\tau^H(w_\tau^u, w_\tau^{-u})\right] - \left[f_\tau^H(z_\tau^u, w_\tau^{-u}; w_{\tau-1}, \Phi_\tau) + V_\tau^H(z_\tau^u, w_\tau^{-u})\right]$$
$$\leq \Phi_\tau^u(w_\tau^u) - \Phi_\tau^u(z_\tau^u) + d \cdot 2c_{\text{edge}} + 2 \cdot 2c_{\text{time}},$$

where the last inequality is because changing node action at $u$ at time $\tau$ affects at most $d$ spatial edges and 2 temporal edges.

Since $\Delta_\tau^u \mathbb{I}\{E_\tau\} \geq 0$, we have the following bound under $E_\tau$,

$$\Phi_\tau^u(w_\tau^u) - \Phi_\tau^u(z_\tau^u) \geq -(d \cdot 2c_{\text{edge}} + 2 \cdot 2c_{\text{time}}).$$

Moreover, since $z_\tau^u$ is optimal under $\pi(H+1)$,

$$0 \leq \left[f_\tau^{H+1}(z_\tau^u, z_\tau^{-u}; z_{\tau-1}, \Phi_\tau) + V_\tau^{H+1}(z_\tau^u, z_\tau^{-u})\right] - \left[f_\tau^{H+1}(w_\tau^u, z_\tau^{-u}; z_{\tau-1}, \Phi_\tau) + V_\tau^{H+1}(w_\tau^u, z_\tau^{-u})\right]$$
$$\leq \Phi_\tau^u(z_\tau^u) - \Phi_\tau^u(w_\tau^u) + d \cdot 2c_{\text{edge}} + 2 \cdot 2c_{\text{time}},$$

which leads to

$$\Phi_\tau^u(w_\tau^u) - \Phi_\tau^u(z_\tau^u) \leq (d \cdot 2c_{\text{edge}} + 2 \cdot 2c_{\text{time}}).$$

Combining these two bounds above, we have

$$\mathbb{P}(w_\tau^u \neq z_\tau^u | w_\tau^{\Gamma(u)}, z_\tau^{\Gamma(u)}) \leq \mathbb{P}(-(2dc_{\text{edge}} + 4c_{\text{time}}) \leq \Phi_\tau^u(w_\tau^u) - \Phi_\tau^u(z_\tau^u) \leq$$
$$(2dc_{\text{edge}} + 4c_{\text{time}}) | w_\tau^{\Gamma(u)}, z_\tau^{\Gamma(u)})$$
$$\leq g \cdot 2(2dc_{\text{edge}} + 4c_{\text{time}}) = \rho.$$

Q.E.D.

*Proof of Lemma 2.4.2.* We prove the lemma by induction on the ST distance. By Claim 1 above, for $0 \leq \tau < H$ and $u \in B(v, H+1)$,

$$\mathbb{P}_\Phi(w_\tau^u \neq z_\tau^u) \leq \rho.$$

This serves as the base case for proof of Lemma 2.4.2: when $(u, \tau)$ satisfies $\text{dist}^{\text{st}}((v,t),(u,\tau)) \geq H+1$, Lemma 2.4.2 holds. Suppose that for all $k' > k$ for some $0 \leq k \leq H$, we have that if a node $(u,\tau)$ satisfies $\text{dist}^{\text{st}}((v,t),(u,\tau)) \leq k'$, then $\mathbb{P}_\Phi(w_\tau^u \neq z_\tau^u) \leq \xi^{H+1-k'}\rho$.

For the inductive step, we consider nodes $(u,\tau)$ with $\text{dist}^{\text{st}}((v,t),(u,\tau)) \leq k$ for $0 \leq k \leq H$. For the following, to simply the notations, we write $\mathbb{P}_\Phi$ as $\mathbb{P}$.

$$\begin{aligned}
\mathbb{P}(w_\tau^u \neq z_\tau^u) &= \mathbb{P}(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} \neq z_\tau^{\Gamma(u)}) + \mathbb{P}(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)}) \\
&= \mathbb{P}(w_\tau^u \neq z_\tau^u \mid w_\tau^{\Gamma(u)} \neq z_\tau^{\Gamma(u)})\mathbb{P}(w_\tau^{\Gamma(u)} \neq z_\tau^{\Gamma(u)}) + \mathbb{P}(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)}) \\
&\leq \rho(d \cdot \xi^{H-k}\rho) + \mathbb{P}(w_\tau^u \neq z_\tau^u, w_\tau^{\Gamma(u)} = z_\tau^{\Gamma(u)}) \\
&\leq \rho(d \cdot \xi^{H-k}\rho) + \mathbb{P}(w_{\tau-1}^u \neq z_{\tau-1}^u)\rho + \mathbb{P}(w_{\tau+1}^u \neq z_{\tau+1}^u)\rho \\
&\leq \rho(d \cdot \xi^{H-k}\rho) + 2(\xi^{H-k}\rho)\rho \\
&= (d\rho + 2\rho)\xi^{H-k}\rho \\
&\leq \xi^{H+1-k}\rho
\end{aligned}$$

where the first inequality is by induction hypothesis since the spatial neighbors of $u$ have ST distance to $(v,t)$ at most $k+1$ as well as Claim 1; the second inequality is by Lemma 2.4.1; the third inequality is again by the induction hypothesis. Hence we complete the induction step. Q.E.D.

### 2.8.3 Proof of Proposition 2.4.2.

Suppose $\text{Alg}_t^v \neq \pi_t^v(L)$. By optimality,

$$\phi_t^v(\pi_t^v(L)) - \phi_t^v(\text{Alg}_t^v) + \Delta_t^{-v} \coloneqq \max_{x_t^{B(v,L)}: x_t^v = \pi_t^v(L)} \left(f_t^L(x_t) + V_t^L(x_t)\right) - \max_{x_t^{B(v,L)}: x_t^v = \text{Alg}_t^v} \left(f_t^L(x_t) + V_t^L(x_t)\right) \geq 0.$$

Similarly,

$$\phi_t^v(\text{Alg}_t^v) - \phi_t^v(\pi_t^v(L)) + \Delta^{-v,n} \coloneqq \max_{x_t^{B(v,L)}: x_t^v = \text{Alg}_t^v} \left(f_t^L(x_t) + \widehat{V}_t^{L,n}(x_t)\right) - \max_{x_t^{B(v,L)}: x_t^v = \pi_t^v(L)} \left(f_t^L(x_t) + \widehat{V}_t^{L,n}(x_t)\right) \geq 0.$$

Therefore,

$$-\Delta_t^{-v} \leq \phi_t^v(\pi_t^v(L)) - \phi_t^v(\text{Alg}_t^v) \leq \Delta^{-v,n}.$$

By the third condition in the Assumption 2.3.1, we have that

$$\begin{aligned}\mathbb{P}\left(\text{Alg}_t^v(L) \neq \pi_t^v(L)\right) &\leq \mathbb{P}\left(-\Delta_t^{-v} \leq \Phi_t^v(\pi_t^v(L)) - \Phi_t^v(\text{Alg}_t^v) \leq \Delta^{-v,n}\right) \\ &\leq g\,\mathbb{E}\left[\Delta^{-v,n} + \Delta_t^{-v}\right].\end{aligned}$$

By definitions of $\Delta_t^{-v}$ and $\Delta^{-v,n}$ above, we have

$$\Delta^{-v,n} + \Delta_t^{-v} \leq \max_{x_t^{B(v,L)}: x_t^v = \pi_t^v(L)} \left(V_t^L(x_t) - \widehat{V}_t^{L,n}(x_t)\right) + \max_{x_t^{B(v,L)}: x_t^v = \text{Alg}_t^v} \left(\widehat{V}_t^{L,n}(x_t) - V_t^L(x_t)\right)$$

Therefore, it suffices to show that

$$\mathbb{E}\left[\max_{x_t^{B(v,L)}} \left(V_t^L(x_t) - \widehat{V}_t^{L,n}(x_t)\right)\right] \leq \frac{\epsilon}{4gC} \text{ and } \mathbb{E}\left[\max_{x_t^{B(v,L)}} \left(\widehat{V}_t^{L,n}(x_t) - V_t^L(x_t)\right)\right] \leq \frac{\epsilon}{4gC}.$$

By definition, given any $x_{t+L} \in \mathcal{A}^{B(v,L)}$, $\widehat{V}_{t+L}^{L,n}(x_{t+L}) = V_{t+L}^{L}(x_{t+L}) = 0$. Hence,

$$\mathbb{E}[\max x_{t+L}(\widehat{V}_{t+L}^{L,n}(x_{t+L}) - V_{t+L}^{L}(x_{t+L}))] = 0.$$

Next, for $t \leq \tau \leq t+L-1$, we derive a recursive relation between $\mathbb{E}[\max x_\tau(\widehat{V}_\tau^{L,n}(x_\tau) - V_\tau^L(x_\tau))]$ and $\mathbb{E}[\max x_{\tau+1}(\widehat{V}_{\tau+1}^{L}(x_{\tau+1}) - V_{\tau+1}^{L}(x_{\tau+1}))]$. Given any $x_\tau \in \mathcal{A}^{B(v,L)}$

$$\begin{aligned}
\widehat{V}_\tau^{L,n}(x_\tau) - V_\tau^L(x_\tau) &= \frac{1}{n}\sum_{s=1}^{n} \max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \phi_{\tau+1}^{(s)}) + \widehat{V}_{\tau+1}^{L,n}(x_{\tau+1})) - \mathbb{E}[\max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \Phi_{\tau+1}) + V_{\tau+1}^L(x_{\tau+1}))] \\
&= \frac{1}{n}\sum_{s=1}^{n} \max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \phi_{\tau+1}^{(s)}) + \widehat{V}_{\tau+1}^{L,n}(x_{\tau+1})) - \frac{1}{n}\sum_{s=1}^{n} \max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \phi_{\tau+1}^{(s)}) + V_{\tau+1}^L(x_{\tau+1})) \\
&+ \frac{1}{n}\sum_{s=1}^{n} \max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \phi_{\tau+1}^{(s)}) + V_{\tau+1}^L(x_{\tau+1})) - \mathbb{E}[\max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \Phi_{\tau+1}) + V_{\tau+1}^L(x_{\tau+1}))] \\
&\leq \max_{x_{\tau+1}}(\widehat{V}_{\tau+1}^{L,n}(x_{\tau+1}) - V_{\tau+1}^L(x_{\tau+1})) + \frac{1}{n}\sum_{s=1}^{n} \max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \phi_{\tau+1}^{(s)}) + V_{\tau+1}^L(x_{\tau+1})) \\
&- \mathbb{E}[\max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \Phi_{\tau+1}) + V_{\tau+1}^L(x_{\tau+1}))]
\end{aligned}$$

We now bound the expectation of latter two terms on the right-hand side of the above inequality. Let $Y^{(s)} \coloneqq \max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \phi_{\tau+1}^{(s)}) + V_{\tau+1}^L(x_{\tau+1}))$. Then, $\{Y^{(s)}\}_{1\leq s\leq n}$ are independent random variables with expectation $\mu \coloneqq \mathbb{E}[\max_{x_{\tau+1}}(f_{\tau+1}^L(x_{\tau+1}; x_\tau, \Phi_{\tau+1}) + V_{\tau+1}^L(x_{\tau+1}))]$. Note that

$$\mathbb{E}[\frac{1}{n}\sum_{s=1}^{n} Y^{(s)} - \mu] = \frac{1}{n}\,\mathbb{E}[\sum_{s=1}^{n} Y^{(s)} - n\mu] \leq \frac{1}{n}\,\mathbb{E}\left|\sum_{s=1}^{n} Y^{(s)} - n\mu\right|$$

Let $\widetilde{m}$ denote the number of nodes in $B(v,L)$. $\widetilde{m} = 1 + d + \cdots + d^L \leq \frac{d^{L+1}}{d-1}$. Since $\widetilde{m}d/2$ is the maximum number of edges in $B(v,L)$ by the handshaking lemma, we have that

$$Y^{(s)} = \mathrm{RV}_\tau^L(x_\tau; \phi_{\tau+1}(s)) \geq -L\widetilde{m}C_{\text{node}} - L\widetilde{m}\frac{d}{2}c_{\text{edge}} - L\widetilde{m}c_{\text{time}} =: \mathrm{lb},$$

where this lower bound is achieved when all nodes receive the worst possible individual reward $-C_{\text{node}}$, all temporal interactions are $-c_{\text{time}}$, and all edge interactions are $-c_{\text{edge}}$. Similarly, we ob-

tain the upper bound when all nodes receive the best possible individual reward $C_{\text{node}}$, all temporal interactions are $c_{\text{time}}$ and all edge interactions are $c_{\text{edge}}$,

$$Y^{(s)} = \text{RV}_\tau^L(x_\tau; \phi_{\tau+1}(s)) \leq L\widetilde{m}C_{\text{node}} + L\widetilde{m}\frac{d}{2}c_{\text{edge}} + L\widetilde{m}c_{\text{time}} =: \text{ub}.$$

Since $\text{lb} \leq Y^{(s)} \leq \text{ub}$,

$$\begin{aligned}
\frac{1}{n}\,\mathbb{E}\Big|\sum_{s=1}^{n} Y^{(s)} - n\mu\Big| &= \frac{1}{n}\int_{t\geq 0} \mathbb{P}(\Big|\sum_{s=1}^{n} Y^{(s)} - n\mu\Big| \geq t)dt \\
&\leq \frac{2}{n}\int_{t\geq 0} e^{-\frac{2t^2}{n(\text{ub}-\text{lb})^2}}dt \\
&= \frac{1}{\sqrt{n}}(\text{ub}-\text{lb})\int_{x\geq 0} e^{-\frac{x^2}{2}}dx \\
&= \frac{\sqrt{2\pi}}{2\sqrt{n}}(\text{ub}-\text{lb})
\end{aligned}$$

where the first equality is by the property of expectation of non-negative random variables, the first inequality is by Hoeffding's inequality, the second equality is by change of variables (i.e., we define a new variable $x \coloneqq \frac{2t}{\sqrt{n}(\text{ub}-\text{lb})}$), and the last equality is since $\int_{x\geq 0} e^{-\frac{x^2}{2}}dx = \frac{\sqrt{2\pi}}{2}$.

Taking expectation of $\max_{x_\tau}(\widehat{V}_\tau^{L,n}(x_\tau) - V_\tau^L(x_\tau))$, we have

$$\begin{aligned}
\mathbb{E}[\max_{x_\tau}(\widehat{V}_\tau^{L,n}(x_\tau) - V_\tau^L(x_\tau))] &\leq \mathbb{E}[\max_{x_{\tau+1}}(\widehat{V}_{\tau+1}^{L}(x_{\tau+1}) - V_{\tau+1}^L(x_{\tau+1}))] + \frac{\sqrt{2\pi}}{2\sqrt{n}}(\text{ub}-\text{lb}) \\
&\leq \mathbb{E}[\max_{x_{\tau+1}}(\widehat{V}_{\tau+1}^{L}(x_{\tau+1}) - V_{\tau+1}^L(x_{\tau+1}))] + \frac{\sqrt{2\pi}}{2\sqrt{n}}L\widetilde{m}C
\end{aligned} \tag{2.15}$$

where the last inequality is due to $C \coloneqq 2C_{\text{node}} + 2dc_{\text{edge}} + 4c_{\text{time}} \geq 2C_{\text{node}} + dc_{\text{edge}} + 2c_{\text{time}}$.

Applying Equation (2.15) $L$ times for $t \leq \tau \leq t+L-1$, we have

$$\mathbb{E}[\max_{x_t}(\widehat{V}_t^L(x_t) - V_t^L(x_t))] \leq \frac{\sqrt{2\pi}}{2\sqrt{n}}L^2\widetilde{m}C \leq \frac{\epsilon}{4gC},$$

for $n \geq N(\epsilon, d, g, C) = \frac{8\pi g^2 C^4 L^4 \widetilde{m}^2}{\epsilon^2}$. Under the condition in the Theorem 2.3.1, let $L = \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$.

Then,

$$\begin{aligned} N(\epsilon, d, g, C) &= \frac{8\pi g^2 C^4}{\epsilon^2}(\log_2 \frac{4C}{\epsilon})^4 (\frac{4C}{\epsilon})^{2\log_2 d}(\frac{d}{d-1})^2 \\ &= O((\frac{4C}{\epsilon})^{2\log_2 d} g^2 C^4) \end{aligned} \tag{2.16}$$

where $O(\cdot)$ notation omits logarithmic factors. Similarly, for such $n \geq N(\epsilon, d, g, C)$, we have

$$\mathbb{E}[\max_{x_t}(V_t^L(x_t) - \widehat{V}_t^L(x_t))] \leq \frac{\sqrt{2\pi}}{2\sqrt{n}} L^2 \widetilde{m} C \leq \frac{\epsilon}{4gC}.$$

Altogether, we have

$$\mathbb{E}\left[\max_{x_t^{B(v,L)}} \left(V_t^L(x_t) - \widehat{V}_t^{L,n}(x_t)\right)\right] \leq \frac{\epsilon}{4gC} \text{ and } \mathbb{E}\left[\max_{x_t^{B(v,L)}} \left(\widehat{V}_t^{L,n}(x_t) - V_t^L(x_t)\right)\right] \leq \frac{\epsilon}{4gC}$$

as desired.

### 2.9 Bound on the computational requirement.

To establish Proposition 2.5.1, we prove the following lemma on computation requirement for any $L$-local ($L \geq 1$) algorithm. Then, the computation needed in Proposition 2.5.1 is by letting $L = \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$ and $n = N(\epsilon, d, g, C)$ defined in Equation (2.16).

**Lemma 2.9.1.** *The computation requirement for* $\text{Alg}_t^v$ *for* $v \in V$ *and* $t \in [\mathcal{T}]$ *under Algorithm 1 is* $LK^2 n$ *where* $n$ *is the sample size and* $K = |\mathcal{A}|^{d^L}$ *is an upper bound on the number of decision vectors to enumerate over for the optimization problem in* (2.7)*.*

Proof of Lemma 2.9.1. We show this by induction. Let $a_\tau$ denote the amount of computation needed to compute $\widehat{V}_\tau^{L,n}(\cdot)$. Since $\widehat{V}_{t+L}^{L,n}(\cdot) = 0$, $a_{t+L} = 0$. Suppose now we obtain $\widehat{V}_\tau^{L,n}(\cdot)$ function with computational effort $a_\tau$. Given a decision vector $x_{\tau-1}$ and a realization $\phi_\tau$, the optimal $x_\tau$ can be solved by enumerating all possible decision vectors (the number of nodes in $B(v, L)$ without constraints are $\sum_{i=0}^{L-1} d^i \leq d^L$), whose cardinality is at most $K$. Under the assumption that $\{\Phi_t\}_t$

are independent, we can use the same estimation for $\widehat{V}_{\tau}^{L,n}(\cdot)$ for different realizations of $\Phi_{\tau-1}$. This implies:

$$a_{\tau-1} = K \cdot n \cdot K + a_{\tau},$$

where the first $K$ is the number of possible decision vectors $x_{\tau-1}$, $n$ is the number of samples, and the second $K$ is the computation needed for enumeration. Hence, we have $a_t = LK^2n = L|\mathcal{A}|^{2d^L}n$.

Proof of Proposition 2.5.1 Let $L = \lfloor \log_2 \frac{4C}{\epsilon} \rfloor$. Then, we have

$$|\mathcal{A}|^{2d^L} \leq \mathrm{e}^{2\ln|\mathcal{A}|\cdot(\frac{4C}{\epsilon})^{\log_2 d}}.$$

Moreover, we let the sample size $n = N(\epsilon, d, g, C) = O((\frac{4C}{\epsilon})^{2\log_2 d} g^2 C^4)$ defined in Equation (2.16). With $d, g, C$ and $|\mathcal{A}|$ as constants, by Lemma 2.9.1, the computation needed for Algorithm 1 is upper bounded by

$$L|\mathcal{A}|^{2d^L}n = O(\mathrm{e}^{2\ln|\mathcal{A}|\cdot(\frac{4C}{\epsilon})^{\log_2 d}}(\frac{1}{\epsilon})^{2\log_2 d}) = O(\mathrm{e}^{\mathrm{poly}(\frac{1}{\epsilon})}).$$

## 2.10 Interactions must be small to have correlation decay

In this section, we construct a sequence of static (i.e., single period) decision networks indexed by graph degree $d$ with $c_{\text{edge}} = \Theta(1/d)$ such that there is no near-optimal local algorithm for these networks. The decision networks we construct satisfy all parts of Assumption 2.3.1 *except* the small-interaction requirement $4g(dc_{\text{edge}} + 2c_{\text{time}}) \leq \frac{1}{2(d+2)}$. Thus, our construction justifies the need for the upper bound on the strength of the interactions in Assumption 2.3.1. Admittedly, there is some gap between our assumption $c_{\text{edge}} \leq \Theta(1/d^2)$, and the scale $c_{\text{edge}} = \Theta(1/d)$ at which we show here that long-range correlations arise. In comparison, previous work (Gamarnik, Goldberg, and Weber, 2014) also assumed $c_{\text{edge}} \leq \Theta(1/d^2)$ to obtain correlation decay in a static random decision network. In our dynamic setting, we have the same scaling to ensure no long-range correlation.

**Definition 2.10.1.** *A $d$-regular graph $G = (V, E)$ is an* $\gamma$-edge expander *for $\gamma \in (0, 1)$ if for any $S \subseteq V$ such that $|S| \leq |V|/2$, the number of edges between $S$ and $V \backslash S$ (the "cut size") is at least $|S|d\gamma$, i.e.,*

$$\text{cut}(S) \coloneqq |\{(i, j) \in E : i \in S, j \in V \backslash S\}| \geq |S|d\gamma \,.$$

**Construction.** Fix degree $d \geq 35$. It is well known that there exists $m_1 < \infty$, such that for any even integer $m$ with $m > m_1$, there is a $d$-regular graph with $m$ nodes that is a $\frac{1}{3}$-edge expander (Friedman, 2008; Vadhan et al., 2012). In fact, a random $d$-regular graph has this property asymptotically almost surely (a.a.s.).[1] That is, let $G$ be uniformly drawn from random $d$-regular graphs with $m$ nodes where $m \geq m_1$, then $G$ is a $\frac{1}{3}$-edge expander almost surely. We define a static random decision network $(G, \Phi)$ with action set $\{0, 1\}$ as follows:

- Node rewards: The node rewards $\Phi^v(1)$ are i.i.d. from Uniform$[-1, 1]$; and $\Phi^v(0) = 0$.
- Edge rewards: The edge rewards are "ferromagnetic":

$$\Phi^{u,v}(x^u, x^v) \coloneqq \begin{cases} c_{\text{edge}} & \text{if } x^u = x^v \\ 0 & \text{otherwise} \,, \end{cases}$$

  where $c_{\text{edge}} \coloneqq 6/d$.

Since the constructed decision network is static, there are no temporal interactions in our construction and hence we omit $x_0$.

The following claim shows that there does not exist near-optimal local algorithms when the small interaction condition in Assumption 2.3.1 does not hold.

**Claim 2.** *For the decision network $(G, \Phi)$ uniformly drawn from $d-$regular random graph with $m$ nodes, the optimal action vector is either all $1$s or all $0$s a.s. Each of these possibilities arises with probability $1/2$. In particular, the optimal solution has long-range correlations. In particular,*

[1] Random $d$-regular graphs are "almost Ramanujam", i.e., the absolute value of the second largest eigenvalue of their adjacency matrix is bounded above by $2\sqrt{d-1} + \epsilon$ a.a.s. as proved in (Friedman, 2008). The claimed edge expansion property then follows, e.g., using (Vadhan et al., 2012, Theorem 4.14).

*any $L$-local algorithm which treats the possible node actions $1$ and $0$ symmetrically achieves an expected payoff at least $m/3$ below the optimal.*

*Proof of Claim 2.* Consider any action vector $x$ such that the majority of actions is $1$. We show that the payoff of $x$ is less than the payoff of all 1s. Let $S$ be the set of nodes where $x$ takes action $0$. Since $G$ is a $\frac{1}{3}$-edge expander a.s., $\text{cut}(S) \geq |S|d/3$. It follows that the total edge rewards under $x$ are at least $c_{\text{edge}}|S|d/3 = 2|S|$ smaller than that under all 1s. On the other hand, the difference between the total node rewards under $x$ and that under all 1s is $\sum_{v\in S}\Phi^v(1) \geq -|S|$ since $\Phi^v(1) \in [-1,1]$, i.e., the total rewards under $x$ is at least $2|S| - |S| = |S|$ smaller than the total rewards under all 1s. Similarly, one can show that for any action vector $x$ such that the majority of actions is $0$, the total reward under $x$ is at least $m - |S|$ smaller than the total reward under all 0s. It follows that the optimal solution is either all 1s or 0s. Moreover, the optimal solution is all 1s if $\sum_{v\in V}\phi^v(1) \geq 0$ and all 0s otherwise. Since the distribution of i.i.d Uniform distribution $[-1,1]$ is symmetric, each of these above possibilities arises with probability $\frac{1}{2}$.

Now consider any given $L$ and any $L$-local algorithm which treats $1$ and $0$ symmetrically. By symmetry, each node decision is a priori equally likely to be $1$ or $0$. By symmetry, each node decision is a priori equally likely to be $1$ or $0$. It follows that in a large network, about half the decisions will be $1$ and the other half will be $0$ under the $L$-local algorithm. Formally speaking, the expected number of 1s is $m/2$, and the variance in the total number of 1s is $\text{Var}[\sum_{v\in V}\mathbb{I}\{x^v = 1\}] = \sum_{v\in V}\text{Var}[\mathbb{I}\{x^v = 1\}] = \sum_{v\in V}(\frac{1}{2})^2 = \frac{1}{4}m$.

Hence for any $m > 250$, we know by Chebyshev's inequality that with probability at least $0.9$, the number of 1s will be in the range $|S| \in (0.4m, 0.6m)$, i.e., the payoff will be at least $0.4m$ below the optimal (see the previous paragraph) Combining, the local algorithm suffers an expected payoff loss at least $0.9 \times 0.4m \geq m/3$.

Q.E.D.

## 2.11 Description of the Experiment Setup

In the following, we explain in detail the simulation environment of our experiment. There are $5$ dynamic decision networks parameterized by interaction strength $c$ for both the spatial and temporal dimensions, with $c = 0.1, 0.2, 0.3, 0.4$, and $0.5$. These decision networks share all other components, which we list below.

- *Graph $G$*: Using the *NetworkX* package in Python, we randomly generate[2] a $3$-regular graph with $500$ vertices.
- *Time horizon $\mathcal{T}$:* To simplify the simulation and reduce the computational effort, we set the time horizon to be $2$ for all decision networks.
- *Interaction function*: Both the spatial interaction and the temporal interaction are *ferromagnetic*, meaning that agreeing actions incur a bonus of $c$, whereas disagreeing actions result in no reward.
- *Action set $\mathcal{A}$*: We assume a binary action set – that is, for all $v \in V$ and $t \in [\mathcal{T}]$, $x_t^v \in \{0, 1\}$, where action zero is viewed as the default action, meaning that $\Phi_t^v(0) = 0$.
- *Node reward*: The random node rewards, for both time periods, when taking action $1$ are assumed to i.i.d. and follow the uniform distribution on $[-1, 1]$.

We sample $n_1 = 10$ instances for each decision network, where these instances differ in terms of the realized node rewards at the first time period. Having multiple realizations allows us to compute the confidence intervals for the performance of our algorithm. We denote by $\{\phi_1^{v,(i)}\}_{v \in V}$ the realized first-period node rewards (when taking action $1$) in the $i$-th instance, where each $\phi_1^{v,(i)}$ is sampled according to the node reward distribution, i.e., uniformly from $[-1, 1]$. Note that the realized node rewards $\{\phi_1^{v,(i)}\}_{v \in V}$ for each $i \in [n_1]$ are shared by all the $i$-th instances of all decision networks.

[2]statistics source: `https://networkx.org/documentation/stable/reference/generated/networkx.generators.random_graphs.random_regular_graph.html`

To remove the loss in rewards due to sampling, we control the variability in the second-period node rewards. That is, we pre-generate two independent sets of samples of node rewards for the second time period. The first set contains $n_{2,est} = 100$ samples, which are used to compute solutions at the first time period; and the second set contains $n_{2,eval} = 30$ samples, which are used to estimate the total payoff under the solutions computed using the first set of samples. We denote the node rewards when taking action 1 by $\{\phi_{2,est}^{v,(i)}\}_{v\in V}$ for the $i$-th sample in the first set and by $\{\phi_{2,eval}^{v,(i)}\}_{v\in V}$ for the $i$-th sample in the second set.

For each instance, we compute several solutions, with one obtained by solving the global optimization problem, and the others obtained by our local algorithms with different locality parameters. To solve the network optimization problem, either globally or locally, we write down the problem as a MIP. The decision variables of the MIP are:

- node actions for the first time period: $\{x_1^v\}_{v\in V}$;
- disagreement indicator of neighboring nodes for $t = 1$: $\{y_1^e\}_{e\in E}$;
- node actions for the second time period for each sample $j$: $\{x_2^{v,(j)}\}_{v\in V, j\in[n_{2,est}]}$;
- disagreement indicator of neighboring nodes for $t = 2$ for each sample $j$: $\{y_2^{e,(j)}\}_{e\in E, j\in[n_{2,est}]}$;
- temporal disagreement indicator for each node for each sample $j$: $\{y^{v,(j)}\}_{v\in V, j\in[n_{2,est}]}$.

And the formulation of our MIP is given below.

$$
\begin{aligned}
\max \quad & \sum_{v \in V} \phi_1^{v,(i)} \cdot x_1^v + \sum_{e=(u,v) \in E} c \cdot (1 - y_1^e) + \\
& \frac{1}{n_{2,est}} \sum_{j \in [n_{2,est}]} \left[ \sum_{v \in V} \phi_{2,est}^{v,(j)} \cdot x_2^{v,(j)} + \sum_{e \in E} c \cdot (1 - y_2^{e,(j)}) + \sum_{v \in V} c \cdot (1 - y^{v,(j)}) \right] \\
\text{s.t.} \quad & y_1^e \geq x_1^u - x_1^v \qquad \forall\, e = (u,v) \in E \\
& y_1^e \geq x_1^v - x_1^u \qquad \forall\, e = (u,v) \in E \\
& y_2^{e,(j)} \geq x_2^{u,(j)} - x_2^{v,(j)} \qquad \forall\, e = (u,v) \in E,\ j \in [n_{2,est}] \\
& y_2^{e,(j)} \geq x_2^{v,(j)} - x_2^{u,(j)} \qquad \forall\, e = (u,v) \in E,\ j \in [n_{2,est}] \\
& y^{v,(j)} \geq x_1^v - x_2^{v,(j)} \qquad \forall\, v \in V,\ j \in [n_{2,est}] \\
& y^{v,(j)} \geq x_2^{v,(j)} - x_1^v \qquad \forall\, v \in V,\ j \in [n_{2,est}] \\
& x_1^v, y_1^e, x_2^{v,(j)}, y_2^{e,(j)}, y^{v,(j)} \in \{0,1\} \qquad \forall\, v \in V,\ e \in E,\ j \in [n_{2,est}]
\end{aligned}
$$

Note that $V$ and $E$ are either nodes and edges of the entire graph when solving for the global optimal solution, or nodes and edges of a local graph when solving for the solution using our local algorithm. Although the MIP is given for obtaining a first time period solution, a similar MIP can be used to estimate the payoff of a given first time period solution, where we take $\{x_1^v\}_{v \in V}$ as given and replace rewards $\{\phi_{2,est}^{v,(j)}\}$ with $\{\phi_{2,eval}^{v,(j)}\}$.

# Chapter 3: Decentralized Online Convex Optimization in Networked Systems

## 3.1 Introduction

A wide variety of multi-agent systems can be modeled as optimization tasks in which individual agents must select actions based on local information with the goal of cooperatively learning to minimize a global objective in an uncertain, time-varying environment. This general setting emerges in applications such as formation control (Chen and Wang, 2005; Oh, Park, and Ahn, 2015), power systems control (Molzahn et al., 2017; Shi et al., 2021), and multiproduct price optimization (Caro and Gallien, 2012; Candogan, Bimpikis, and Ozdaglar, 2012). In all these cases, it is key that the algorithms used by agents use only local information due to the computational burden created by the size of the systems, the information constraints in the systems, and the need for fast and/or interpretable decisions.

At this point, there is a mature literature focused on decentralized optimization, e.g. (Bertsekas and Tsitsiklis, 1989a; Boyd, Parikh, and Chu, 2011; Shi et al., 2015; Nedić, Olshevsky, and Rabbat, 2018), see (Xin et al., 2020) for a survey; however, the design of learning policies for uncertain, time-varying environments requires decentralized *online* optimization. The literature studying decentralized online optimization is still nascent (see the related work section for a discussion of recent papers, e.g. (Li, Yi, and Xie, 2021b; Yuan, Proutiere, and Shi, 2021; Yi et al., 2020)) and many challenging open questions remain.

Three issues of particular importance for real-world applications are the following.

*First,* temporal coupling in actions is often of first-order importance to applications. For example, startup costs, ramping costs, and switching costs are prominent in settings such as power systems and cloud computing, and lead to penalties for changing actions dramatically over time.

The design of online algorithms to address such temporal interaction costs has received significant attention in the single-agent case recently, e.g, smoothed online optimization (Goel et al., 2019; Lin, Goel, and Wierman, 2020), convex body chasing (Argue et al., 2020; Sellke, 2020), online optimization with memory (Agarwal et al., 2019; Shi et al., 2020), and dynamic pricing (Besbes and Lobel, 2015; Chen and Farias, 2018).

*Second,* spatial interaction costs are of broad importance in practical applications. Such costs arise because of the need for actions of nearby agents to be aligned with one another, and are prominent in settings such as economic team theory (Marschak, 1955; Marschak and Radner, 1972), combinatorial optimization over graphs (Hochba, 1997; Gamarnik and Goldberg, 2010), and statistical inference (Wainwright and Jordan, 2008). An example is (dynamic) multiproduct pricing, where the price of a product can impact the demand of other related products (Song and Chintagunta, 2006).

*Third,* leveraging predictions of future costs has long been recognized as a promising way to improve the performance of online agents (Morari and Lee, 1999; Lin et al., 2012; Badiei, Li, and Wierman, 2015; Chen et al., 2016; Shi, Lin, and Jiao, 2019; Li, Qu, and Li, 2020). As learning tools become more prominent, the role of predictions is growing. By collecting data from repeated trials, data-driven learning tools make it possible to provide accurate predictions for near future costs. For example, in multiproduct pricing, good demand forecasts can be constructed up to a certain time horizon and are invaluable in setting prices (Caro and Gallien, 2012).

In addition to the three issues above, we would like to highlight that existing results for decentralized online optimization focus on designing algorithms with low (static) regret (Hosseini, Chapman, and Mesbahi, 2016; Li, Yi, and Xie, 2021b) , i.e., algorithms that (nearly) match the performance of the best static action in hindsight. In a time-varying environment, it is desirable to instead obtain stronger bounds, such as those on the dynamic regret or competitive ratio, which compare to the dynamic optimal actions instead of the best static action in hindsight, e.g., see results in the centralized setting such as (Lin, Goel, and Wierman, 2020; Li, Qu, and Li, 2020; Shi et al., 2020).

This paper aims to address decentralized online optimization with the three features described above. In particular, we are motivated by the open question: *Can a decentralized algorithm make use of predictions to be competitive for networked online convex optimization in an adversarial environment when spatial and temporal costs are considered?*

**Contributions.** This paper provides the first competitive algorithm for decentralized learning in networked online convex optimization. Agents in a network must each make a decision at each time step, to minimize a global cost which is the sum of convex node costs, spatial interaction costs and temporal interaction costs. We propose a predictive control framework called Localized Predictive Control (LPC, Algorithm 2) and prove that it achieves a competitive ratio of $1 + \widetilde{O}(\rho_T^k) + \widetilde{O}(\rho_S^r)$, which approaches 1 exponentially fast as the prediction horizon $k$ and communication radius $r$ increase simultaneously. Our results quantify the improvement in competitive ratio from increasing the communication radius $r$ (which also increases the computational requirements) versus increasing the prediction horizon $k$, and imply that – as a function of problem parameters – one of the two "resources" $k$ and $r$ emerges as the bottleneck to algorithmic performance. Given any target competitive ratio, we find the minimum required prediction horizon $k$ and communication radius $r$ as functions of the temporal interaction strength and the spatial interaction strength, resp.

Further, we show that LPC is order-optimal in terms of $k$ and $r$ by proving a lower bound on the competitive ratio for any online algorithm. We formalize the near optimality of our algorithm by showing that a resource augmentation bound follows from our upper and lower bounds: our algorithm with given $k$ and $r$ performs at least as well as the best possible algorithm that is forced to work with $k'$ and $r'$ which are a constant factor smaller than $k$ and $r$ respectively.

The algorithm we propose, LPC, is inspired by Model Predictive Control (MPC). After fixing the prediction horizon $k$ and the communication radius $r$, each agent makes an individual decision by solving a $k$-time-step optimization problem, on a local neighborhood centered at itself and with radius $r$. In doing so, the algorithm utilizes all available information and makes a "greedy" decision. One benefit of this algorithm is its simplicity and interpretability, which is often important

for practical applications. Moreover, since the algorithm is local, the computation needed for each agent is independent of the network size.

Our main results are enabled by a new analysis methodology which obtains two separate decay factors for the propagation of decision errors (a temporal decaying factor $\rho_T$ and spatial decaying factor $\rho_S$) through a novel perturbation analysis. Specifically, the perturbation analysis seeks to answer the following question: If we perturb the boundary condition of an agent $v$'s $r$-hop neighborhood at the time step which is $\tau$-th later than the present, how does that affect $v$'s current decision, in terms of spatial distance $r$ and temporal distance $\tau$? With our analysis, we are able to bound the impact on $v$'s current decision by $O(\rho_T^\tau \rho_S^r)$, where the decay factors $\rho_T$ and $\rho_S$ increase with the strength of temporal/spatial interactions among individual decisions. This novel analysis is critical for deriving a competitive ratio that distinguishes the decay rate for temporal and spatial distances.

To illustrate the use of our results in a concrete application, Section 3.5 provides a detailed discussion of dynamic multiproduct pricing, which is a central problem in revenue management. The resulting revenue maximization problem fits into our theoretical framework, and we deduce from our results that LPC guarantees near optimal revenue, in addition to reducing the computational burden (Schlosser, 2016) and providing interpretable prices (Biggs, Sun, and Ettl, 2021) for products.

**Related Work.** This paper contributes to the literature in three related areas, each of which we describe below.

*Distributed Online Convex Optimization.* Our work relates to a growing literature on distributed online convex optimization with time-varying cost functions over multi-agent networks. Many recent advances have been made including distributed OCO with delayed feedback (Cao and Basar, 2021), coordinating behaviors among agents (Li, Yi, and Xie, 2021a; Cao and Başar, 2021), and distributed OCO with a time-varying communication graph (Hosseini, Chapman, and Mesbahi, 2016; Akbari, Gharesifard, and Linder, 2017; Yuan, Proutiere, and Shi, 2021; Li, Yi, and Xie, 2021b; Yi et al., 2020). A common theme of the previous literature is the idea that agents can only

access partial information of time-varying global loss functions, thus requiring local information exchange between neighboring agents. To the best of our knowledge, our paper is the first in this literature to provide competitive ratio bounds or consider spatial and temporal costs, e.g., switching costs.

*Online Convex Optimization (OCO) with Switching Costs.* Online convex optimization with switching costs was first introduced in (Lin et al., 2012) to model dynamic power management in data centers. Different forms of cost functions have been studied since then, e.g., (Chen, Goel, and Wierman, 2018; Shi et al., 2020; Lin, Goel, and Wierman, 2020), in order to fit a variety of applications from video streaming (Joseph and Veciana, 2012) to energy markets (Kim and Giannakis, 2017). The quadratic form of switching cost was first proposed in (Goel and Wierman, 2019) and yields connections to optimal control, which were further explored in (Lin et al., 2021). The literature has focused entirely on the centralized, single-agent setting. Our paper contributes to this literature by providing the first analysis of switching costs in a networked setting with a decentralized algorithm.

*Perturbation Analysis of Online Algorithms.* Sensitivity analysis of convex optimization problems studying the properties of the optimal solutions as a function of the problem parameters has a long and rich history (see (Fiacco and Ishizuka, 1990) for a survey). The works that are most related to ours consider the specific class of problems where the decision variables are located on a horizon axis, or consider a general network and aim to show the impact of a perturbation on a decision variable is exponentially decaying in the graph distance from that variable, e.g., (Shin, Anitescu, and Zavala, 2021; Shin and Zavala, 2021; Lin et al., 2021). The idea of using exponentially decaying perturbation bounds to analyze an online algorithm is first proposed in (Lin et al., 2021), where only the temporal dimension is considered. This style of perturbation analysis is key to the proof of our competitive bounds and, to prove our competitive bounds, we provide new perturbation results that separate the impact of spatial and temporal costs in a network for the first time. Additionally, our analysis is enabled by new results on the decay rate of a product of exponential decay matrices, which may be of independent interest.

**Notation.** A complete notation table can be found in Section A.1. Here we describe the most commonly used notation. In a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we use $d_{\mathcal{G}}(v, u)$ to denote the distance (i.e. the length of the shortest path) between two vertices $v$ and $u$. $N_v^r$ denotes the $r$-hop neighborhood of vertex $v$, i.e., $N_v^r := \{u \in \mathcal{V} \mid d_{\mathcal{G}}(u, v) \leq r\}$. $\partial N_v^r$ denotes the boundary of $N_v^r$, i.e., $\partial N_v^r = N_v^r \setminus N_v^{r-1}$. We generalize these notations to temporal-spatial graphs as follows. Let $\times$ denote the Cartesian product of sets, and $N_{(t,v)}^{(k,r)} := \{\tau \in \mathbb{Z} \mid t \leq \tau < t + k\} \times N_v^r$, $\partial N_{(t,v)}^{(k,r)} := N_{(t,v)}^{(k,r)} \setminus N_{(t,v)}^{(k-1,r-1)}$. For any subset of vertices $S$, we use $\mathcal{E}(S)$ to denote the set of all edges that have both endpoints in $S$, and define $S_+ = \{u \in \mathcal{V} \mid \exists v \in S \text{ s.t. } \text{dist}(u, v) \leq 1\}$ (i.e., $S$ and its 1-hop neighbors). Let $\Delta$ denote the maximum degree of any vertex in $\mathcal{G}$; $h(r) := \sup_v |\partial N_v^r|$. We say a function is in $C^2$ if it is twice continuously differentiable. We use $\|\cdot\|$ to denote the (Euclidean) 2-norm for vectors and the induced 2-norm for matrices.

### 3.2 Problem Setting

We consider a set of agents in a networked system where each agent individually decides on an action at each time step and the agents cooperatively seek to minimize a global cost over a finite time horizon $H$. Specifically, we consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of agents. Each vertex $v \in \mathcal{V}$ denotes an individual agent, and two agents $v$ and $u$ interact with each other if and only if they are connected by an undirected edge $(v, u) \in \mathcal{E}$. At each time step $t = 1, 2, \ldots, H$, each agent $v$ picks an $n$-dimensional local action $x_t^v \in D_t^v$, where $n$ is a positive integer and $D_t^v \subset \mathbb{R}^n$ is a convex set of feasible actions. The global action at time $t$ is the vector of agent actions $x_t = \{x_t^v\}_{v \in \mathcal{V}}$, and incurs a global state cost, which is the sum of three types of local cost functions:

- **Node costs**: Each agent $v$ incurs a time-varying node cost $f_t^v(x_t^v)$, which characterizes agent $v$'s local preference for its local action $x_t^v$.
- **Temporal interaction costs**: Each agent $v$ incurs a time-varying temporal interaction cost $c_t^v(x_t^v, x_{t-1}^v)$, that characterizes how agent $v$'s previous local action $x_{t-1}^v$ interacts with its current local action $x_t^v$.

- **Spatial interaction costs**: Each edge $e = (v, u)$ incurs a time-varying spatial interaction cost[1] $s_t^e(x_t^v, x_t^u)$. This characterizes how agents $v$ and $u$'s current local actions affect each other.

In our model, the node cost is the part of the cost that only depends the agent's current local action. If the other two types of costs are zero functions, each agent will trivially pick the minimizer of its node cost. Temporal interaction costs encourage agents to choose a local action that is "compatible" with their previous local action. For example, a temporal interaction could be a switching cost which penalizes large deviations from the previous action, in order to make the trajectory of local actions "smooth". Such switching costs can be found in work on single-agent online convex optimization, e.g., (Chen, Goel, and Wierman, 2018; Goel et al., 2019; Lin, Goel, and Wierman, 2020). Spatial interaction costs, on the other hand, can be used to enforce some collective behavior among the agents. For example, spatial interaction can model the probability that one agent's actions affects its neighbor's actions in diffusion processes on social networks (Kempe, Kleinberg, and Tardos, 2015); or model interactions between complement/substitute products in multiproduct pricing (Candogan, Bimpikis, and Ozdaglar, 2012).

Our analysis is based on standard smoothness and convexity assumptions on the local cost functions (see Section A.1 for definitions of smoothness and strong convexity):

**Assumption 3.2.1.** *For $\mu > 0, \ell_f < \infty, \ell_T < \infty, \ell_S < \infty$, the local cost functions and feasible sets for all $t, v, e$ satisfy:*

- $f_t^v : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *is $\mu$-strongly convex, $\ell_f$-smooth, and in $C^2$;*
- $c_t^v : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *is convex, $\ell_T$-smooth, and in $C^2$;*
- $s_t^e : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *is convex, $\ell_S$-smooth, and in $C^2$;*
- $D_t^v \subseteq \mathbb{R}^n$ *satisfies $int(D_t^v) \neq \emptyset$ and can be written as $D_t^v := \{x_t^v \in \mathbb{R}^n \mid (g_t^v)_i(x_t^v) \leq 0, \forall 1 \leq i \leq m_t^v\}$, where each $(g_t^v)_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function in $C^2$.*

Note that the assumptions above are common, even in the case of single-agent online convex optimization, e.g., see (Li, Qu, and Li, 2020; Shi et al., 2020; Lin et al., 2021).

[1]Since $e$ is an undirected edge, the order we write the two inputs (the action of $v$ and the action of $u$) does not matter. Note that $s_t^e$ can be asymmetric for agents $v$ and $u$, e.g., $s_t^e(x_t^v, x_t^u) = s_t^e(x_t^u, x_t^v) = \|x_t^v + 2x_t^u\|^2$.

It is useful to separate the global stage costs into two parts based on whether the cost term depends only on the current global action or whether it also depends on the previous action. Specifically, the part that only depends on the current global action $x_t$ is the sum of all node costs and spatial interaction costs. We refer to this component as the *(global) hitting cost* and denote it as

$$f_t(x_t) := \sum_{v \in \mathcal{V}} f_t^v(x_t^v) + \sum_{(v,u) \in \mathcal{E}} s_t^{(v,u)}(x_t^v, x_t^u).$$

The rest of the global stage cost involves the current global action $x_t$ and the previous global action $x_{t-1}$. We refer to it as the *(global) switching cost* and denote it as

$$c_t(x_t, x_{t-1}) = \sum_{v \in \mathcal{V}} c_t^v(x_t^v, x_{t-1}^v).$$

Combining the global hitting and switching costs, the networked agents work cooperatively to minimize the total global stage costs in a finite horizon $H$ starting from a given initial global action $x_0$ at time step 0: $cost(ALG) := \sum_{t=1}^{H} \left( f_t(x_t) + c_t(x_t, x_{t-1}) \right)$, where $ALG$ denotes the decentralized online algorithm used by the agents. The offline optimal cost is the clairvoyant minimum cost one can incur on the same sequence of cost functions and the initial global action $x_0$ at time step $0$, i.e., $cost(OPT) := \min_{x_{1:H}} \sum_{t=1}^{H} \left( f_t(x_t) + c_t(x_t, x_{t-1}) \right)$.

We measure the performance of any online algorithm ALG by the competitive ratio (CR), which is a widely-used metric in the literature of online optimization, e.g., (Chen, Goel, and Wierman, 2018; Goel et al., 2019; Argue, Gupta, and Guruganesh, 2020).

**Definition 3.2.1.** *The competitive ratio of online algorithm $ALG$ is the supremum of $cost(ALG)/cost(OPT)$ over all possible problem instances, i.e., $CR(ALG) := \sup_{\mathcal{G}, H, x_0, \{f_t^v, c_t^v, s_t^e, D_t^v\}} cost(ALG)/cost(OPT)$.*

Finally, we define the partial hitting and switching costs over subsets of the agents. In particular, for a subset of agents $S \subseteq \mathcal{V}$, we denote the joint action over $S$ as $x_t^S := \{x_t^v \mid v \in S\}$ and

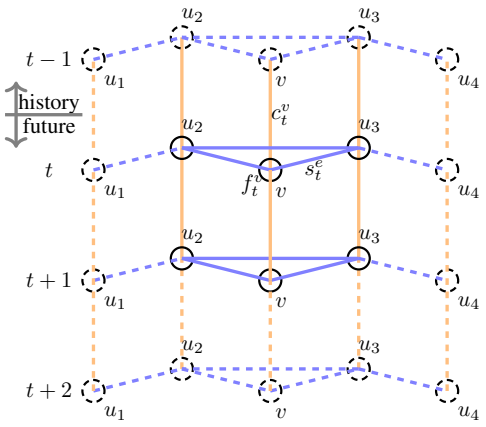


Figure 3.1: Illustration of information model

define the partial hitting cost and partial switching cost over $S$ as

$$f_t^S(x_t^{S_+}) := \sum_{v \in S} f_t^v(x_t^v) + \sum_{(v,u) \in \mathcal{E}(S_+)} s_t^{(v,u)}(x_t^v, x_t^u),$$
$$c_t^S(x_t^S, x_{t-1}^S) := \sum_{v \in S} c_t^v(x_t^v, x_{t-1}^v), \tag{3.1}$$

This notation is useful for presenting decentralized online algorithms where the optimizations are performed over the $r$-hop neighborhood of each agent.

### 3.2.1 Information Availability Model

We assume that each agent has access to local cost functions up to a *prediction horizon* $k$ into the future, for themselves and their neighborhood up to a *communication radius* $r$. In more detail, recall that $N_v^r$ denotes the $r$-hop neighborhood of agent $v$, i.e., $N_v^r := \{u \in \mathcal{V} \mid d_{\mathcal{G}}(u, v) \leq r\}$. Before picking a local action $x_t^v$ at time $t$, agent $v$ can observe $k$ steps of future node costs, temporal interaction costs, and spatial interaction costs within its $r$-hop neighborhood, $\{\{(f_\tau^u, c_\tau^u) \mid u \in N_v^r\}, \{s_\tau^e \mid e \in \mathcal{E}(N_v^r)\}\}_{t \leq \tau < t+k}$, and the previous local actions in $N_v^r$: $\{x_{t-1}^u \mid u \in N_v^r\}$.

We provide an illustration of the local cost functions known to agent $v$ at time $t$ with $k = 2$, $r = 1$, $\mathcal{V} = \{u_1, u_2, v, u_3, u_4\}$ and $\mathcal{E} = \{(u_1, u_2), (u_2, v), (v, u_3), (u_2, u_3), (u_3, u_4)\}$) in Figure 3.1. In the figure, the black circles, blue lines, and orange lines denote the node costs, temporal

interaction costs, and spatial interaction costs respectively. The known functions are marked by solid lines. Note that, in addition to the local cost functions, agent $v$ also knows the local actions in $N_v^r$ at time $t-1$, which are not illustrated in the figure.

To simplify notation, in cases when the prediction horizon exceeds the whole horizon length $H$, we adopt the convention that $f_t^v(x_t^v) = \frac{\mu}{2}\left\|x_t^v\right\|^2$, $c_t^v \equiv s_t^e \equiv 0$ and $D_t^v = \mathbb{R}^n$ for $t > H$. These extended definitions do not affect our original problem with horizon $H$.

As in many previous works studying the power of predictions in the online optimization literature, e.g., (Yu et al., 2020; Lin, Goel, and Wierman, 2020; Li, Qu, and Li, 2020; Lin et al., 2021), we assume the $k$-step predictions of cost functions are *exact* and leave the case of *inexact* predictions for future work. This model is reasonable in the case where the predictors can be trained to be very accurate for the near future. Although we focus on exact predictions, we also discuss how to extend this available information model to include inexact predictions in Section 3.4.3.

### 3.3 Algorithm and Main Results

**Algorithm 2** Localized Predictive Control (for agent $v$)

**Parameters:** $k$ and $r$.
**for** $t = 1$ **to** $H$ **do**
Receive information $\{x_{t-1}^u \mid u \in N_v^r\}$ and

$$\{\{(f_\tau^u, c_\tau^u) \mid u \in N_v^r\}, \{s_\tau^e \mid e \in \mathcal{E}(N_v^r)\}\}_{t \le \tau < t+k}.$$

Choose local action $x_t^v$ to be the $(t,v)$-th element in

$$\psi_{(t,v)}^{(k,r)}\left(\{x_{t-1}^u \,|\, u \in N_v^r\}, \left\{\theta_\tau^u \,|\, (\tau,u) \in \partial N_{(t,v)}^{(k,r)}\right\}\right)$$

the solution of (3.2), where $\theta_\tau^u \coloneqq \arg\min_{y \in D_\tau^u} f_\tau^u(y)$.
**end for**

In this section we present our main results, which show that our simple and practical LPC algorithm can achieve an order-optimal competitive ratio for the networked online convex optimization problem. We first introduce LPC in Section 3.3.1. Then, we present the key idea that leads to our competitive ratio bound: a novel perturbation-based analysis (Section 3.3.2). Next, we use our

perturbation analysis to derive bounds on the competitive ratio in Section 3.3.3. Finally, we show that the competitive ratio of LPC is order-optimal in Section 3.3.4. An outline that highlights the major novelties in our proofs can be found in Section 3.4.

### 3.3.1 Localized Predictive Control (LPC)

The design of LPC is inspired by the classical model predictive control (MPC) framework (Garcia, Prett, and Morari, 1989), which leverages all available information at the current time step to decide the current local action "greedily". In our context, when an agent $v$ wants to decide its action $x_t^v$ at time $t$, the available information includes previous local actions in the $r$-hop neighborhood and $k$-step predictions of all local node costs, temporal/spatial interaction costs. The boundaries of all available information, which are formed by $\{t-1\} \times N_v^r$ and $\partial N_{(t,v)}^{(k,r)}$, are illustrated in Figure 3.2 where the underlying graph is replicated over the time dimension. The orange node marks the decision variable at $(t, v)$. The green part denotes the decisions in $N_v^r$ at time $(t-1)$. The blue "U" shape denotes the boundary of available information for node $v$ at time $t$. Edge $e \coloneqq (v, q)$.

The pseudocode for LPC is presented in Algorithm 2. For each agent $v$ at time step $t$, LPC fixes the actions on the boundaries of available information and then solves for the optimal actions inside the boundaries. Specifically, define $\psi_{(t,v)}^{(k,r)}\left(\{y_{t-1}^u \mid u \in N_v^r\}, \{z_\tau^u \mid (\tau, u) \in \partial N_{(t,v)}^{(k,r)}\}\right)$ as the optimal solution of the problem

$$
\begin{aligned}
\min \sum_{\tau=t}^{t+k-1} & \left(f_\tau^{(N_v^{r-1})}\left(x_\tau^{(N_v^r)}\right) + c_\tau^{(N_v^r)}\left(x_\tau^{(N_v^r)}, x_{\tau-1}^{(N_v^r)}\right)\right) \\
\text{s.t. } & x_{t-1}^u = y_{t-1}^u, \forall u \in N_v^r, \\
& x_\tau^u = z_\tau^u, \forall (\tau, u) \in \partial N_{(t,v)}^{(k,r)}, \\
& x_\tau^u \in D_\tau^u, \forall (\tau, u) \in N_{(t,v)}^{(k-1,r-1)},
\end{aligned} \tag{3.2}
$$

where the partial hitting cost and partial switching cost $f_\tau^S$ and $c_\tau^S$ for a subset $S$ of agents were defined in (3.1). Note that $\psi_{(t,v)}^{(k,r)}\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ is a matrix of actions (in $\mathbb{R}^n$) indexed by $(\tau, u) \in N_{(t,v)}^{(k-1,r-1)}$. (When the context is clear, we use the shorthand $\psi_{(t,v)}^{(k,r)}\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$.) Once the

parameters $\{y^u_{t-1}\}$ and $\{z^u_\tau\}$ are fixed, the agent $v$ can leverage its knowledge of the local cost functions to solve (3.2).

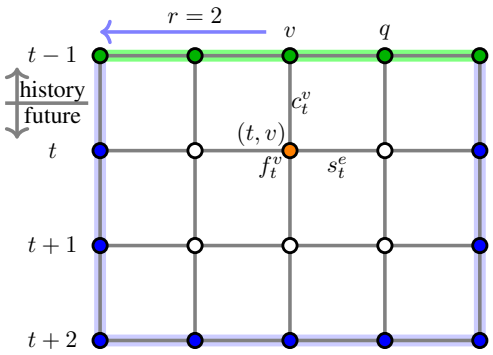


Figure 3.2: Illustration of LPC with $k = 3, r = 2$ on a line graph.

LPC fixes the parameters $\{y^u_{t-1}\}$ to be $\{x^u_{t-1}\}$, which are the previous local actions in $N^r_v$, and fixes the parameters $\{z^u_\tau\}$ to be the minimizers of local node cost functions at nodes in $\partial N^{(k,r)}_{(t,v)}$. The selection of the parameters at nodes in $\partial N^{(k,r)}_{(t,v)}$ plays a similar role as the terminal cost of classical MPC in centralized settings.

For a single-agent system, MPC-style algorithms are perhaps the most prominent approach for optimization-based control (Garcia, Prett, and Morari, 1989) because of their simplicity and excellent performance in practice. LPC extends the ideas of MPC to a multi-agent setting in a networked system by leveraging available information in both the temporal and spatial dimensions, whereas classical MPC focuses only on the temporal dimension. This change leads to significant technical challenges in the analysis.

### 3.3.2 Perturbation Analysis

The key idea underlying our analysis of LPC is that the impact of perturbations to the actions at the boundaries of the available information of an agent decay quickly, in fact exponentially fast, in the distance of the boundary from the agent. This quick decay means that small errors cannot build up to hurt algorithm performance.

In this section, we formally study such perturbations by deriving several new results which generalize perturbation bounds for online convex optimization problems on networks. Our bounds

capture both the effect of temporal interactions as well as spatial interactions between agent actions, which is a more challenging problem compared to previous literature which considers either temporal interactions (Lin et al., 2021) or spatial interactions (Shin, Anitescu, and Zavala, 2021) but not both simultaneously.

More specifically, recall from Section 3.3.1 that for each agent $v$ at time $t$, LPC solves an optimization problem $\psi_{(t,v)}^{(k,r)}$ where actions on the boundaries of available information (i.e., $\{t-1\} \times N_v^r$ and $\partial N_{(t,v)}^{(k,r)}$) are fixed. By the principle of optimality, we know that if the actions on the boundaries are selected to be identical with the offline optimal actions, the agent can decide its current action optimally by solving $\psi_{(t,v)}^{(k,r)}$. However, due to the limits on the prediction horizon and communication radius, LPC can only approximate the offline optimal actions on the boundaries (we do this by using the minimizer of node cost functions). The key idea to our analysis of the optimality gap of LPC is by first asking: *If we perturb the parameters of $\psi_{(t,v)}^{(k,r)}$, i.e., the actions on the information boundaries, how large is the resulting change in a local action $x_{t_0}^{v_0}$ for $(t_0, v_0) \in N_{(t,v)}^{(k,r)} \setminus \partial N_{(t,v)}^{(k,r)}$ (in the optimal solution to* (3.2)*)?*

Ideally, we would like the above impact to decay exponentially fast in the graph distance between node $v_0$ and the communication boundary for node $v$ (i.e., $r$ minus the graph distance between $v_0$ and $v$), as well as in the temporal distance between $t_0$ and $t$. We formalize this goal as *exponentially decaying local perturbation bound* in Definition 3.3.1. We then show in Theorem 3.3.1 and Theorem 3.3.2 that such bounds hold under appropriate assumptions.

**Definition 3.3.1.** *Define $x_{t_0}^{v_0} \coloneqq \psi_{(t,v)}^{(k,r)}\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)_{(t_0,v_0)}$, and $(x_{t_0}^{v_0})' \coloneqq \psi_{(t,v)}^{(k,r)}\left(\{(y_{t-1}^u)'\}, \{(z_\tau^u)'\}\right)_{(t_0,v_0)}$ for arbitrary boundary parameters $\{(y_{t-1}^u)\}, \{(z_\tau^u)\}$ and $\{(y_{t-1}^u)'\}, \{(z_\tau^u)'\}$. We say an **exponentially decaying local perturbation bound** holds if for non-negative constants*

$$
\begin{aligned}
C_1 &= C_1(\ell_T/\mu, (\Delta\ell_S)/\mu) < \infty, \\
C_2 &= C_2(\ell_T/\mu, (\Delta\ell_S)/\mu) < \infty, \\
\rho_T &= \rho_T(\ell_T/\mu) < 1, \rho_S = \rho_S((\Delta\ell_S)/\mu) < 1,
\end{aligned}
$$

*for any* $(t_0, v_0)$ *and arbitrary boundary parameters* $\{(y^u_{t-1})'\}, \{(z^u_\tau)'\}, \{(y^u_{t-1})\}, \{(z^u_\tau)\}$*, we have:*

$$\begin{aligned}
&\left\|x^{v_0}_{t_0} - (x^{v_0}_{t_0})'\right\| \\
\leq\ & C_1 \sum_{(u,\tau)\in\partial N^{(k,r)}_{(t,v)}} \rho_T^{|t_0-\tau|}\rho_S^{\text{dist}(v_0,u)} \left\|z^u_\tau - (z^u_\tau)'\right\| \\
&+ C_2 \sum_{u\in N^r_v} \rho_T^{t_0-(t-1)}\rho_S^{\text{dist}(v_0,u)} \left\|y^u_{t-1} - (y^u_{t-1})'\right\|.
\end{aligned}$$

Perturbation bounds were recently found to be a promising tool for the analysis of adaptive control and online optimization models (Lin et al., 2021). The exponentially decaying local perturbation bound defined above is similar in spirit to two recent results, i.e., (Lin et al., 2021) derives a similar perturbation bound for line graphs and (Shin, Anitescu, and Zavala, 2021) for general graphs with local perturbations. In fact, one may attempt to derive such a bound by applying these results directly; however, a major weakness of the direct approach is that it will yield $\rho_T = \rho_S$, i.e., it cannot distinguish between spatial and temporal dependencies, and the bound deteriorates as $\max\{\ell_T/\mu, \ell_S/\mu\}$ increases. For instance, even if the temporal interactions are weak (i.e., $\ell_T/\mu \approx 0$), $\rho_T = \rho_S$ can still be close to 1 if $\ell_S/\mu$ is large, leading to a large slack in the perturbation bound for small prediction horizons $k$.

We overcome this limitation by redefining the action variables. Specifically, to focus on the temporal decay effect, we regroup all local actions in $\{\tau\} \times N^r_v$ as a "large" decision variable for time $\tau$ (in Figure 3.1 we would group each horizontal blue plane in $N^r_v$ to create a new variable). After regrouping, we have $(k+1)$ "large" decision variables located on a line graph, where the strength of the interactions between consecutive variables is upper bounded by $\ell_T$. On the other hand, to focus on spatial decay, we regroup all local actions in $\{\tau \mid t-1 \leq \tau < t+k\} \times \{v\}$ as a decision variable (in Figure 3.1 we would group each vertical orange line connecting from $t-1$ to $t+k-1$ to create a new variable). After regrouping, we have $|\mathcal{V}|$ "large" decision variables located on $\mathcal{G}$, where the strength of the interactions between two neighbors is upper bounded by $\ell_S$. Averaging over the two perturbation bounds (since we have two valid bounds, their average is

also a valid bound) provides the following exponentially decaying local perturbation bound (see (A.4) in Section A.2.1 for details of the proof).

**Theorem 3.3.1.** *Under Assumption 3.2.1, the exponentially decaying local perturbation bound (Definition 3.3.1) holds with* $C_1 = C_2 = \frac{2\sqrt{\Delta\ell_S \ell_T}}{\mu}$*, and*

$$\rho_T = \sqrt{1 - 2\left(\sqrt{1 + (2\ell_T/\mu)} + 1\right)^{-1}},$$
$$\rho_S = \sqrt{1 - 2\left(\sqrt{1 + (\Delta\ell_S/\mu)} + 1\right)^{-1}}.$$

Note that, as $\ell_T/\mu$ (respectively $\ell_S/\mu$) tends to zero, $\rho_T$ (respectively $\rho_S$) in Theorem 3.3.1 also tends to zero with the scaling $\rho_T = \Theta(\sqrt{\ell_T/\mu})$ (resp. $\rho_S = \Theta(\sqrt{\ell_S/\mu})$).

Next, we provide a tighter bound (through a refined analysis) for the regime where $\mu$ is much larger than $\ell_T, \ell_S$. Specifically, we establish a bound with the scaling $\rho_T = \Theta(\ell_T/\mu)$ and $\rho_S = \Theta(\ell_S/\mu)$. Again, it is not possible to obtain this result from previous perturbation bounds in the literature.

**Theorem 3.3.2.** *Recall* $h(\gamma) \coloneqq \sup_{v\in\mathcal{V}} |\partial N_v^\gamma|$*. Given any* $b_1, b_2 > 0$*, define* $a = \sum_{\gamma\geq 0}(\frac{1+b_1}{1+b_1+b_2})^\gamma h(\gamma)$*,* $\widetilde{a} = \sum_{\gamma\geq 0}(\frac{1}{1+b_1})^\gamma h(\gamma)$ *and* $\gamma_S = \frac{\sqrt{1+\Delta\ell_S/\mu}-1}{\sqrt{1+\Delta\ell_S/\mu}+1}$*. Suppose Assumption 3.2.1 holds,* $a, \widetilde{a} < \infty$ *and* $\mu \geq \max\{8\widetilde{a}\ell_T, \Delta\ell_S(b_1+b_2)/4\}$*. Then the exponentially decaying local perturbation bound (Definition 3.3.1) holds with* $C_1 = C_2 = \max\{\frac{a^2}{2\widetilde{a}(1-4\widetilde{a}l_T/\mu)}, \frac{2a^2\Delta\ell_S/\mu}{\gamma_S(1+b_1+b_2)(1-4\widetilde{a}l_T/\mu)}\}$

$$\rho_T = \frac{4\widetilde{a}\ell_T}{\mu}, \quad \rho_S = (1+b_1+b_2)\gamma_S.$$

Note that $\rho_T, \rho_S < 1$ follow from the condition on $\mu$. Also observe that $\gamma_S = \Theta(\ell_S/\mu)$ as $\ell_S/\mu \to 0$.

The main difference between this result and Theorem 3.3.1 is, instead of dividing and redefining the action variables, we explicitly write down the perturbations along spatial edges and along temporal edges in the original temporal-spatial graph. We observe that per-time-step spatial interactions are characterized by a banded matrix and that the inverse of the banded matrix exhibits

exponential correlation decay, which implies the exponentially decaying local perturbation bounds holds if the perturbed boundary action and the impacted local action we consider are at the same the time step. However, for a multi-time-step problem, to characterize the impact at a local action at some time step due to perturbation at a boundary action at a different time step is a difficult problem. The main technical contribution of this proof is to establish that a product of exponentially decaying matrices still satisfies exponential decay under the conditions in Theorem 3.3.2. In addition, we obtain a tight bound on the decay rate of the product matrix (see Lemma 3.4.3), which may be of independent interest.

Our condition on $a, \widetilde{a} < \infty$ and $\mu > \max\{8\widetilde{a}\ell_T, \Delta\ell_S(b_1 + b_2)/4\}$ characterizes a tradeoff between the allowable neighborhood boundary sizes $h(\gamma)$, and how large $\mu$ needs to be compared to the interaction cost parameters $\ell_T, \ell_S$. At one extreme, if $h(\gamma) = \Delta^\gamma$, then by setting $b_1 = 2\Delta - 1$ and $b_2 = 4\Delta^2 - 2\Delta$, we obtain $a = \widetilde{a} = 2$ but must make a strong requirement on $\mu$, namely, $\mu > \max\{16\ell_T, \Delta^3\ell_S(1 - \frac{1}{4\Delta^2})\}$. At the other extreme, if $h(\gamma) \leq O(poly(\gamma))$ (as is the case if $\mathcal{G}$ is a grid), then $a, \widetilde{a} < \infty$ holds for any $b_1, b_2 > 0$, we can impose a weaker requirement on $\mu$: for example, taking $b_1 = b_2 = 1$ yields a requirement $\mu > \max\{8\widetilde{a}\ell_T, \Delta\ell_S/2\}$ (where $\widetilde{a} = \sum_{\gamma \geq 0}(\frac{1}{2})^\gamma h(\gamma)$); which grows only linearly in $\Delta$, and compares favorably with the $\mu > \Omega(\Delta^3)$ requirement which arose earlier.

Proofs of Theorem 3.3.1 and Theorem 3.3.2 are in Section A.2.

### 3.3.3 From Perturbations to Competitive Bounds

We now present our main result, which bounds the competitive ratio of LPC using the exponentially decaying local perturbation bounds defined in the previous section.

Before presenting the result, we first provide some intuition as to why the perturbation bounds are useful for deriving the competitive ratio bound. Specifically, to bound the competitive ratio requires bounding the gap between LPC's trajectory and the offline optimal trajectory. This gap comes from the following two sources: (i) the per-time-step error made by LPC due to its limited prediction horizon and communication radius; and (ii) the cumulative impact of all per-time-step

errors made in the past. Intuitively, the local perturbation bounds we derive in Section 3.3.2 allow us to bound the per-step error made jointly by all agents in LPC, and then we use the perturbation bounds from (Lin et al., 2021) help us to bound the second type of cumulative errors.

We present our main result in the following theorem and defer a proof outline to Section 3.4.2. A formal proof can be found in Section A.3.3.

**Theorem 3.3.3.** *Suppose Assumption 3.2.1 and the exponentially decaying local perturbation bound in Definition 3.3.1 holds. Define* $\rho_G \coloneqq 1 - 2 \cdot \left(\sqrt{1 + (2\ell_T/\mu)} + 1\right)^{-1}$, *and define* $C_3(r) \coloneqq \sum_{\gamma=0}^{r} h(\gamma) \cdot \rho_S^{\gamma}$. *If parameters* $r$ *and* $k$ *of LPC are large enough such that*

$$O\left(h(r)^2 \cdot \rho_S^{2r} + C_3(r)^2 \cdot \rho_T^{2k} \cdot \rho_G^{2k}\right) \leq \frac{1}{2},$$

*then the competitive ratio of LPC is upper bounded by*

$$1 + O\left(h(r)^2 \cdot \rho_S^{r}\right) + O\left(C_3(r)^2 \cdot \rho_T^{k}\right).$$

*Here the* $O(\cdot)$ *notation hides factors that depend polynomially on* $\ell_f/\mu, \ell_T/\mu$, *and* $(\Delta\ell_S)/\mu$; *see Section A.3.3.*

Recall that $h(r)$ denotes the size of the largest $r$-hop boundary in $\mathcal{G}$. The bound in Theorem 3.3.3 implies that if $h(r)$ can be upper bounded by $poly(r) \cdot \rho_S^{-\frac{(1-\iota)r}{2}}$ for some constant $\iota > 0$, the competitive ratio of LPC can be upper bounded by $1 + O(\rho_S^{\iota r}) + O(\rho_T^k)$, because $C_3(r)$ can be upper bounded by some constant that depends on $\iota$ in this case. Therefore, the competitive ratio improves exponentially with respect to the prediction horizon $k$ and communication radius $r$.

Note that the assumption $h(r) \leq poly(r) \cdot \rho_S^{-\frac{(1-\iota)r}{2}}$ is not particularly restrictive: For commonly seen graphs like an $m$-dimensional grid, $h(r)$ is polynomial in $r$, so $\iota = 1$ works. More generally, for graphs with bounded degree $\Delta < \infty$, there exists $\delta = \delta(\Delta) > 0$ such that, for any graph with node degrees bounded above by $\Delta$ and any $\ell_S/\mu \leq \delta$, we have $\rho_S$ (from either Theorem 3.3.1 or Theorem 3.3.2) will be small enough that, e.g., $h(r) \leq \Delta^r = O(\rho_S^{-\frac{r}{4}})$; i.e., $\iota = 1/2$ works.

Thus we can eliminate the dependence on $h(r)$ and $C_3(r)$ in the competitive ratio by making additional assumptions on $\ell_S/\mu$. This result is stated in Corollary 3.3.4 whose proof is deferred to Section A.3.4. Corollary 3.3.4 is a corollary of Theorem 3.3.3 and Theorem 3.3.2. We use the bound in Theorem 3.3.2 and not the bound in Theorem 3.3.1 because Theorem 3.3.2 is tighter when $\ell_S/\mu$ is small.

**Corollary 3.3.4.** *Suppose Assumption 3.2.1 and inequalities $\ell_S/\mu \leq \Delta^{-7}$, and $\ell_T/\mu \leq 1/16$ hold. If $r$ and $k$ satisfy that $O\left(\rho_S^r + \rho_T^{2k} \cdot \rho_G^{2k}\right) \leq \frac{1}{2}$, then the competitive ratio of LPC is upper bounded by $1 + O\left(\rho_S^{r/2}\right) + O\left(\rho_T^k\right)$, where $\rho_S$ and $\rho_T$ are given by Theorem 3.3.2 with parameters $b_1 = 2\Delta - 1$ and $b_2 = 4\Delta^2 - 2\Delta$. The $O(\cdot)$ notation hides factors that depend polynomially on $\ell_f/\mu, \ell_T/\mu$, and $(\Delta\ell_S)/\mu$, see Section A.3.4 for the full constants.*

### 3.3.4 A Lower Bound

We show that the competitive ratio in Theorem 3.3.3 is order-optimal by deriving a lower bound on the competitive ratio of any decentralized online algorithm with prediction horizon $k$ and communication radius $r$. The specific constants and a proof of Theorem 3.3.5 can be found in Section A.4.

**Theorem 3.3.5.** *When $\Delta \geq 3$, the competitive ratio of any decentralized online algorithm is lower bounded by $1 + \Omega(\lambda_T^k) + \Omega(\lambda_S^r)$. Here, the decay factor $\lambda_T$ is given by*

$$\lambda_T = \left(1 - 2\left(\sqrt{1 + (4\ell_T/\mu)} + 1\right)^{-1}\right)^2.$$

*The decay factor $\lambda_S$ is given by*

$$\lambda_S = \begin{cases} \frac{(\Delta\ell_S/\mu)}{3+3(\Delta\ell_S/\mu)} & \text{if } \Delta\ell_S/\mu < 48 \\ \max\left(\frac{(\Delta\ell_S/\mu)}{3+3(\Delta\ell_S/\mu)}, \left(1 - 4\sqrt{3} \cdot (\Delta\ell_S/\mu)^{-\frac{1}{2}}\right)^2\right) & \text{otherwise.} \end{cases}$$

*The $\Omega(\cdot)$ notation hides factors that depend polynomially on $1/\mu, \ell_T$, and $\ell_S$.*

While Theorem 3.3.5 highlights that Theorem 3.3.3 is order-optimal, the decay factors $\lambda_T, \lambda_S$ in the lower bound differ from their counterparts $\rho_T, \rho_S$ in the upper bound for LPC. To understand the magnitude of the difference, we compare the bounds on graphs with bounded degree $\Delta$. The decay factors are a function of the interaction strengths, which are measured by $\ell_S/\mu$ and $\ell_T/\mu$. Our lower bound on the temporal decay factor $\lambda_T$ and upper bound $\rho_T$ only differ by a constant factor in the log-scale, and the same holds for the lower/upper bound in terms of the spatial decay factor.

To formalize this comparison, we derive a resource augmentation bound that bounds the additional "resources" that LPC needs to outperform the optimal decentralized online algorithm.[2] Here the prediction horizon $k$ and the communication radius $r$ can be viewed as the "resources" available to a decentralized online algorithm in our setting. We ask *how large do $k$ and $r$ given to LPC need to be, to ensure that it beats the optimal decentralized online algorithm given a communication radius $r^*$ and prediction horizon $k^*$?*

We formally state our result in the following corollary and provide a proof in Section A.5.

**Corollary 3.3.6.** *Under Assumption 3.2.1, suppose the optimal decentralized online algorithm achieves a competitive ratio of $c(k^*, r^*)$ with prediction horizon $k^*$ and communication radius $r^*$. Additionally assume that $h(\gamma) = \widetilde{O}\left(\rho_S^{-\gamma/4}\right)$ and $\Delta \geq 3$, where the $\widetilde{O}$ notation hides a factor that depends polynomially on $\gamma$. As $k^*, r^* \to \infty$, LPC achieves a competitive ratio at least as good as that of the optimal decentralized online algorithm when LPC uses a prediction horizon of $k = (4 + o(1))k^*$ and a communication radius of $r = (32 + o(1))r^*$.*

Finally, note that we establish Corollary 3.3.6 based on the local perturbation bound in Theorem 3.3.1 rather than Theorem 3.3.2 for simplicity, because it does not make assumptions on the relationship among $\mu, \ell_T$, and $\ell_S$. We expect that Theorem 3.3.2 can give better resource augmentation bounds under stronger assumptions on $\mu, \ell_T$, and $\ell_S$.

[2] See, e.g., Roughgarden (2020), for an introduction to this flavor of bounds for expressing the near-optimality of an algorithm.

## 3.4 Proof Outline

In this section, we outline the major novelties in our proofs for the tighter exponentially decaying local perturbation bound in Theorem 3.3.2 and the main competitive ratio bound for LPC in Theorem 3.3.3. The full details of the proofs of these and other results are in the appendices following this one.

### 3.4.1 Refined Analysis of Perturbation Bounds

We begin by outlining the four-step structure we use to prove Theorem 3.3.2. Our goal is to highlight the main ideas, while deferring a detailed proof to Section A.2.2.

**Step 1. Establish first order equations**

We define $h$ as the objective function in (3.2), where actions on the boundary are fixed as $\{z_\tau^u | (\tau, u) \in \partial N_{(t,v)}^{(k,r)}\}$ and the actions at time $t-1$ are fixed as $\{x_{t-1}^u | u \in N_v^r\}$. We denote those fixed actions as system parameter

$$\zeta \coloneqq (x_{t-1}^{(N_v^r)}, \{z_\tau^u | (\tau, u) \in \partial N_{(t,v)}^{(k,r)}\}).$$

To avoid writing the time index $t$ repeatedly, we use $\widehat{x}_i$ to denote actions at time $t-1+i$ for $0 \le i \le k$. The main lemma in for this step is the following.

**Lemma 3.4.1.** *Given $\theta \in \mathbb{R}$, system parameter $\zeta$ and perturbation vector $e$, we have*

$$\frac{d}{d\theta}\psi(\zeta + \theta e) = M^{-1}\left(R^{(1)}e_0 + R^{(k-1)}e_k + \sum_{\tau=1}^{k-1} K^{(\tau)}e_\tau\right)$$

*where*

$$M = \nabla^2_{\widehat{x}_{1:k-1}} h(\psi(\zeta + \theta e), \zeta + \theta e),$$

$$R^{(1)} \coloneqq -\nabla_{\widehat{x}_0}\nabla_{\widehat{x}_{1:k-1}} h(\psi(\zeta + \theta e), \zeta + \theta e),$$

$$R^{(k-1)} := -\nabla_{\widehat{x}_k} \nabla_{\widehat{x}_{1:k-1}} h(\psi(\zeta + \theta e), \zeta + \theta e),$$

$$K^{(\tau)} := -\nabla_{\widehat{x}_\tau^{(\partial N_v^T)}} \nabla_{\widehat{x}_{1:k-1}} h(\psi(\zeta + \theta e), \zeta + \theta e).$$

The proof for Lemma 3.4.1 using first order conditions at the global optimal solution for convex function $h(\cdot, \zeta + \theta e)$ and then takes derivatives with respect to to $\theta$. See Section A.2.2 for a proof.

**Step 2: Exploit the structure of matrix $M$**

$M$ is a hierarchical block matrix with the first level of dimension $(k-1) \times (k-1)$. When fixing the first level indices (i.e. time indices) in $M$, the lower level matrices are non-zero only if the difference in the time indices is $\leq 1$. Hence we decompose $M$ to a block diagonal matrix $D$ and tri-diagonal block matrix $A$ with zero matrix on the diagonal. Each diagonal block in $D$ is a graph-induced banded matrix, which captures the Hessian of $h$ in a single time step. Denote each diagonal block as $D_{i,i}$ for $1 \leq i \leq k-1$. Further, for $1 \leq i \leq k-1$, $A_{i,i-1}$ (similarly $A_{i,i+1}$) captures the temporal correlation of individual's action between consecutive time steps. Given this decomposition,

$$M^{-1} = (D + A)^{-1} = D^{-1}(I + AD^{-1})^{-1}.$$

For the ease of notation, we denote $I + AD^{-1}$ by $P$. Note that $P$ is not necessarily a symmetric matrix. Nevertheless, under technical conditions on $P$'s eigenvalues, we have the following power series expansion (Shin, Zavala, and Anitescu, 2020). The details are presented in the Lemma 3.4.2 in Section A.2.2.

**Lemma 3.4.2.** *Under the condition $\mu > 2\ell_T$, we have*

$$P^{-1} = \sum_{\ell \geq 0} (I - P)^\ell. \tag{3.3}$$

To understand the the power series in (3.3), consider the special case where each block $A_{i,j} = \ell_T \cdot I$, and $D_{i,i} = Q$. Denote $P - I$ as $J$, which is equivalent to $AD^{-1}$. Then, we have $J_{i,i} = 0$, $J_{i,i-1} = J_{i,i+1} = \ell_T Q^{-1}$, $J_{i,j} = 0$ when $|i - j| > 1$. Intuitively, $J$ captures the "correlation over

actions" after one time step. More generally, for $\ell \geq 0$ and any two time indices $\tau'$, $\tau$,

$$J^{\ell}_{\tau',\tau} = \ell^{\ell}_{T} Q^{-\ell} b(\ell, \tau, \tau'),$$

where $b(\ell, \tau, \tau')$ is a constant depending on $\ell, \tau, \tau'$ and equal to zero if $\ell < |\tau - \tau'|$.

Given that $Q$ is a graph-induced banded matrix, $Q^{-1}$ satisfies exponential-decay properties, which makes it plausible that $Q^{-\ell}$ is an exponential decay matrix with a slower rate.

For the general case, we need to bound terms similar to $\left\|(D^{-1}_{i_1,i_1} D^{-1}_{i_2,i_2} \cdots D^{-1}_{i_\ell,i_\ell})_{u,v}\right\|$. This is the goal of Step 3.

**Step 3: Properties for general exponential-decay matrices**

The goal of this step is to establish that a product of exponential decay matrices still exhibits exponential decay property under technical conditions about the underlying graph.

**Lemma 3.4.3.** *Given any graph $\mathcal{M} = (\mathcal{V}', \mathcal{E}')$ and integers $d, \ell \geq 1$, suppose block matrices $A_i \in \mathbb{R}^{|\mathcal{V}'|d \times |\mathcal{V}'|d}$ all satisfy exponential decay properties, i.e. exists $C_i \geq 0$, and $0 \leq \lambda < 1$, s.t.,*

$$\|(A_i)_{u,q}\| \leq C_i \lambda^{d_{\mathcal{M}}(u,q)} \quad \text{for any node } u, q \in \mathcal{M}.$$

*Select some $\delta > 0$ s.t. $\lambda' = \lambda + \delta < 1$. If $\widetilde{a} \coloneqq \sum_{k=0}^{\infty} (\frac{\lambda}{\lambda'})^k (\sup_{u \in \mathcal{V}'} |\partial N^k_u|) < \infty$, then $\prod_{i=1}^{\ell} A_i$ satisfies exponential decay properties with decay rate $\lambda'$, i.e.,*

$$\left\| (\prod_{i=1}^{\ell} A_i)_{u,v} \right\| \leq C' (\lambda')^{d_M(u,v)}.$$

*where $C' = (\widetilde{a})^{\ell} \prod_{i=1}^{\ell} C_i$.*

A proof of Lemma 3.4.3 can be found in the Section A.2.2.

**Step 4: Establish correlation decay properties of matrix $M$**

The last step of the proof is to study the properties of $M$. To accomplish this, we first show that, for time indices $i, j \geq 1$, $J^\ell$ has the following properties:

- $(J^\ell)_{i,j} = 0$ if $\ell < |i-j|$ or $\ell - |i-j|$ is odd.
- $(J^\ell)_{i,j}$ is a summation of terms $\prod_{k=1}^{\ell} A_{j_k,i_k} D^{-1}_{i_k,i_k}$ and the number of such terms is bounded by $\binom{\ell}{(\ell-|i-j|)/2}$.

We formally state and prove the above observation in Lemma A.2.2. We can further use Theorem 3.4.3 on block matrices $A_{j_k,i_k} D^{-1}_{i_k}$, which gives the following lemma.

**Lemma 3.4.4.** *Recall* $\gamma_S \coloneqq \frac{\sqrt{1+(\Delta \ell_S/\mu)}-1}{\sqrt{1+(\Delta \ell_S/\mu)}+1}$. *Select* $\delta > 0$ *s.t.* $\gamma'_S = \gamma_S + \delta < 1$ *and* $b \coloneqq \sum_{\gamma \geq 0} (\frac{\gamma_S}{\gamma'_S})^\gamma h(\gamma)$. *Given* $\ell, i, j \geq 1$ *and* $u, q \in \mathcal{V}$*, we have*

$$\left\| ((J^\ell)_{i,j})_{u,q} \right\| \leq \binom{\ell}{(\ell - |i-j|)/2} (b \frac{2\ell_T}{\mu})^\ell (\gamma'_S)^{\text{dist}(u,q)}.$$

Intuitively speaking, Lemma 3.4.4 bounds the correlations over actions for node $u$ at time step $t-1+i$ and action for node $q$ at time step $t-1+j$. We present its proof in the Section A.2.2.

Recall that, for $1 \leq i, j \leq k-1$,

$$M^{-1}_{i,j} = D^{-1}_{i,i} \sum_{\ell \geq 0} (-J)^\ell_{i,j}.$$

With the exponential decaying bounds on matrix $J^\ell$, we can thus conclude Theorem 3.3.2 by following a similar procedure as in the proof of Theorem 3.1 of (Lin et al., 2021). We present the details in Lemma A.2.3.

### 3.4.2 From Perturbation to Competitive Ratio

We now show how to use the result proven in the previous section to prove our competitive ratio bounds in Theorem 3.3.3. Our starting point is the assumption that the exponentially decaying local

perturbation bound in Definition 3.3.1 holds for some $C_1, C_2 > 0$ and $\rho_S, \rho_T \in [0, 1)$, which is established using the proof approach outlined in the Section 3.3.3.

As we discussed in Section 3.3.3, our proof contains two key parts: (i) we bound the per-time-step error of LPC (Lemma 3.4.5); and (ii) show that the per-time-step error does not accumulate to be unbounded (Theorem 3.4.2).

A key observation that enables the above analysis approach is that the aggregation of local per-time-step error made by each agent at $x_t^v$ can be viewed as a global per-time-step error in the joint global action $x_t$. Following this observation, we first introduce a global perturbation bound that focuses on the global action $x_t$ rather than the local actions $x_t^v$. Recall that $f_t$ denotes the global hitting cost (see Section 3.2). Define the optimization problem that solves the optimal trajectory from global state $x_{t-1}$ to $x_{t+p-1}$

$$\widetilde{\psi}_t^p(y, z) = \underset{x_{t:t+p-1}}{\arg\min} \sum_{\tau=t}^{t+p-1} \left(f_\tau(x_\tau) + c_\tau(x_\tau, x_{\tau-1})\right)$$
$$\text{s.t. } x_{t-1} = y, x_{t+p-1} = z, \tag{3.4}$$

and another one that solves the optimal trajectory from global state $x_{t-1}$ to the end of the game

$$\widetilde{\psi}_t(y) = \underset{x_{t:T}}{\arg\min} \sum_{\tau=t}^{T} \left(f_\tau(x_\tau) + c_\tau(x_\tau, x_{\tau-1})\right)$$
$$\text{s.t. } x_{t-1} = y. \tag{3.5}$$

The following global perturbation bound can be derived from Theorem 3.1 in (Lin et al., 2021):

**Theorem 3.4.1** (Global Perturbation Bound)**.** *Under Assumption 3.2.1, the following perturbation bounds hold for optimization problems* (3.4) *and* (3.5)*:*

$$\left\|\widetilde{\psi}_t^p(y, z)_{t_0} - \widetilde{\psi}_t^p(y', z')_{t_0}\right\| \leq C_G \rho_G^{t_0-t+1} \|y - y'\| + C_G \rho_G^{t+p-1-t_0} \|z - z'\|,$$
$$\left\|\widetilde{\psi}_t(y)_{t_0} - \widetilde{\psi}_t(y')_{t_0}\right\| \leq C_G \rho_G^{t_0-t+1} \|y - y'\|,$$

*where* $\rho_G = 1 - 2 \cdot \left(\sqrt{1 + \frac{2\ell_T}{\mu}} + 1\right)^{-1}$ *and* $C_G = \frac{2\ell_T}{\mu}$.

To make the concept of *per-time-step error* rigorous, we formally define it as the distance between the actual next action picked by LPC and the clairvoyant optimal next action from previous action $x_{t-1}$ to the end of the game:

**Definition 3.4.1** (Per-step error magnitude)**.** *At time step* $t$*, given the previous state* $x_{t-1}$*, the (decentralized) online algorithm* $ALG$ *picks* $x_t \in D_t$*. We define error* $e_t$ *as*

$$e_t := \left\| x_t - \widetilde{\psi}_t(x_{t-1})_t \right\|.$$

Using the local perturbation bound in Definition 3.3.1, we show the per-time-step error of LPC decays exponentially with respect to prediction length $k$ and communication range $r$. This result is stated formally in Lemma 3.4.5, and the proof can be found in Section A.3.2.

**Lemma 3.4.5.** *For* $LPC$ *with parameters* $r$ *and* $k$*,* $e_t$ *satisfies*

$$\begin{aligned} e_t^2 = {} & O\left(h(r)^2 \cdot \rho_S^{2r} + C_3(r)^2 \cdot \rho_T^{2k} \rho_G^{2k}\right) \cdot \left\| x_{t-1} - x_{t-1}^* \right\|^2 \\ & + O\left(h(r)^2 \cdot \rho_S^{2r}\right) \sum_{\tau=t}^{t+k-1} \rho_T^{\tau-t} f_\tau(x_\tau^*) \\ & + O\left(C_3(r)^2 \cdot \rho_T^{2k}\right) f_{t+k-1}(x_{t+k-1}^*), \end{aligned}$$

*where* $C_3(r) := \sum_{\gamma=0}^{r} h(\gamma) \cdot \rho_S^\gamma$.

Using the global perturbation bound in Theorem 3.4.1, we show $\sum_{t=1}^{T} \|x_t - x_t^*\|^2$ can be upper bounded by the sum of per-time-step errors of LPC in Theorem 3.4.2. The proof can be found in Appendix A.3.1.

**Theorem 3.4.2.** *Let* $x_0, x_1^*, x_2^*, \ldots, x_H^*$ *denote the offline optimal global trajectory and* $x_0, x_1, x_2, \ldots, x_H$ *denote the trajectory of* $ALG$*. The trajectory of* $ALG$ *satisfies that*

$$\sum_{t=1}^{H} \|x_t - x_t^*\|^2 \leq \frac{C_0^2}{(1 - \rho_G)^2} \sum_{t=1}^{H} e_t^2,$$

*where $C_0 := \max\{1, C_G\}$ and $C_G$ is defined in Theorem 3.4.1.*

To understand the bound in Theorem 3.4.2, we can set all per-time-step error $e_t$ to be zero except a single time step $\tau$. We see the impact of $e_\tau$ on the total squared distance $\sum_{t=1}^{T} \|x_t - x_t^*\|^2$ is up to some constant factor of $e_\tau$. This is because the impact of $e_\tau$ on $\|x_t - x_t^*\|$ decays exponentially as $t$ increases from $\tau$ to $T$.

By substituting the per-time-step error bound in Lemma 3.4.5 into Theorem 3.4.2, one can bound $\sum_{t=1}^{T} \|x_t - x_t^*\|^2$ by the offline optimal cost, which can be converted to the competitive ratio bound in Theorem 3.3.3.

### 3.4.3 Roadmap to Generalize the Proof to Inexact Predictions

In this section, we present a roadmap to generalize our proof to the case where predictions of future cost functions are inexact. In the information availability model in Section 3.2.1, one can study inexact predictions by introducing additional disturbance parameters $\{\delta_t^v, w_t^v, w_t^e\}$ to the 3 types of cost functions and generalize them to $f_t(x_t^v, \delta_t^v), c_t^v(x_t^v, x_{t-1}^v, w_t^v), s_t^e(x_t^v, x_t^u, w_t^e)$ for all $t \in [H], v \in \mathcal{V}, e = (v, u) \in \mathcal{E}$. $\{\delta_t^v, w_t^v, w_t^e\}$ represents the disturbances in the cost functions that are hard to predict exactly. Before the decentralized online algorithm decides each local action, it receives the generalized cost functions $f_t^v(\cdot, \cdot), c_t^v(\cdot, \cdot, \cdot), s_t^e(\cdot, \cdot, \cdot)$ and noisy predictions of the true disturbance parameters $\{\delta_t^v, w_t^v, w_t^e\}$ within $k$ time steps and an $r$-hop neighborhood. In the LPC algorithm (Algorithm 2), the optimization problem $\psi_{(t,v)}^{(k,r)}$ can then be solved with the noisy predictions of disturbance parameters. To analyze the performance of LPC in the presence of prediction errors, one can first generalize the exponentially decaying perturbation bounds in Theorem 3.3.1 and Theorem 3.3.2 to include the perturbations on disturbance parameters similar to what we already did in Theorem A.2.1. The prediction error on disturbance parameters will result in an additional additive term in the per-step error bound in Lemma 3.4.5. If one is willing to assume that the total sum of prediction errors is $O(cost(OPT))$, as in (Antoniadis et al., 2020), one can derive a competitive ratio for LPC by substituting the per-step error bound into Theorem 3.4.2. It is worth noting that the resulting competitive ratio will inevitably depend on the

quality of predictions, and will converge to a limit larger than 1 (under imperfect predictions) as the prediction horizon $k$ and the communication radius $r$ increase.

### 3.5 Application: Multiproduct Pricing

The networked online convex optimization problem captures many applications where decisions at each node must be made online in a networked system. In this section, we give a concrete example - the multiproduct pricing problem, which has been proposed/studied in the previous revenue management literature (Talluri and Ryzin, 2006; Gallego and Topaloglu, 2019; Song and Chintagunta, 2006; Caro and Gallien, 2012; Candogan, Bimpikis, and Ozdaglar, 2012; Chen and Chen, 2015).

We consider a setting where a large company sells $n$ different products and wants to maximize its revenue by adjusting the prices of these products adaptively in a time-varying market. Each vertex/agent $v \in \mathcal{V}$ corresponds to a product and $x_t^v$ denotes its price at time $t$. An edge connects two products $v$ and $u$ if they interact, e.g., because the products are complements or substitutes.

We assume a classical linear demand model based on Talluri and Ryzin (2006) and Gallego and Topaloglu (2019) where the demand of $v$ at time $t$, denoted as $d_t^v$, is given by

$$d_t^v = \underbrace{a_t^v - k_t^v x_t^v}_{\text{Part 1}} \overbrace{+ \sum_{u \in N_v^1 \setminus \{v\}} \eta_t^{(u \to v)} x_t^u}^{\text{Part 2}} \underbrace{+ b_t^v x_{t-1}^v}_{\text{Part 3}},$$

with parameters $a_t^v, k_t^v, b_t^v > 0$, $\eta_t^{(u \to v)} \in \mathbb{R}$ and $x_0^v \coloneqq 0$ for any node $v$. Here, Part 1 corresponds to the nominal demand at price $x_t^v$; Part 2 adds the network externalities from $v$'s complements/substitutes, and Part 3 reflects the pent-up demand of product $v$ due to high price at time $t-1$. Note that the coefficient $\eta_t^{(u \to v)}$ can be different with $\eta_t^{(v \to u)}$. For each undirected edge $e = (u, v)$, we define an aggregate coefficient

$$\gamma_t^e \coloneqq \frac{1}{2} \left( \eta_t^{(u \to v)} + \eta_t^{(v \to u)} \right). \tag{3.6}$$

Moreover, only the aggregated $\{\gamma_t^e\}_{t,e}$ appear as network parameters after our reformulation.

The full revenue maximization problem can be written as

$$\begin{aligned} \max \quad & \sum_{t=1}^{H}\sum_{v\in\mathcal{V}} x_t^v d_t^v = \sum_{t=1}^{H}\sum_{v\in\mathcal{V}} x_t^v (a_t^v - k_t^v x_t^v + \sum_{u\in N_v^1\setminus\{v\}} \eta_t^{(u\to v)} x_t^u + b_t^v x_{t-1}^v) \\ \text{s.t.} \quad & 0 \le x_t^v \le \overline{p}_t^v, \end{aligned} \tag{3.7}$$

Under minor conditions, we can reformulate the revenue maximization problem above into our theoretical framework. The proof details are deferred in Section A.6.1.

**Lemma 3.5.1.** *Let*[3] $\xi_t^v := k_t^v - \sum_{u\in N_v^1\setminus v} |\gamma_t^{(u,v)}| - \frac{b_t^v + b_{t+1}^v}{2}$ *for* $v \in \mathcal{V}$ *and* $1 \le t \le H$. *Suppose there exists* $\mu > 0$, $\ell_f > 0$, $b > 0$ *and* $\gamma > 0$, *s.t.* $\inf_{t,v} \xi_t^v \ge \mu/2 > 0$, $\sup_{v\in\mathcal{V},t\in H} k_t^v \le \ell_f/2$, $\sup_{v\in\mathcal{V},t\in H} b_t^v \le b$, *and* $\sup_{(u,v)\in\mathcal{E},t\in H} |\eta_t^{(u\to v)}| \le \gamma$. *Then, Equation* (3.7) *has the same global optimization solution as the following problem:*

$$\begin{aligned} \min \quad & \sum_{t=1}^{H}\sum_{v\in\mathcal{V}} f_t^v(x_t^v) + c_t^v(x_t^v, x_{t-1}^v) + \sum_{t=1}^{H}\sum_{e\in\mathcal{E}} s_t^e(x_t^u, x_t^v) \\ \textit{s.t.} \quad & 0 \le x_t^v \le \overline{p}_t^v \end{aligned} \tag{3.8}$$

*where*

$$\begin{aligned} f_t^v(x_t^v) &:= \xi_t^v \left(x_t^v - \frac{a_t^v}{2\xi_t^v}\right)^2, \\ s_t^{(u,v)}(x_t^u, x_t^v) &:= |\gamma_t^{(u,v)}| \left(x_t^u - sgn\left(\gamma_t^{(u,v)}\right)\cdot x_t^v\right)^2, \\ c_t^v(x_t^v, x_{t-1}^v) &:= \frac{b_t^v}{2}\left(x_t^v - x_{t-1}^v\right)^2. \end{aligned}$$

In the cost minimization formulation, the node cost function $f_t^v(x_t^v)$ is nonnegative, $\mu$-strongly convex, and $\ell_f$-smooth; the spatial interaction function is nonnegative, convex and $(4\gamma)$-smooth; the temporal interaction function is nonegative, convex and $(2b)$-smooth.

In the remainder of this section, we discuss the performance and practical benefits of LPC

[3] $b_{H+1}^v := 0$ for any node $v$

algorithms in the multiproduct pricing application. The decentralized nature of LPC is important in this setting. Interpretable pricing algorithms (Biggs, Sun, and Ettl, 2021) are attractive in practice. Our local pricing algorithm is indeed interpretable since the current price of a given product is transparently determined by reliable predictions of demand in the near future as well as interactions with directly related products. In addition, LPC needs less computation effort when exactly solving the global multiproduct pricing problem is computationally challenging in practice. Specifically, large online e-commerce companies maintain millions of products, which makes the entire network difficult to store, let alone do computation over. Moreover, due to the ease of changing prices, e-commerce companies often use dynamic pricing and change prices on a weekly (or quicker) basis, which magnifies the computational burden.

### 3.5.1 Competitive Bound

We first state an immediate corollary of Theorem 3.3.3 and Lemma A.6.1. This result establishes the competitive bound in the revenue maximization setting of the product networks. We defer the details of Lemma A.6.1 in Section A.6.2.

**Corollary 3.5.1.** *Define* $\widetilde{b} \coloneqq \sup_{(u,v)\in\mathcal{E},t\in[H]} a_t^u/a_t^v$, $\widetilde{c} \coloneqq \sup_{v\in\mathcal{V},t\in[H]} \frac{a_t^v}{p_t^v}$. *Under the same conditions as stated in Theorem 3.3.3, the competitive ratio of LPC for the revenue maximization problem is at least*

$$1 - \frac{\eta}{2}(O\left(h(r)^2 \cdot \rho_S^r\right) + O\left(C_3(r)^2 \cdot \rho_T^k\right)),$$

*where* $\Delta$ *denotes the degree of the product network and* $\eta \coloneqq \max\{2(\ell_f + \Delta\widetilde{b}\gamma)/\mu, \widetilde{c}/\mu\}$.

### 3.5.2 Numerical study

In this section, we illustrate the performance of our algorithm in a practical setting. We introduce our experiment setup, including the construction of a product graph, parameter regimes of the demand function, several candidate pricing policies, and their performance measured in the relative ratios (defined as the revenue achieved by a given algorithm divided by the global optimal revenue).

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| degree | 2202.0 | 6.1 | 11.8 | 1.0 | 1.0 | 3.0 | 6.0 | 99.0 |

Table 3.1: Degree quantiles of the product graph.

**Product graph**

We construct product graphs based on the previous work in McAuley, Pandey, and Leskovec (2015). Specifically, we utilize their results for popular products in the Baby clothes category on the Amazon.com website. Methodologically, they build topic models trained to automatically discover topics from the text (for example, product reviews and descriptions) that are successful at predicting and explaining product relationships. They also incorporate other features such as rating, product specifications, prices, and brands. For each given product in the Baby clothes category, McAuley, Pandey, and Leskovec (2015) outputs its top 25 complements, and top 25 substitutes, ranked from the highest likelihood to the lowest likelihood. Since the product graph is deterministic in our model framework, we set a threshold of $-0.1$ for the likelihood scores to translate from the probabilistic product relationship in McAuley, Pandey, and Leskovec (2015). If the score is great than or equal to $-0.1$, we add the corresponding directed edge with its edge type (i.e., complementary or substitutable) to the product graph. Then, we remove nodes [4] with degrees more than 100 from the graph and further remove the isolated nodes after the previous removal. Finally, we obtain a product graph with 2202 nodes and 6680 (directed) edges. The quantiles of degree distribution are presented in Table 3.1.

**Demand functions**

We consider a dynamic pricing problem with a horizon of six weeks i.e. $H = 6$. During each week, the intercept for the demand function, $a_t^v$, is generated from a uniform distribution in the

[4]Two reasons are driving this decision. First, when the degree is greater than 100, it is greater than 95% percentiles of the degree distribution. We view those nodes as outliers from the typical degree values and should be handled separately. Second, for our problem remains solvable (or converge in a reasonable time) for Gurobi, we need to maintain a certain level of convexity for the node cost functions. For nodes with degrees greater than 100, the node cost functions are not convex.

interval $[1, 9]$. We consider the feasible price interval to be $[0, 5]$ and the slope $k_t^v := k^v$ is sampled from a uniform distribution in $[0.95, 1.05]$ for [5] all $t \in [H]$ and $v \in V$. We consider the cross-product interaction strengths $c$ to be bounded above by $0.09$. Then, for each node, we simulate the cross-product sensitivity coefficients $\eta_t^{u \to v}$ from a uniform distribution $[\frac{c}{2}, c]$ if there is a complementary edge from $u$ to $v$, or from a uniform distribution $[-c, -\frac{c}{2}]$ if there is a substitutable edge from $u$ to $v$. Similarly, we simulate cross-period sensitivity coefficients $b_t^v := b^v$ from a uniform distribution $[b - 0.01, b]$ for all $t \in [H]$ and $v \in V$ where we consider $b \in \{0.20, 0.30\}$. We further reduce this directed graph to an undirected graph and redefine the cross-product price sensitivities on the undirected edges using the formula where $e := (u, v)$:

$$\gamma^e := \frac{1}{2} \left( \eta^{(u \to v)} + \eta^{(v \to u)} \right).$$

In the corresponding undirected graph, the number of nodes is 2202 and the number of edges is 6637. $81\%$ of the edges are complementary type while $19\%$ of the edges are substitutable type. This difference is since the original inference model in (McAuley, Pandey, and Leskovec, 2015) has greater confidence (and also better accuracy) in predicting the complements.

**Results**

We derive the closed-form solutions for four different pricing policies and compare their performance against the optimal pricing policy which can be solved as a nonlinear programming problem via Gurobi (Gurobi Optimization, LLC, 2022).

*Naive policy.* A naive solution often used in practice is to decide prices based only on the elasticity of the product itself. In this case, the pricing policy is to solve the reformulated optimization problem with only the node cost $\sum_{t \in [H], v \in V} \xi_t^v (x_t^v - \frac{a_t^v}{2\xi_t^v})^2$, which gives

$$\text{naive}_t^v = \frac{a_t^v}{2\xi_t^v}$$

[5] With an average of the base demand (i.e., the intercept) 5, slope 1, the average actual demand for each product is nonnegative on the $[0, 5]$ price interval.

for each product $v$ and time $t$. The total revenue under this naive pricing policy is

$$Rev_{\text{naive}} = \sum_{t=1}^{H} \text{naive}_t^v \cdot d_t^v(\text{naive}),$$

where $d_t^v(\text{naive})$ is the demand for product $v$ at time $t$ when implementing the naive policy.

*Lookahead policy.* We consider a policy that uses the predicted future price of each product but still ignores the network effects. Specifically, we use a lookahead policy with $k = 2$. In the local problem, the price of the product's future week is set in a greedy way as $\frac{a_{t+1}^v}{2\xi_{t+1}^v}$. For any node $v \in V$ and any $t \leq H$,

$$\begin{aligned}\text{lookahead}_t^v &= \arg\min_x \{\xi_t^v(x - \frac{a_t^v}{2\xi_t^v})^2 + \frac{b_t^v}{2}(x - x_{t-1}^v)^2 + \frac{b_{t+1}^v}{2}(x - \frac{a_{t+1}^v}{2\xi_t^v})^2\} \\ &= \frac{a_t^v + b_t^v x_{t-1}^v + b_{t+1}^v a_{t+1}^v/(2\xi^v)}{2\xi_t^v + b_t^v + b_{t+1}^v}.\end{aligned}$$

The total revenue under this lookahead pricing policy is

$$Rev_{\text{lookahead}} = \sum_{t=1}^{H} \text{lookahead}_t^v \cdot d_t^v(\text{lookahead}),$$

where $d_t^v(\text{lookahead})$ is the demand for product $v$ at time $t$ when implementing the lookahead policy.

*Network pricing policy.* Another heuristic is to consider the network effects but ignores the temporal interactions. Specifically, we use the network policy with $r = 1$. In the local problem, the price of the focal node's spatial neighbors is set greedily to be $\frac{a_t^u}{2\xi_t^u}$ for such $u$'s. For any $t \leq H$,

$$\begin{aligned}\text{network}_t^v &= \arg\min_x \{\xi_t^v(x - \frac{a_t^v}{2\xi_t^v})^2 + \sum_{(u,v)\in\mathcal{E}} |\gamma_t^{(u,v)}|(\frac{a_t^u}{2\xi_t^u} - sgn(\gamma_t^{(u,v)})x)^2\} \\ &= \frac{a_t^v + \sum_{(u,v)\in\mathcal{E}} \gamma_t^{(u,v)} \frac{a_t^u}{\xi_t^u}}{2\xi_t^v + 2\sum_{(u,v)\in\mathcal{E}} |\gamma_t^{(u,v)}|}\end{aligned}$$

for each product $v$. The total revenue under the network pricing policy is

$$Rev_{\text{network}} = \sum_{t=1}^{H} \text{network}_t^v d_t^v(\text{network}),$$

where $d_t^v(\text{network})$ is the demand for product $v$ at time $t$ when implementing the network policy.

*LPC pricing policy.* We apply our LPC algorithm with $r = 1$ and $k = 2$ in this multiproduct pricing setting. Suppose our focal node is $v$, then the prices of its spatial neighbors are fixed as $\frac{a_t^u}{2\xi_t^u}$ and the price of its future week is set to be $\frac{a_{t+1}^v}{2\xi_{t+1}^v}$ in the local problem. For $t \leq H$,

$$\begin{aligned}\text{LPC}_t^v &= \arg\min_x \xi_t^v(x - \frac{a_t^v}{2\xi_t^v})^2 + \sum_{(u,v)\in\mathcal{E}} |\gamma_t^{(u,v)}|(\frac{a_t^u}{2\xi_t^u} - sgn(\gamma_t^{(u,v)})x)^2 + \frac{b_t^v}{2}(x - x_{t-1}^v)^2 + \frac{b_{t+1}^v}{2}(x - \frac{a_{t+1}^v}{2\xi_{t+1}^v})^2, \\ &= \frac{a_t^v + \sum_{(u,v)\in\mathcal{E}} \gamma_t^{(u,v)} \frac{a_t^u}{\xi_t^u} + b_t^v x_{t-1}^v + b_{t+1}^v \frac{a_{t+1}^v}{2\xi^v}}{2\xi_t^v + 2\sum_{(u,v)\in\mathcal{E}} |\gamma_t^{(u,v)}| + b_t^v + b_{t+1}^v}\end{aligned}$$

for any node $v$. The total revenue under the LPC pricing policy is

$$Rev_{\text{LPC}} = \sum_{t=1}^{H} \text{LPC}_t^v d_t^v(\text{LPC}),$$

where $d_t^v(\text{LPC})$ is the demand for product $v$ at time $t$ when implementing the LPC policy.

*Optimal pricing policy.* Since the problem is quadratic, we can obtain its global optimal solution via a nonlinear programming formulation solved by Gurobi software. This global optimal solution gives us the offline global optimal revenue benchmark. Note that a requirement for this problem to be convex is that $k_t^v - \sum_{u\in N_v^1\setminus v} |\gamma_t^{(u,v)}| - \frac{b_t^v + b_{t+1}^v}{2} \geq 0$.

We present our results for varying strength of network interactions, denoted by $c$. $c$ varies from $0.01$ to $0.09$ with each $0.01$ as increment. Recall that the temporal interaction strength $b$ is selected from $\{0.20, 0.30\}$ throughout the experiments. We report the relative performance ratios of the heuristic policies against the offline global optimal benchmark in Figure 3.3. The performance ratio is calculated by the actual revenue under each given policy divided by the offline global optimal revenue. We also calculate the 95% confidence intervals for the performance ratios

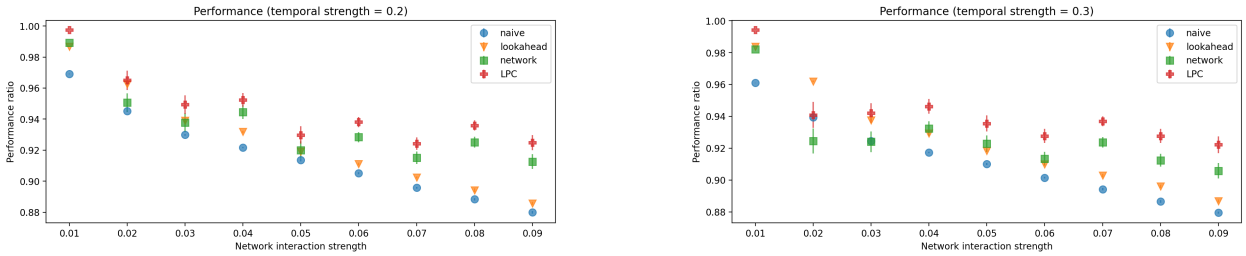


Figure 3.3: Performance ratios of different pricing policies.

through bootstrapped samples. We simulate 30 samples for each specified network strength $c$. We observe for each subgraph, the performance of the Naive and Lookahead pricing policy drops when the network interaction strength increases while the performance of the Network and LPC pricing policy is relatively stable [6]. We also observe that the performance gap between LPC and Network policy (similarly the performance gap between Lookahead and Naive policy) increases as the temporal interaction strength increases from $0.2$ to $0.3$ since the performance of Network and Naive policy degrades when the temporal interaction strength increases.

In addition, we also observe LPC is able to closely follow the pricing trajectory of the globally optimal policy even though it consumes much less information and enjoys a much faster implementation due to its convenient closed-form solution in the multiproduct pricing setting. LPC solves our problem in 0.31 seconds on average while the Gurobi solvers takes 85.24 seconds (the computation time is consistent across different simulated samples and parameter regimes. The details are presented in Figure A.2 of the Appendix). While the Lookahead policy smoothes out the base demand curve with smaller jumps in magnitude, the LPC policy balances between Lookahead and Network policies which can result in higher expected total revenue. We track price trajectories under different policies for a random set of products and present them in Figure A.3.

Finally, we add another set of experiments to verify the performance of the proposed policies when there exists a "prediction error" where instead of knowing the exact future intercept (i.e., $a_t^v$),

[6] At the network strength 0.02 and temporal strength 0.3, the average performance of LPC is worse than Lookahead policy. The potential reasons are two-fold: 1. at this particular parameter setup, the confidence interval (indicated by the errorbar) is significantly larger and we are less certain about comparing relative performances of different heuristics; 2. due to large temporal strength, we have used the nonconvex optimization solver in Gurobi. Although the global solution converges, we are less confident about the exactness of the solutions when the violation of convexity exists.

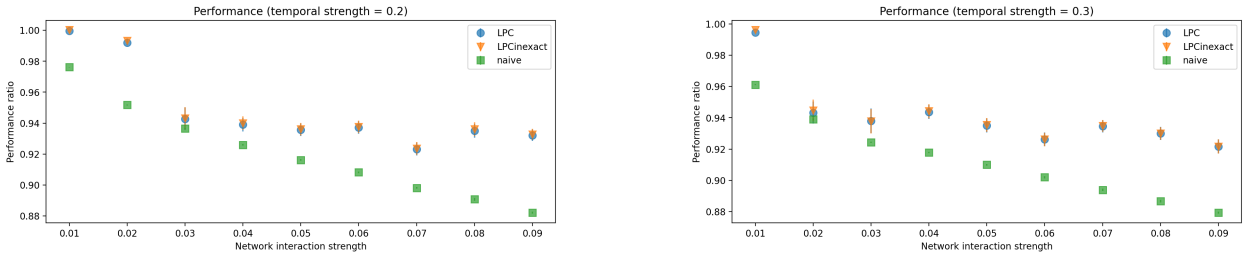


Figure 3.4: Performance ratios with exact/inexact predictions (LPC vs LPC-inexact).

we use its mean in the lookahead step.

*$LPC_{\text{inexact}}$ pricing policy.* To deal with the inexact prediction issue, we consider a policy called $\text{LPC}_{\text{inexact}}$ that uses a heuristic price $\frac{\overline{a}^v}{2\xi_{t+1}^v}$ where $\overline{a}^v$ is the expected value of $a_t^v$. In other words, we assume the decision-maker is able to estimate the mean of the intercept based on historical data. The prices of its spatial neighbors are still fixed as $\frac{a_t^u}{2\xi_t^u}$. For any node $v \in V$ and any $t \leq H$,

$$\begin{aligned}(\text{LPC}_{\text{inexact}})_t^v &= \arg\min_x \xi_t^v (x - \frac{a_t^v}{2\xi_t^v})^2 + \sum_{(u,v)\in\mathcal{E}} |\gamma_t^{(u,v)}| (\frac{a_t^u}{2\xi_t^u} - sgn(\gamma_t^{(u,v)})x)^2 \\ &+ \frac{b_t^v}{2}(x - x_{t-1}^v)^2 + \frac{b_{t+1}^v}{2}(x - \frac{\overline{a}^v}{2\xi_t^v})^2, \\ &= \frac{a_t^v + \sum_{(u,v)\in\mathcal{E}} \gamma_t^{(u,v)} \frac{a_t^u}{\xi_t^u} + b_t^v x_{t-1}^v + b_{t+1}^v \frac{\overline{a}^v}{2\xi_t^v}}{2\xi_t^v + 2\sum_{(u,v)\in\mathcal{E}} |\gamma_t^{(u,v)}| + b_t^v + b_{t+1}^v}.\end{aligned}$$

The total revenue under the $\text{LPC}_{\text{inexact}}$ pricing policy is

$$Rev_{\text{LPC}_{\text{inexact}}} = \sum_{t=1}^{H} (\text{LPC}_{\text{inexact}})_t^v d_t^v (\text{LPC}_{\text{inexact}}),$$

where $d_t^v(\text{LPC}_{\text{inexact}})$ is the demand for product $v$ at time $t$ when implementing the $\text{LPC}_{\text{inexact}}$ policy. In Figure 3.4, we observe no practical performance degradation from the inexact predictions.

## 3.6 Concluding Remarks

In this work, we introduce and study a novel form of decentralized online convex optimization in a networked system, where the local actions of each agent are coupled by temporal interactions

and spatial interactions. We propose a decentralized online algorithm, LPC, which leverages all available information within a prediction horizon of length $k$ and a communication radius of $r$ to achieve a competitive ratio of $1 + \widetilde{O}(\rho_T^k) + \widetilde{O}(\rho_S^r)$. Our lower bound result shows that this competitive ratio is order optimal. Our results imply that the two types of resources, the prediction horizon and the communication radius, must be improved *simultaneously* in order to obtain a competitive ratio that converges to $1$. That is, it is not enough to either have a large communication radius or a long prediction horizon, the combination of both is necessary to approach the hindsight optimal performance.

# Chapter 4: Dynamic matchmaking on gaming platforms

## 4.1 Introduction

The online video gaming industry has consistently been expanding in the recent decade. The global market is estimated to be worth $268.8 billion U.S. dollars annually in 2025, up from $178 billion U.S. dollars in 2021. Currently, there are an estimated 1 billion online gamers worldwide. In 2025, online gaming audiences are projected to surpass 1.3 billion [1]. Among the entire market of online video games, one large sector is online competitive games, where players play against one another (one-on-one or in teams). For competitive games, players often have a rating score that measures their skill attribute in this game. Players can improve their scores gradually by winning games and the increase in the rating depends on the other opponent's rating as well. For example, winning a game where the opponent has a poor rating does not increase the winning player's rating much. Players' ratings can decrease as well if losing a game. Managing users' engagement is crucial to gaming platforms, especially to free-to-play video games since the majority of revenue (the sales of digital products and advertising) is directly from the daily active users (DAU). In this work, we consider a general dynamic match-making model to improve players' engagement, specifically through balancing players' average waiting time and matching quality. Moreover, we focus on understanding a popular matching heuristic in practice - the bubble algorithm, establish the theoretical guarantees of this heuristic under our model and provide insights for practitioners on tuning the parameter of this algorithm to achieve better engagement measurement.

**Related Work.** Our proposed work is related to recent studies on different match-making algorithms to improve players' engagement with online gaming platforms. Chen et al. (2017) consider a static model where the objective is to minimize players' churn risk. They assume all players

[1] statistics source: `https://www.statista.com/topics/1551/online-gaming/`

have arrived and then define a complete graph connecting every pair of nodes. The edge weights are defined as the probabilities of churning for either of the connected players. Huang, Jasin, and Manchanda (2019) propose a two-stage procedure where they first estimate user engagement state (high, medium, low) with a hidden Markov model and then propose a heuristic matching algorithm that leverages the engagement state information of players. Chen, Elmachtoub, and Lei (2021) focus on a geometric losing streak model where players have a fixed probability to churn after each consecutive loss in the game. In contrast, our proposal aims to understand the long-run behavior of the players incorporating both match-making quality as well as waiting times. Our model also enables us to analytically obtain the performance guarantee of the Bubble algorithm compared to the offline optimal benchmark.

Our proposed work also contributes to the literature on the general dynamic/online matching problem where the same trade-off is studied between matching available pairs now versus waiting for (potentially) better matches in the future. Ashlagi et al. (2019) provide an approximation matching algorithm when customers arrive online and stay for a fixed number of periods before leaving. Aouad and Saritaç (2020) propose approximation algorithms for dynamic matching over edge-weighted graphs, where customers arrive and leave the platform following a stochastic process. In contrast, we consider the setting where the players arrive stochastically and stay in the platform until matched and penalize the waiting time by imposing a waiting cost, and the edge weight (i.e., matching cost) is related to the ratings of matched players, both of which costs are estimated directly from our collected real-world dataset.

### 4.2 Model

In this section, we introduce our dynamic matching model for 1-versus-1 online competitive games. We study a continuous-time model, where players arrive according to a Poisson process with a rate $\lambda$. We consider a fairly long time horizon $T$ and index the players by their arriving sequence. Denote the time when the $i$-th player arrives as $J_i \in (0, T]$ for $i = 1, 2, \cdots$. $N_t$ denotes the number of players arrived by time $t$ for $t \in (0, T]$. After arriving, players stay in the system

until they get matched to an opponent or a bot. Once matched, the matched pair leaves the system.

The skill attribute of each player is measured by Elo rating. We denote the skill for $i$-th arriving player as $X_i$. We assume $\{X_i\}_{i\geq 1}$ are i.i.d independent. The distribution of $X_i$ is known but we observe its realization at time $J_i$. We consider a parametric class of matching cost and waiting cost functions. The matching cost is a power of the Elo difference for the matched pair, i.e., $f(X_i, X_j) = |X_i - X_j|^{k_M}$ for some $k_M > 0$ if the $i$-th player is matched to the $j$-th player [2]. For any $i = 1, \ldots, N_T$, we denote the time waited by $i$-th player before getting matched as $W_i$. The waiting cost is a power of the waiting time: $g(W_i) = W_i^{k_W}$ for some $k_W > 0$. Although the class of power law cost functions defined on $x \in [0, +\infty)$ are simple, it contains a broad spectrum of possibilities on how players react to waiting time or matching quality, for example, it covers linear cost functions when $k = 1$; convex cost functions when $k > 1$; concave cost function when $k < 1$. Notably, when $k \to 0$, it models the cost function where the penalty is zero when there is no waiting time or mismatch and a constant cost otherwise; when $k \to +\infty$, it models the cost function where the penalty exists only when the waiting time or mismatch is above a threshold of 1 and 0 otherwise. The relative weight between matching cost and waiting cost is controlled by another model primitive $c \in \mathbb{R}_+$.

To define non-anticipate matching policies, we denote $\{\mathcal{F}_t\}_{t\geq 0}$ as the canonical filtration generated by the Poisson arrival process, current waiting times and realized Elo ratings of arrived players, i.e., $\mathcal{F}_t = \sigma(N_{t'}, W_i, X_i : t' \leq t, J_i \leq t)$. A matching policy $\pi$ describes an $\mathcal{F}_t$-adapted stochastic process $\{M_t^\pi\}_{t\geq 0}$ where $M_t^\pi$ is a matching made at time $t$ within arrived players (including players arriving at time $t$). We let $M^\pi := \cup_{0\leq t\leq T} M_t^\pi$ denote all matchings during time $[0, T]$. The objective is to minimize the average waiting and matching cost per player in the system. Note that we assume an even number of players such that all players can be matched. The average cost per player of policy $\pi$ is given by

$$C^\pi = \limsup_{T\to\infty} \frac{1}{N_T} \left[ \sum_{k:J_k\leq T} g(W_k) + c \cdot \sum_{i,j:(i,j)\in M^\pi} \frac{1}{2} f(X_i, X_j) \right]. \tag{4.1}$$

[2]For player $i$ or player $j$, the matching cost is $\frac{1}{2} f(X_i, X_j)$, which adds up to the matching cost for this pair.

## 4.3 Bubble algorithm

In this section, we introduce a popular heuristic that is actively used in the gaming industry to trade off between waiting cost and matching cost and show its theoretical performance guarantee under our model assumption. A bubble algorithm operates as follows: every player is placed on the Elo line once arrived; then the algorithm grows a bubble around the player which expands over time; when two bubbles touch, the two players get matched and leave the system. We denote the bubble expansion rate as $\alpha$ and the resulting matching as $M^{\alpha}$. Larger values of $\alpha$ lead to a shorter waiting time for players but potentially worse matching quality. We illustrate this trade-off via a simple example below.

*Example 1.* Consider four players arrive sequentially at times 1, 2, 3, 4 and their Elo ratings are the following: $X_1 = 500$, $X_2 = 700$, $X_3 = 550$ and $X_4 = 750$. A large $\alpha$ (e.g. $\alpha_H = 200$ Elo per time unit) will match players 1 and 2 once player 2 arrives, and then match players 3 and 4; while a small $\alpha$ (e.g. $\alpha_L = 50$ Elo per time unit) will match player 1 and 3 once player 3 arrives, and then player 2 and 4. The waiting time under $\alpha_H$ is 1 time unit for players 1 and 3, 0 waiting time for players 2 and 4; the waiting time under $\alpha_L$ is 2 time units for players 1 and 2, 0 waiting time for player 3 and 4. The average matching distance under $\alpha_H$ is 200 while the average matching distance under $\alpha_L$ is 50.

In the remainder of this section, we characterize how one selects the optimal expansion rate $\alpha$ to trade off the matching and waiting costs under different model primitives. Our first lemma states an important property of the bubble algorithm relating the matching distance to the waiting times of any matched pair: *the distance between two matched players is upper bounded by bubble expansion rate $\alpha$ multiplied with the sum of waiting times of the two players.*

**Lemma 4.3.1.** *Suppose the $i$-th arriving player is matched with the $j$-th arriving player by a Bubble algorithm with rate $\alpha$. Then,*

$$|X_i - X_j| \leq \alpha(W_i + W_j)$$

*where $X_i$ (resp. $X_j$) is the Elo rating of player $i$ and player $j$, and $W_i$ (resp. $W_j$) is the waiting time of player $i$ (resp. player $j$) on the platform.*

*Proof of Lemma 4.3.1.* W.l.o.g, we assume player $i$ arrives earlier than player $j$. Once player $i$ arrives, the bubble around him/her starts to expand at the rate of $\alpha$. Since player $i$ matches with player $j$, player $j$ must be the first player that $i$'s bubble touches. There are two possible cases.

- Case 1: player $j$ arrives within player $i$'s bubble. Then their matching distance is bounded by the the bubble radius after waiting for $W_i$ time, i.e., $|X_i - X_j| \leq \alpha W_i$.
- Case 2: player $j$ arrives outside player $i$'s bubble. Then player $j$ starts to grow his/her bubble and two bubbles touch. Their matching distance is bounded by the summation of two bubbles' radii, i.e., $|X_i - X_j| \leq \alpha W_i + \alpha W_j$. Q.E.D.

When $\alpha$ is arbitrarily close to $0$, the corresponding bubble algorithm matches players as if using offline greedy policy; when $\alpha$ approaches infinity, the corresponding bubble algorithm matches players as if using online greedy policy. Although the bubble algorithm is greedy in nature, with an appropriate expansion rate $\alpha$, it still performs well in practice. In Section 4.3.1 and Section 4.3.2, we establish the theoretical performance guarantee for bubble algorithms under our model.

### 4.3.1 Lower bound on the average cost

We develop a lower bound on the long-term average cost per player. Recall that $k_M$(resp. $k_M$) is the power of matching cost function $f(\cdot, \cdot)$ (resp. waiting cost function $g(\cdot)$), and $c$ is the relative weight between waiting cost and matching cost.

**Theorem 4.3.1.** *Assume players arrive according to a Poisson process with rate $\lambda$ and Elo ratings are independent uniform distribution rescaled between $[0, 1]$. We further restrict to policies that have steady-state distributions. The long-run average cost per player under any such policy $\pi$, has the following lower bound in terms of arrival rate and model parameter $c$:*

$$\inf_{\pi} C^{\pi} = \Omega(\lambda^{-\frac{k_W k_M}{k_W + k_M}} c^{\frac{k_W}{k_W + k_M}}).$$

*Proof of Theorem 4.3.1.* $X$ (resp. $Y$) denotes the Elo rating difference (resp. arrival time difference) of matched pairs under a given policy $\pi$ in the steady state. Then, we define the following random variable,

$$H \coloneqq \frac{c}{2} \cdot |X|^{k_M} + |Y|^{k_W}.$$

The long-run average cost is lower bounded by the expectation of $H$, i.e.,

$$C^\pi \geq \mathbb{E}[H] = \int_0^{+\infty} \mathbb{P}(H > b) db.$$

We represent an arrival by a point on the 2-D plane, where the x-axis denotes the player's rating $x$ and the y-axis denotes the arrival time $t$. The distance between any two arrival $(x_1, t_1)$, $(x_2, t_2)$ are defined as follows:

$$\text{dist}((x_1, t_1), (x_2, t_2)) = \frac{c}{2}|x_1 - x_2|^{k_M} + |t_1 - t_2|^{k_W}.$$

The event of $H > b$ is equivalent to the event that there is no other arrival within a total distance of $b$ to the current arrival. Then, we have

$$\mathbb{P}(H > b) = \mathbb{P}(\text{no another arrival within cost } b) \geq e^{-\lambda A(b)}$$

where $A(b)$ is the area of $\{(x, y) : \frac{c}{2}|x|^{k_M} + |y|^{k_W} \leq b\}$. Recall that $\lambda$ is the arrival rate of the Poisson process. The last equality is since the arrival process can be viewed as a Poisson process with the uniform rate $\lambda$ on $[0, 1] \times \mathbb{R}_+$. Moreover, $A(b) \subset \{(x, y) : \frac{c}{2}|x|^{k_M} \leq b, |y|^{k_W} \leq b\}$. Therefore,

$$A(b) \leq 4b^{\frac{1}{k_W}} (\frac{2b}{c})^{\frac{1}{k_M}} = 4(\frac{c}{2})^{-\frac{1}{k_M}} b^{\frac{1}{k_W} + \frac{1}{k_M}}.$$

The long-run average cost is lower bounded by the following:

$$\begin{aligned} C^{\pi} &\geq \int_0^{+\infty} e^{-4\lambda(\frac{c}{2})^{-\frac{1}{k_M}} b^{\frac{1}{k_W}+\frac{1}{k_M}}} db \\ &= \int_0^{+\infty} e^{-\widetilde{\lambda} b^k} db \end{aligned}$$

where $\widetilde{\lambda} = 4\lambda(\frac{c}{2})^{-\frac{1}{k_M}}$ and $k = \frac{1}{k_W} + \frac{1}{k_M}$. Moreover,

$$\begin{aligned} \int_0^{+\infty} e^{-\widetilde{\lambda} b^k} db &= \frac{1}{k} \int_0^{+\infty} e^{-\widetilde{\lambda} x} x^{\frac{1}{k}-1} dx \\ &= \frac{1}{k} (\frac{1}{\widetilde{\lambda}})^{\frac{1}{k}} \int_0^{+\infty} e^{-t} t^{\frac{1}{k}-1} dt \\ &= \frac{1}{k} (\frac{1}{\widetilde{\lambda}})^{\frac{1}{k}} \Gamma(\frac{1}{k}) \\ &= (\frac{1}{\widetilde{\lambda}})^{\frac{1}{k}} \Gamma(\frac{1}{k}+1) \end{aligned}$$

where the first equality is by a change of variable $x = b^k$, the second equality is also by a change of variable $t = \widetilde{\lambda} x$, the third equality is by definition of Gamma functions, and the last inequality is by property $\Gamma(x+1) = x\Gamma(x)$.

$$\begin{aligned} C^{\pi} &\geq (\frac{1}{\widetilde{\lambda}})^{\frac{1}{k}} \Gamma(\frac{1}{k}+1) \\ &= (4\lambda)^{-\frac{k_W k_M}{k_W+k_M}} (\frac{c}{2})^{\frac{k_W}{k_W+k_M}} \Gamma(\frac{k_W k_M}{k_W+k_M}+1) \\ &= \Omega(\lambda^{-\frac{k_W k_M}{k_W+k_M}} c^{\frac{k_W}{k_W+k_M}}). \end{aligned}$$

### 4.3.2 Performance bound for Bubble algorithm

We further show that Bubble algorithm with the optimal expansion rate achieves a cost within a constant factor compared to $\inf_{\pi} C^{\pi}$.

**Theorem 4.3.2.** *Under the same modeling assumptions as in Theorem 4.3.1, with the optimal bubble expansion rate* $\alpha^* = c^{-\frac{2}{k_W+k_M}} \lambda^{\frac{k_M-k_W}{k_W+k_M}} \widetilde{c}$ *where* $\widetilde{c}$ *is a constant depending only on* $k_W$ *and*

*$k_M$, the long-run average cost per player under bubble algorithm has the following upper bound:*

$$C^{\text{Bubble}} = O(\lambda^{-\frac{k_M k_W}{k_M+k_W}} c^{\frac{k_W}{k_W+k_W}}).$$

To establish Theorem 4.3.2, we use the following lemma to bound the tail distribution of waiting time for any given player.

**Lemma 4.3.2.** *Given a Bubble algorithm with expansion rate $\alpha$, under the steady state, a player's waiting time $W$ has the following probabilistic bound, for a given $x \geq 0$,*

$$\mathbb{P}(W > x) \leq \exp(-\lambda\alpha x^2).$$

*Proof of Lemma 4.3.2.* $W$ is upper-bounded by the time it takes for some future player to land within the current player's growing bubble. The bubble around a given player starts to expand with rate $\alpha$ once this player arrives. Given any $\epsilon > 0$, divide the time $0 = t_0 < t_1 < \cdots < t_n = x$ such that $\sup_i |t_{i+1} - t_i| \leq \epsilon$. During time $(t_i, t_{i+1}]$, the bubble contains at least a length $2\lambda\alpha t_i$ interval on the Elo line.

$$\begin{aligned}\mathbb{P}(\text{no arrivals in } (t_i, t_{i+1}]) &\leq \mathbb{P}(\text{no arrivals in } (t_i, t_{i+1}] \text{ for a Poiss}(2\lambda\alpha t_i) \text{ process}) \\ &= \exp(-2\lambda\alpha t_i(t_{i+1} - t_i)).\end{aligned}$$

Since the Poisson process has independent increments,

$$\mathbb{P}(\text{no arrivals in } (0, t]) \leq \prod_{i=0}^{n-1} \exp(-2\lambda\alpha t_i(t_{i+1} - t_i)) = \exp(-2\lambda\alpha \int_0^x t dt) = \exp(-\lambda\alpha x^2).$$

*Proof of Thereom 4.3.2.* Using Lemma 4.3.2, we have the following bound on the expectation of

the waiting cost, $\mathbb{E}[g(W)]$,

$$\begin{aligned}
\mathbb{E}[g(W)] &= \int_{\mathbb{R}_+} g(x) dF_W(x) = \int_{\mathbb{R}_+} \int_0^x g'(y) dy dF_W(x) \\
&= \int_0^{+\infty} g'(y) \int_y^{+\infty} dF_W(x) dy = \int_0^{+\infty} g'(y) \mathbb{P}(W > y) dy \\
&\leq \int_0^{+\infty} g'(y) e^{-\lambda\alpha y^2} dy \\
&= k_W \int_0^{+\infty} y^{k_W - 1} e^{-\lambda\alpha y^2} dy \\
&= k_W \int_0^{+\infty} (\frac{t}{\lambda\alpha})^{\frac{k_W-1}{2}} e^{-t} \frac{1}{\sqrt{\lambda\alpha}} dt^{\frac{1}{2}} \\
&= (\frac{1}{\lambda\alpha})^{\frac{k_W}{2}} \Gamma(\frac{k_W}{2} + 1)
\end{aligned} \tag{4.2}$$

where the third equality is by Fubini–Tonelli theorem for non-negative functions, the first inequliaty is by Lemma 4.3.2, the second last equality is by a change of variable $t = \lambda\alpha y^2$, and the last equality is by the definition of Gamma functions.

Consider the matching distance in the steady state between two matched players $i$ and $j$. Then, by the property of bubble algorithm in Lemma 4.3.1,

$$|X_i - X_j| \leq \alpha(W_i + W_j).$$

We further bound the matching cost by a function of the waiting time.

$$\begin{aligned}
\mathbb{E}[f(X_i, X_j)] &= \mathbb{E}[|X_i - X_j|^{k_M}] \\
&\leq \alpha^{k_M} \mathbb{E}[(W_i + W_j)^{k_M}] \\
&\leq \alpha^{k_M} \max\{1, 2^{k_M - 1}\} (\mathbb{E}[W_i^{k_M}] + \mathbb{E}[W_j^{k_M}]) \\
&\leq 2\alpha^{k_M} \max\{1, 2^{k_M - 1}\} (\frac{1}{\lambda\alpha})^{\frac{k_M}{2}} \Gamma(\frac{k_M}{2} + 1) \\
&= 2(\frac{\alpha}{\lambda})^{\frac{k_M}{2}} \max\{1, 2^{k_M - 1}\} \Gamma(\frac{k_M}{2} + 1)
\end{aligned}$$

where the second inequality is by Jensen's inequality for $k_M \geq 1$ and by first principles for $0 <$

$k_M < 1$; the third inequality is by (4.2).

Given a bubble expansion rate $\alpha$, let $C^\alpha$ denote the corresponding long-run average cost per player. Then,

$$C^\alpha \leq (\frac{1}{\lambda\alpha})^{\frac{k_W}{2}}\Gamma(\frac{k_W}{2}+1) + c(\frac{\alpha}{\lambda})^{\frac{k_M}{2}}\max\{1, 2^{k_M-1}\}\Gamma(\frac{k_M}{2}+1),$$

where we observe that the matching cost is increasing with $\alpha$ and the waiting cost is decreasing with $\alpha$.

$$\begin{aligned} C^{\text{Bubble}} &\leq \inf_\alpha \left\{ (\frac{1}{\lambda\alpha})^{\frac{k_W}{2}}\Gamma(\frac{k_W}{2}+1) + c(\frac{\alpha}{\lambda})^{\frac{k_M}{2}}\max\{1, 2^{k_M-1}\}\Gamma(\frac{k_M}{2}+1) \right\} \\ &= c^{\frac{k_W}{k_W+k_W}}\lambda^{-\frac{k_M k_W}{k_M+k_W}}(\widetilde{c}^{\frac{k_M}{2}}c_1 + \frac{c_2}{\widetilde{c}^{\frac{k_W}{2}}}) \end{aligned}$$

where $c_1 \coloneqq \max\{1, 2^{k_M-1}\}\Gamma(\frac{k_M}{2}+1)$, $c_2 \coloneqq \Gamma(\frac{k_W}{2}+1)$, and $\widetilde{c} \coloneqq (\frac{c_2 k_W}{c_1 k_M})^{\frac{2}{k_W+k_M}}$. The optimal $\alpha^* = c^{-\frac{2}{k_W+k_M}}\lambda^{\frac{k_M-k_W}{k_M+k_W}}\widetilde{c}$. Q.E.D.

### 4.4 Numerical studies

In this section, we demonstrate how a company could use the bubble algorithm to increase the engagement of players. We use data from a start-up gaming company Prismata with relatively a smaller active user base. On the Prismata platform, the waiting time for matching with another player is usually not negligible, and balancing between waiting time and matching quality is a key question for players' engagement.

**Prismata Dataset.** We use the Prismata dataset from $2020/01/18$ to $2021/03/05$. On the Prismata platform, players hit *start_automatch* when they enter the platform and look for matches; after waiting some time, players either get matched to an opponent or bot or leave the platform. We also have additional information on departures by looking into the *cancel_automatch* activities. A player can also leave without hitting the button of *cancel_automatch*, however, in this case, the departure is not logged in the dataset. From our historical data, we have information about

the arrivals and departures: 1) the time stamp when players enter the platform; 2) the time stamp when some players get matched; 3) the time stamp when some players hit *cancal_automatch*. In summary, there are 230,217 logged arrivals within the period. The departure rate is 11.0% among all arrivals, and the overall matched rate is 89.0%, including 38.5% are matched with a bot and 50.5% are matched with an actual player.

### 4.4.1 Estimation of system parameters

In this section, we estimate system parameters: arrival rate, waiting power, matching power, and relative weighting between waiting cost and matching cost under our model.

**Estimation of arrival rate.**

We first estimate the average arrivals per minute at different hours of the day. In Table 4.1, we present the arrival rate from hour 0 to hour 23 in UTC.

We further bucket the arrival process into High and Low arrival rate regimes, specifically UTC hour $0-3$ and $17-21$ as the High arrival rate regime, and $4-16$ and $22-23$ as the Low arrival rate regime. The user base is mostly in North America, and hence the observed high arrival regime is during afternoons and evenings in players' timezone. The average arrival rate per minute is summarized in Table 4.2.

**Estimation of waiting power**

The waiting time and matching distance are often correlated since they are both (implicitly) decided by the current matching policy on the Prismata platform. The endogeneity makes it difficult to disentangle the effect of waiting time from the matching quality on players' engagement. However, one nice characteristic of our dataset helps us separate the effect of waiting time and matching quality: players only know the quality of their matching (i.e., rating difference between players) once matched. Therefore, the departed players are not departed due to the current matching quality.

Next, we explain our procedure for estimating the waiting power in the cost function via maxi-

| hour | total arrivals | arrival per minute |
|---|---|---|
| 0 | 14851 | 0.60 |
| 1 | 12072 | 0.49 |
| 2 | 10388 | 0.42 |
| 3 | 10512 | 0.42 |
| 4 | 8928 | 0.36 |
| 5 | 8681 | 0.35 |
| 6 | 7869 | 0.32 |
| 7 | 8079 | 0.33 |
| 8 | 8232 | 0.33 |
| 9 | 8655 | 0.35 |
| 10 | 8856 | 0.36 |
| 11 | 8571 | 0.35 |
| 12 | 8843 | 0.36 |
| 13 | 8833 | 0.36 |
| 14 | 8568 | 0.35 |
| 15 | 8775 | 0.35 |
| 16 | 8970 | 0.36 |
| 17 | 10255 | 0.41 |
| 18 | 10486 | 0.42 |
| 19 | 10304 | 0.42 |
| 20 | 10376 | 0.42 |
| 21 | 10230 | 0.41 |
| 22 | 9289 | 0.37 |
| 23 | 9594 | 0.39 |

Table 4.1: Arrival rates at each UTC hour

mum log-likelihood methods. We model players' waiting behavior as follows: each player $i$ spends a maximum amount of time $Z_i$ waiting on the platform and leaves the platform if not matched after time $Z_i$. In the Prismata dataset, we observe a truncated version of $Z_i$, which are the logged waiting times $W_i$. $Z_i$ is $W_i$ when a player cancels his/her match request and is a right censored version of $Z_i$ when a player is matched with either bots or other players. We denote the players' (homogeneous) survival function as $S_W(t)$. $S_W(t) \coloneqq \mathbb{P}(Z_i \geq t)$ defines the survival probability that a player is still on the platform after waiting for $t$ minutes for any $t \geq 0$. $h(t)$ denotes the corresponding hazard function, i.e., $h(t) = -\frac{S'(t)}{S(t)}$. In the following, we derive the maximum log-likelihood procedure to estimate the waiting power.

| arrival rate regime | total arrivals | number of hours | arrivals per minute |
|---|---|---|---|
| Low | 130743 | 15 | 0.35 |
| High | 99474 | 9 | 0.45 |

Table 4.2: High/Low arrival rates regimes

**Maximum log-likelihood Procedure.** We consider the following regression model for the hazard rate function:

$$h(t) = \beta_W t^{k_W - 1} \tag{4.3}$$

where $\beta_W, k_W > 0$. Then, by integrating the hazard rate function, we have

$$S(t) = \exp^{-\int_0^t h(\tau) d\tau} = \exp^{-\beta_W \frac{t^{k_W}}{k_W}} . \tag{4.4}$$

Recall $W_i$ denotes the waiting time for the $i$-th player and we further define the dependent variable as $Y_i \coloneqq \mathbb{I}\{\text{player } i \text{ is matched}\}$. For a given $\beta_W, k_W$, the log-likelihood $\ell(\beta_W, k_W)$ follows:

$$\ell(\beta, k) = \log\left(\prod_{i=1}^{n} \mathbb{P}(Y_i = y_i, W_i = w_i)\right) = \sum_{i=1}^{n}\left(\log(\mathbb{P}(Y_i = y_i | W_i = w_i)) + \log(\mathbb{P}(W_i = w_i))\right) \tag{4.5}$$

The distribution of $W_i$ is considered to be determined under the current matching policy at Prismata, which is not a function of $\beta_W$ and $k_W$. Under this setup, the MLE estimator for $\beta_W$ and $k_W$ is equivalent to the solution to the following minimization problem:

$$\min_{\beta_W, k_W}\left[\sum_{i:y_i=1} \beta_W \frac{w_i^{k_W}}{k_W} - \sum_{i:y_i=0} \log\left(1 - \exp\left(-\beta_W \frac{w_i^{k_W}}{k_W}\right)\right)\right]. \tag{4.6}$$

For numerical stability, we perform a change of variable

$$\gamma_W \coloneqq \frac{\beta_W}{k_W}. \tag{4.7}$$

| | waiting time (in mins) |
|---|---|
| count | 106026 |
| mean | 1.84 |
| std | 2.79 |
| min | 0.00 |
| 25% | 0.08 |
| 50% | 0.62 |
| 75% | 2.14 |
| max | 15.00 |

(a) matched players

| | waiting time (in mins) |
|---|---|
| count | 23192 |
| mean | 3.78 |
| std | 4.95 |
| min | 0.02 |
| 25% | 0.28 |
| 50% | 1.52 |
| 75% | 5.47 |
| max | 22.00 |

(b) departed players

Table 4.3: Waiting time statistics

Then, we rewrite the minimization problem as follows:

$$\min_{\gamma_W, k_W} \left[ \sum_{i:y_i=1} \gamma_W w_i^{k_W} - \sum_{i:y_i=0} \log\left(1 - \exp(-\gamma_W w_i^{k_W})\right) \right]. \tag{4.8}$$

**Estimation results.** When matching with bots, the waiting time is within seconds, which is not comparable when matching with other players or departing the platform. Moreover, we don't observe whether the player joins the game after matching with bots, which causes the waiting time to be less meaningful when representing users' engagement. Hence in the following section and throughout, we only consider the arrivals that are either matched with another player or leaving the platform [3]. We remove the extreme observations which have waiting times beyond 95% quantile in the logged data, and present the statistics of the remained observations in Table 4.3: the waiting time for matched players has a both smaller mean and standard deviation compared to the players who depart the platform. Next, we use the aforementioned MLE procedure to estimate the waiting power. We report the estimation results in Table 4.4. The waiting power is estimated as 0.207 with high statistical significance.

[3] Some players hit *start_automatch* again after waiting a couple of seconds and eventually leave. In this case, we count this as one departure and add two waiting times as the player's waiting time.

| | | | |
|---|---|---|---|
| **Dep. Variable:** | matched | **Log-Likelihood:** | -59141. |
| **Method:** | Maximum Likelihood | **AIC:** | 1.183e+05 |
| **No. Observations:** | 129218 | **BIC:** | 1.183e+05 |
| **Df Residuals:** | 129217 | | |

| | **coef** | **std err** | **z** | **P> \|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| $k_W$ | 0.207 | 0.004 | 55.314 | 0.000 | 0.200 | 0.215 |
| $\gamma_W$ | 0.209 | 0.001 | 151.753 | 0.000 | 0.206 | 0.212 |

Table 4.4: Fitting results for waiting power

**Estimation of matching power**

We use a similar method to estimate $k_M$ in the matching cost. The main difference is that there are no observations where the players *cancel_automatch* due to the large matching distance. This is because of the limitation of our dataset: we only observe matching distance when two players are matched by the platform. In other words, there are no logs of players leaving the system due to a large matching distance.

Due to this data limitation, we use a surrogate dependent variable by looking into the next round interaction (i.e., the next log of *start_automatch*) with the platform. If a player does not enter the platform again within a prefixed time window, we define the player as churned [4]. We assume the following players' behavior: each player $i$ is willing to accept a maximum amount of absolute matching distance $M_i$ and churns if the matching distance is above the threshold. In the Prismata dataset, we observe a censored version of $M_i$, which are the logged matching distance $D_i := |X_i - X_{m(i)}|$ where $m(i)$ denotes the opponent for player $i$ under the current matching policy at Prismata. We denote the players' (homogeneous) survival function as $S_M(x)$. $S_M(d) := \mathbb{P}(M_i \geq d)$ defines the survival probability that a player is not churned after being matched for an Elo distance of $d$ for any $d \geq 0$. $h_M(d)$ denotes the corresponding hazard function, i.e., $h_M(d) = -\frac{S'_M(d)}{S_M(d)}$. In the following, we derive the maximum log-likelihood procedure to estimate the waiting power.

[4]The current threshold is two hours. We also vary the cutoff time window to verify our analysis is not sensitive to this threshold.

| | absolute rating difference |
|---|---|
| count | 85841.00 |
| mean | 143.66 |
| std | 98.13 |
| min | 0.00 |
| 25% | 62.20 |
| 50% | 128.78 |
| 75% | 211.36 |
| max | 392.97 |

(a) no-churn players

| | absolute rating difference |
|---|---|
| count | 14897.00 |
| mean | 150.36 |
| std | 102.04 |
| min | 0.04 |
| 25% | 64.60 |
| 50% | 134.82 |
| 75% | 223.97 |
| max | 392.38 |

(b) churned players

Table 4.5: Current matching distance statistics

**Maximum log-likelihood Procedure.** We assume that the hazard rate function

$$h_M(d) = \beta_M d^{k_M - 1}. \tag{4.9}$$

$y_i = 0$ if not churned and 1 if churned. $d_i$ is the observed matching distance for player $i$. Then, the MLE estimator for $\gamma_M, k_M$ is the solution of the following minimization problem:

$$\min_{\gamma_M, k_M} \left[ \sum_{i:y_i=0} \gamma_M d_i^{k_M} - \sum_{i:y_i=1} \log\left(1 - \exp(-\gamma_M d_i^{k_M})\right) \right].$$

**Estimation results.** In the Prismata dataset, for players who are matched at the current game, the churn rate is 15.06%; for players who are not matched at the current game, the churn rate is 48.82%. The two populations (i.e., players who are matched at the current game and those who are not) are different regarding the churn risk. Moreover, players who are not matched don't observe the matching distance. As a result, we will not be able to analyze the effect of matching distance on those players. In this section, we will focus on the churn risk for the players who have matched in the current game.

We remove the extreme observations in the logged data, i.e. if the matching distance is greater than the 95% quantile of the matching distance for all matched players. We present the statistics of the current matching distance in Table 4.5. For players who do not churn after playing the current

game, their current matching distance has a both smaller mean and standard deviation compared to those who churn.

We use the same MLE procedure after normalizing the absolute matching distance by the sample standard deviation. Note that the dependent variable is now whether the player churns after the current game. We report the estimation results in Table 4.6.

| | | | |
|---|---|---|---|
| **Dep. Variable:** | churn | **Log-Likelihood:** | -42199. |
| **Model:** | MyOLS | **AIC:** | 8.440e+04 |
| **Method:** | Maximum Likelihood | **BIC:** | 8.440e+04 |
| **No. Observations:** | 100738 | | |
| **Df Residuals:** | 100737 | | |

| | **coef** | **std err** | **z** | **P$>$ \|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| $k_M$ | 0.0372 | 0.008 | 4.743 | 0.000 | 0.022 | 0.053 |
| $\gamma_M$ | 0.1599 | 0.001 | 121.802 | 0.000 | 0.157 | 0.162 |

Table 4.6: Fitting results for matching power

The fitting result shows that if our objective is to minimize the risk of churning once matched, the corresponding power for matching distance is $0.0372$ with 95% statistical significance. One potential problem with the fitting result above is that the matching distance and waiting time are correlated, and players may churn based on the long waiting time of the previous game. However, we observe empirically that the correlation between waiting time and churn is weak and waiting time affects mostly whether the player is matched at the current game. We fit the waiting power using churn as the dependent variable for both the matched population and the departed population and present the results in Table 4.7 and Table 4.8. There is no statistical significance on whether waiting time affects the churn behavior for the departed population and although there is some statistical evidence that the waiting time can further affect the churn behavior for the matched population, the waiting power is negligible compared to the magnitude when fitting against the waiting time.

| | | | |
|---|---|---|---|
| **Dep. Variable:** | churn | **Log-Likelihood:** | -16068. |
| **Model:** | MyOLS | **AIC:** | 3.214e+04 |
| **Method:** | Maximum Likelihood | **BIC:** | 3.214e+04 |
| **No. Observations:** | 23192 | | |
| **Df Residuals:** | 23191 | | |

| | **coef** | **std err** | **z** | **P> \|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| $k_W$ | -0.0050 | 0.005 | -1.057 | 0.291 | -0.014 | 0.004 |
| $\gamma_W$ | 0.7174 | 0.007 | 106.572 | 0.000 | 0.704 | 0.731 |

Table 4.7: MLE fitting results on the effect of waiting time for departed population

| | | | |
|---|---|---|---|
| **Dep. Variable:** | churn | **Log-Likelihood:** | -44928. |
| **Model:** | MyOLS | **AIC:** | 8.986e+04 |
| **Method:** | Maximum Likelihood | **BIC:** | 8.986e+04 |
| **No. Observations:** | 106026 | | |
| **Df Residuals:** | 106026 | | |
| **Df Model:** | 0 | | |

| | **coef** | **std err** | **z** | **P> \|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| $k_W$ | 0.0057 | 0.002 | 2.926 | 0.003 | 0.002 | 0.009 |
| $\gamma_W$ | 1.9008 | 0.008 | 244.042 | 0.000 | 1.886 | 1.916 |

Table 4.8: MLE fitting results on the effect of waiting time for matched population

**Relative weight between waiting and matching cost**

The relative weight between waiting and matching costs depends on the platform's objective of balancing current matched players and the future no-churn players once matched. To maximize the matched players, the objective is to match the survival probability:

$$\max_{t,d\geq 0} \exp\left(-\gamma_W t^{k_W}\right), \tag{4.10}$$

which is equivalent to

$$\min_{t,d\geq 0} t^{k_W}. \tag{4.11}$$

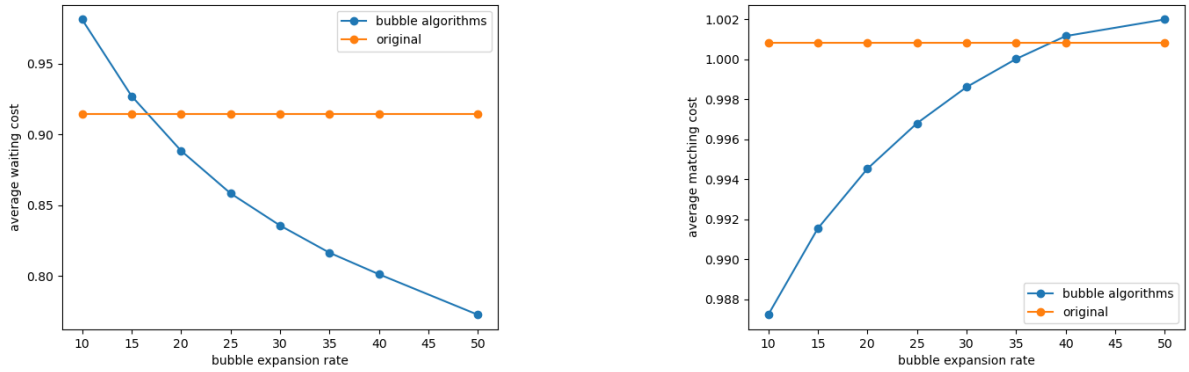


Figure 4.1: Performance of different bubble algorithms

Similarly, to maximize the no-churn probability for the matched players, the optimization problem can be expressed as the following:

$$\max_{t,d\geq 0} \exp\left(-\gamma_W t^{k_W}\right) \cdot \exp\left(-\gamma_M d^{k_M}\right), \tag{4.12}$$

which is equivalent to minimize

$$\max_{t,d\geq 0} t^{k_W} + \frac{\gamma_M}{\gamma_W} d^{k_M}. \tag{4.13}$$

The relative weight between waiting and matching costs depends on how the platform balances the matching of the current players versus the future churn behaviors. Nevertheless, we show that the platform can achieve lower waiting costs and matching costs simultaneously via a clever choice of the bubble expansion rate.

### 4.4.2 Select bubble expansion rate.

We consider the following bubble expansion rates, $\alpha \in [10, 15, 20, 25, 30, 35, 40, 50]$, which is measured as Elo per minute. We simulate the matching from the existing dataset as follows: for a given player, we find the next arrival such that the two player's bubbles will touch and then remove these two players from the dataset. The corresponding waiting times and matching distance are recorded to further calculate the waiting/matching cost of the new matching. We observe the increase in waiting cost and the decrease in matching cost as the bubble expansion rate increases.

| $\alpha$ | waiting cost reduction | matching cost reduction |
|---|---|---|
| 10 | -7.50% | 1.22% |
| 15 | -1.46% | 0.83% |
| 20 | 2.73% | 0.56% |
| 25 | 6.03% | 0.35% |
| 30 | 8.53% | 0.18% |
| 35 | 10.68% | 0.06% |
| 40 | 12.38% | -0.04% |
| 50 | 15.49% | -0.12% |

Table 4.9: Relative cost reduction compared to the benchmark

We plot the original matching average waiting/matching cost as the benchmark. From Figure 4.1, we observe that $\alpha \in \{20, 25, 30, 35\}$ Elo/min achieves a good balance between matching cost and waiting cost, which reduces both costs compared to the benchmark costs in the original dataset.

In Table 4.9, we present the relative cost reduction (the larger, the better) under different Bubble expansion rates. When $\alpha = 20$, the Bubble algorithm has a relative reduction of $2.73\%$ in the waiting cost and $0.56\%$ in the matching cost; when $\alpha = 25$, the Bubble algorithm has a relative reduction of $6.03\%$ in the waiting cost and $0.35\%$ in the matching cost; when $\alpha = 30$, the Bubble algorithm has a relative reduction of $8.53\%$ in the waiting cost and $0.18\%$ in the matching cost. The platform may choose any of the three above rates based on their relative weight on reducing waiting costs versus matching costs.

If the platform has the flexibility to vary the bubble expansion rate based on the arrival rate, it can further improve the waiting and matching cost. Based on the theoretical results, the expansion rate should increase when the arrival rate is lower. We select the different high/low bubble expansion rate pairs and plot their relative improvements over the original matching policy in Figure 4.2. The orange line plots the relative percentage reduction in waiting costs and matching costs for a single bubble expansion rate; the blue circle plots the reduction in both costs for a pair of bubble expansion rates (high expansion rate when the arrival rate is low and low expansion rate when the arrival rate is high); the green triangle plots the reduction in both costs for a pair of bubble expansion rates (low expansion rate when the arrival rate is low and high expansion rate when the arrival

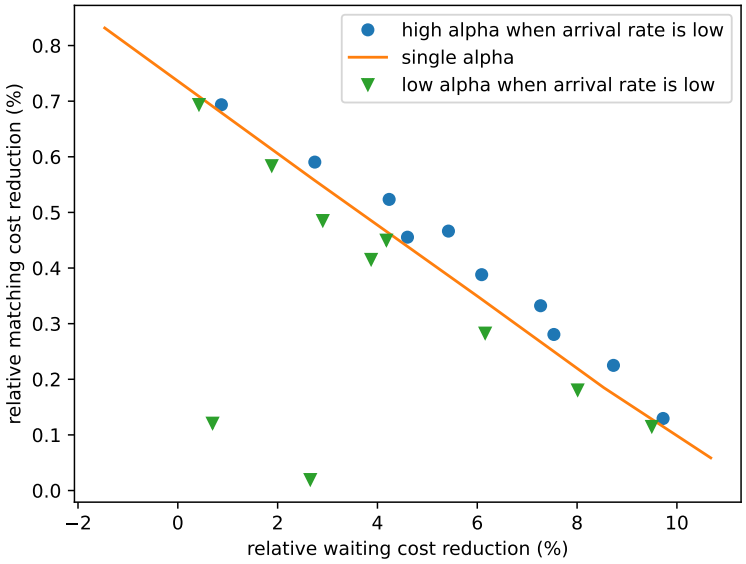


Figure 4.2: Relative cost reduction with different expansion rates

rate is high). We observe that the blue circles achieve a higher matching cost reduction at the same waiting cost reduction level compared to the single expansion rate policy while the green triangles achieve a lower matching cost reduction at the same waiting cost reduction level compared to the single expansion rate policy, which empirically verifies our theoretical insights on increasing the expansion rate when the arrival rate is lower.

## 4.5 Conclusion

In this work, we introduce and study a general dynamic match-making model to improve players' overall engagement. For companies with a relatively thinner market, there is a key trade-off between waiting time and matching quality. We show that under a weighted combination of power-law waiting/matching costs, the Bubble algorithm achieves a constant factor competitive ratio compared to the offline optimal benchmark. In the empirical part of this work, we show how companies can estimate the powers from players' activity logs data and further tune the Bubble expansion rate based on players' arrival rates.

# References


Agarwal, Naman, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh (2019). "Online control with adversarial disturbances". In: *International Conference on Machine Learning*. PMLR, pp. 111–119.

Akbari, Mohammad, Bahman Gharesifard, and Tamas Linder (2017). "Distributed online convex optimization on time-varying directed graphs". In: *IEEE Transactions on Control of Network Systems* 4.3.

Akbarpour, Mohammad, Suraj Malladi, and Amin Saberi (2018). "Diffusion, seeding, and the value of network information". In: *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 641–641.

Anderson, Ross, Itai Ashlagi, David Gamarnik, and Yash Kanoria (2017). "Efficient dynamic barter exchange". In: *Operations Research* 65.6, pp. 1446–1459.

Antoniadis, Antonios, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon (2020). "Online metric algorithms with untrusted predictions". In: *International Conference on Machine Learning*. PMLR, pp. 345–355.

Aouad, Ali and Ömer Saritaç (2020). "Dynamic stochastic matching under limited time". In: *Proceedings of the 21st ACM Conference on Economics and Computation*, pp. 789–790.

Argue, CJ, Anupam Gupta, and Guru Guruganesh (2020). "Dimension-Free Bounds for Chasing Convex Functions". In: *Conference on Learning Theory*. PMLR, pp. 219–241.

Argue, CJ, Anupam Gupta, Guru Guruganesh, and Ziye Tang (2020). "Chasing convex bodies with linear competitive ratio". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, pp. 1519–1524.

Ashlagi, Itai, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley (2019). "Edge weighted online windowed matching". In: *ACM EC 2019 - Proceedings of the 2019 ACM Conference on Economics and Computation*. ISBN: 9781450367929.

Badiei, Masoud, Na Li, and Adam Wierman (2015). "Online convex optimization with ramp constraints". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 6730–6736.

Bandyopadhyay, Antar and David Gamarnik (2008). "Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models". In: *Random Structures & Algorithms* 33.4, pp. 452–479.

Banerjee, Suman, Mamata Jenamani, and Dilip Kumar Pratihar (2020). "A survey on influence maximization in a social network". In: *Knowledge and Information Systems*. arXiv: 1808.05502.

Bent, Russell and Pascal Van Hentenryck (Jan. 2004). "Online Stochastic and Robust Optimization". In: pp. 286–300. ISBN: 978-3-540-24087-7.

Bertsekas, Dimitri (2012). *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific.

Bertsekas, Dimitri and John Tsitsiklis (1989a). *Parallel and Distributed Computation: Numeral Methods*. Prentice-Hall Inc.

Bertsekas, Dimitri P (1995). *Dynamic programming and optimal control*. Vol. 1. Athena scientific Belmont, MA.

Bertsekas, Dimitri P. (2011). *Dynamic Programming and Optimal Vol 2*. Vol. II.

Bertsekas, Dimitri P. and John N. Tsitsiklis (1989b). *Parallel and Distributed Computation Numerical Methods Optimization and Neural Computation*.

Besbes, Omar, Yonatan Gur, and Assaf Zeevi (2016). "Optimization in online content recommendation services: Beyond click-through rates". In: *Manufacturing and Service Operations Management*.

Besbes, Omar and Ilan Lobel (2015). "Intertemporal price discrimination: Structure and computation of optimal policies". In: *Management Science* 61.1, pp. 92–110.

Biggs, Max, Wei Sun, and Markus Ettl (2021). "Model distillation for revenue optimization: Interpretable personalized pricing". In: *International Conference on Machine Learning*. PMLR, pp. 946–956.

Boyd, Stephen, Neal Parikh, and Eric Chu (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Candogan, Ozan, Kostas Bimpikis, and Asuman Ozdaglar (2012). "Optimal pricing in networks with externalities". In: *Operations Research* 60.4, pp. 883–905.

Cao, Xuanyu and Tamer Basar (2021). "Decentralized Online Convex Optimization with Feedback Delays". In: *IEEE Transactions on Automatic Control*.

Cao, Xuanyu and Tamer Başar (2021). "Decentralized online convex optimization based on signs of relative states". In: *Automatica* 129.

Cao, Xuanyu, Junshan Zhang, and H. Vincent Poor (2021). "Online Stochastic Optimization With Time-Varying Distributions". In: *IEEE Transactions on Automatic Control* 66.4, pp. 1840–1847.

Caro, Felipe and Jérémie Gallien (2012). "Clearance pricing optimization for a fast-fashion retailer". In: *Operations Research* 60.6, pp. 1404–1422.

Chen, Ming and Zhi Long Chen (2015). "Recent developments in dynamic pricing research: Multiple products, competition, and limited demand information". In: *Production and Operations Management* 24 (5).

Chen, Mingliu, Adam Elmachtoub, and Xiao Lei (2021). "Matchmaking Strategies for Maximizing Player Engagement in Video Games". In: *SSRN Electronic Journal*.

Chen, Niangjun, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman (2016). "Using predictions in online optimization: Looking forward with an eye on the past". In: *ACM SIGMETRICS Performance Evaluation Review* 44.1, pp. 193–206.

Chen, Niangjun, Gautam Goel, and Adam Wierman (2018). "Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent". In: *Proceedings of Conference On Learning Theory (COLT)*, pp. 1574–1594.

Chen, Yang Quan and Zhongmin Wang (2005). "Formation control: A review and a new consideration". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*.

Chen, Yiwei and Vivek F Farias (2018). "Robust dynamic pricing with strategic customers". In: *Mathematics of Operations Research* 43.4, pp. 1119–1142.

Chen, Zhengxing, Navid Aghdaie, Su Xue, Kazi A. Zaman, Magy Seif El-Nasr, John Kolen, and Yizhou Sun (2017). "EOMM: An engagement optimized matchmaking framework". In: *26th International World Wide Web Conference, WWW 2017*. ISBN: 9781450349130. arXiv: 1702.06820.

Chen, Zongchen, Kuikui Liu, and Eric Vigoda (2020). "Rapid Mixing of Glauber Dynamics up to Uniqueness via Contraction". In: arXiv: 2004.09083.

Collina, Natalie, Nicole Immorlica, Kevin Leyton-Brown, Brendan Lucier, and Neil Newman (2020). "Dynamic Weighted Matching with Heterogeneous Arrival and Departure Rates". In: *International Conference on Web and Internet Economics*. Springer, pp. 17–30.

Devari, Aashwinikumar, Alexander G. Nikolaev, and Qing He (2017). "Crowdsourcing the last mile delivery of online orders by exploiting the social networks of retail store customers". In: *Transportation Research Part E: Logistics and Transportation Review*.

Ding, Jian, Allan Sly, and Nike Sun (2015). "Proof of the satisfiability conjecture for large k". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 59–68.

Dobrushin, Roland L (1970). "Prescribing a system of random variables by conditional distributions". In: *Theory of Probability & Its Applications* 15.3, pp. 458–486.

Fajgelbaum, Pablo, Amit Khandelwal, Wookun Kim, Cristiano Mantovani, and Edouard Schaal (2020). *Optimal lockdown in a commuting network*. Tech. rep. National Bureau of Economic Research.

Fatehi, Soraya and Michael R. Wagner (2021). "Crowdsourcing Last-Mile Deliveries". In: *Manufacturing & Service Operations Management*.

Fiacco, Anthony V and Yo Ishizuka (1990). "Sensitivity and stability analysis for nonlinear programming". In: *Annals of Operations Research* 27.1, pp. 215–235.

Forsgren, Anders, Philip E Gill, and Margaret H Wright (2002). "Interior methods for nonlinear optimization". In: *SIAM review* 44.4, pp. 525–597.

Friedman, Joel (2008). *A proof of Alon's second eigenvalue conjecture and related problems*. American Mathematical Soc.

Gallego, Guillermo and Huseyin Topaloglu (2019). "Revenue management and pricing analytics". In: *International Series in Operations Research and Management Science* 279.

Gamarnik, David (2014). "Correlation Decay Method for Decision, Optimization, and Inference in Large-Scale Networks". In: *Theory Driven by Influential Applications*. Chap. Chapter 6, pp. 108–121. eprint: `https://pubsonline.informs.org/doi/pdf/10.1287/educ.2013.0119`.

Gamarnik, David and David A. Goldberg (2010). "Randomized greedy algorithms for independent sets and matchings in regular graphs: Exact results and finite girth corrections". In: *Combinatorics Probability and Computing* 19.1, pp. 61–85.

Gamarnik, David, David A. Goldberg, and Theophane Weber (2014). "Correlation decay in random decision networks". In: *Mathematics of Operations Research* 39.2, pp. 229–261.

Gamarnik, David and Dmitriy Katz (2009). "Sequential cavity method for computing free energy and surface pressure". In: *Journal of Statistical Physics* 137.2, p. 205.

Garcia, Carlos E, David M Prett, and Manfred Morari (1989). "Model predictive control: Theory and practice—A survey". In: *Automatica* 25.3, pp. 335–348.

Goel, Gautam, Yiheng Lin, Haoyuan Sun, and Adam Wierman (2019). "Beyond Online Balanced Descent: An Optimal Algorithm for Smoothed Online Optimization". In: *Neural Information Processing Systems (NeurIPS)*.

Goel, Gautam and Adam Wierman (2019). "An Online Algorithm for Smoothed Regression and LQR Control". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Gurobi Optimization, LLC (2022). *Gurobi Optimizer Reference Manual*.

Hazan, Elad (2021). *Introduction to Online Convex Optimization*. arXiv: `1909.05207 [cs.LG]`.

Hochba, Dorit S (1997). "Approximation algorithms for NP-hard problems". In: *ACM Sigact News* 28.2, pp. 40–52.

Hosseini, Saghar, Airlie Chapman, and Mehran Mesbahi (2016). "Online Distributed Convex Optimization on Dynamic Networks". In: *IEEE Transactions on Automatic Control* 61.11.

Huang, Yan, Stefanus Jasin, and Puneet Manchanda (2019). ""Level Up": Leveraging skill and engagement to maximize player game-play in online video games". In: *Information Systems Research*.

Joseph, Vinay and Gustavo de Veciana (2012). "Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs". In: *Proceedings of the IEEE INFOCOM*, pp. 567–575.

Kempe, David, Jon Kleinberg, and Éva Tardos (2015). "Maximizing the spread of influence through a social network". In: *Theory of Computing* 11.

Kerimov, Azer (2014). "A disagreement-percolation type uniqueness condition for Gibbs states in models with long-range interactions". In: *Journal of Statistical Mechanics: Theory and Experiment*.

Kim, S. and G. B. Giannakis (2017). "An Online Convex Optimization Approach to Real-Time Energy Pricing for Demand Response". In: *IEEE Transactions on Smart Grid* 8.6, pp. 2784–2793.

Leduc, Matt V, Matthew O Jackson, and Ramesh Johari (2017). "Pricing and referrals in diffusion on networks". In: *Games and Economic Behavior* 104, pp. 568–594.

Li, Xiuxian, Xinlei Yi, and Lihua Xie (2021a). "Distributed Online Convex Optimization with an Aggregative Variable". In: *IEEE Transactions on Control of Network Systems*.

Li, Xiuxian, Xinlei Yi, and Lihua Xie (2021b). "Distributed Online Optimization for Multi-Agent Networks with Coupled Inequality Constraints". In: *IEEE Transactions on Automatic Control* 66.8.

Li, Yingying, Guannan Qu, and Na Li (2020). "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit". In: *IEEE Transactions on Automatic Control*.

Lin, Minghong, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew (2012). "Online algorithms for geographical load balancing". In: *Proceedings of the International Green Computing Conference (IGCC)*, pp. 1–10.

Lin, Yiheng, Judy Gan, Guannan Qu, Yash Kanoria, and Adam Wierman (2022). *Decentralized Online Convex Optimization in Networked Systems*.

Lin, Yiheng, Gautam Goel, and Adam Wierman (2020). "Online optimization with predictions and non-convex losses". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4.1, pp. 1–32.

Lin, Yiheng, Yang Hu, Haoyuan Sun, Guanya Shi, Guannan Qu, and Adam Wierman (2021). "Perturbation-based Regret Analysis of Predictive Control in Linear Time Varying Systems". In: *arXiv preprint arXiv:2106.10497*.

Lin, Yiheng, Guannan Qu, Longbo Huang, and Adam Wierman (2020). "Distributed Reinforcement Learning in Multi-Agent Networked Systems". In: *arXiv preprint arXiv:2006.06555*.

Manshadi, Vahideh, Sidhant Misra, and Scott Rodilitz (2020). "Diffusion in random networks: Impact of degree distribution". In: *Operations Research* 68.6, pp. 1722–1741.

Marschak, J. (1955). "Elements for a Theory of Teams". In: *Management Science* 1 (2).

Marschak, Jacob and Roy Radner (1972). *Economic Theory of Teams*. Cowles Foundation for Research in Economics at Yale University.

McAuley, Julian, Rahul Pandey, and Jure Leskovec (2015). "Inferring networks of substitutable and complementary products". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ISBN: 9781450336642. arXiv: 1506.08839.

Molzahn, Daniel K, Florian Dörfler, Henrik Sandberg, Steven H Low, Sambuddha Chakrabarti, Ross Baldick, and Javad Lavaei (2017). "A survey of distributed optimization and control algorithms for electric power systems". In: *IEEE Transactions on Smart Grid* 8.6, pp. 2941–2962.

Montanari, Andrea (2019). "Optimization of the Sherrington-Kirkpatrick hamiltonian". In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, pp. 1417–1433.

Morari, Manfred and Jay H Lee (1999). "Model predictive control: past, present and future". In: *Computers & Chemical Engineering* 23.4-5, pp. 667–682.

Nedić, Angelia, Alex Olshevsky, and Michael G Rabbat (2018). "Network topology and communication-computation tradeoffs in decentralized optimization". In: *Proceedings of the IEEE* 106.5, pp. 953–976.

Oh, Kwang Kyo, Myoung Chul Park, and Hyo Sung Ahn (2015). "A survey of multi-agent formation control". In: *Automatica* 53.

Qu, Guannan, Yiheng Lin, Adam Wierman, and Na Li (2020). "Scalable Multi-Agent Reinforcement Learning for Networked Systems with Average Reward". In: arXiv: 2006.06626.

Roughgarden, Tim (2020). "Resource Augmentation". In: *CoRR* abs/2007.13234. arXiv: 2007.13234.

Schlosser, Rainer (2016). "Stochastic dynamic multi-product pricing with dynamic advertising and adoption effects". In: *Journal of Revenue and Pricing Management* 15.2, pp. 153–169.

Sellke, Mark (2020). "Chasing convex bodies optimally". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, pp. 1509–1518.

Shi, Guanya, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman (2020). "Online Optimization with Memory and Competitive Control". In: *Advances in Neural Information Processing Systems* 33, pp. 20636–20647.

Shi, Ming, Xiaojun Lin, and Lei Jiao (2019). "On the Value of Look-Ahead in Competitive Online Convex Optimization". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3.2, p. 22.

Shi, Wei, Qing Ling, Gang Wu, and Wotao Yin (2015). "Extra: An exact first-order algorithm for decentralized consensus optimization". In: *SIAM Journal on Optimization* 25.2, pp. 944–966.

Shi, Yuanyuan, Guannan Qu, Steven Low, Anima Anandkumar, and Adam Wierman (2021). "Stability Constrained Reinforcement Learning for Real-Time Voltage Control". In: *arXiv preprint arXiv:2109.14854*.

Shin, Sungho, Mihai Anitescu, and Victor M. Zavala (2021). "Exponential Decay of Sensitivity in Graph-Structured Nonlinear Programs". In: *arXiv preprint arXiv:2101.03067*.

Shin, Sungho and Victor M Zavala (2021). "Controllability and observability imply exponential decay of sensitivity in dynamic optimization". In: *IFAC-PapersOnLine* 54.6, pp. 179–184.

Shin, Sungho, Victor M. Zavala, and Mihai Anitescu (2020). "Decentralized Schemes With Overlap for Solving Graph-Structured Optimization Problems". In: *IEEE Transactions on Control of Network Systems* 7.3, 1225–1236.

Sivan, Balasubramanian (2013). "Prior robust optimization". PhD thesis. The University of Wisconsin-Madison.

Song, Inseong and Pradeep K Chintagunta (2006). "Measuring cross-category price effects with aggregate store data". In: *Management Science* 52.10, pp. 1594–1609.

Stanica, Pantelimon (2001). "Good lower and upper bounds on binomial coefficients". In: *Journal of Inequalities in Pure and Applied Mathematics* 2.3, p. 30.

Suomela, Jukka (2013). "Survey of local algorithms". In: *ACM Computing Surveys (CSUR)* 45.2, pp. 1–40.

Talluri, Kalyan and Garrett Van Ryzin (2006). "The Theory and Practice of Revenue Management, Springer". In: *International Series in Operations Research and Management* 68.

Tatikonda, Sekhar C. and Michael I. Jordan (2002). "Loopy Belief Propagation and Gibbs Measures". In: *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*. UAI'02. Alberta, Canada: Morgan Kaufmann Publishers Inc., 493–500. ISBN: 1558608974.

Tong, Guangmo, Weili Wu, Shaojie Tang, and Ding Zhu Du (2017). "Adaptive Influence Maximization in Dynamic Social Networks". In: *IEEE/ACM Transactions on Networking*. arXiv: 1506.06294.

Tretter, Christiane (2008). "Spectral theory of block operator matrices and applications". In: *Spectral Theory of Block Operator Matrices and Applications*.

Tsitsiklis, John and Benjamin Van Roy (1996). "Analysis of temporal-diffference learning with function approximation". In: *Advances in neural information processing systems* 9.

Tsitsiklis, John N. and Benjamin Van Roy (2002). "On average versus discounted reward temporal-difference learning". In: *Machine Learning* 49 (2-3).

Ullah, Farman and Sungchang Lee (2016). "Social content recommendation based on spatial-temporal aware diffusion modeling in social networks". In: *Symmetry*.

Vadhan, Salil P et al. (2012). *Pseudorandomness*. Vol. 7. Now Delft.

Wainwright, Martin J and Michael Irwin Jordan (2008). *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

Weber, Theophane (Sept. 2010). "Correlation Decay and Decentralized Optimization in Graphical Models". In.

Weitz, Dror (2006). "Counting independent sets up to the tree threshold". In: *Proceedings of the Annual ACM Symposium on Theory of Computing*. ISBN: 1595931341.

Xin, Ran, Shi Pu, Angelia Nedić, and Usman A Khan (2020). "A general framework for decentralized optimization with first-order methods". In: *Proceedings of the IEEE* 108.11, pp. 1869–1889.

Yi, Xinlei, Xiuxian Li, Lihua Xie, and Karl H. Johansson (2020). "Distributed Online Convex Optimization with Time-Varying Coupled Inequality Constraints". In: *IEEE Transactions on Signal Processing* 68.

Yin, Hongzhi, Bin Cui, Ling Chen, Zhiting Hu, and Xiaofang Zhou (2015). "Dynamic user modeling in social media systems". In: *ACM Transactions on Information Systems*.

Yu, Chenkai, Guanya Shi, Soon-Jo Chung, Yisong Yue, and Adam Wierman (2020). "The Power of Predictions in Online Control". In: *Advances in Neural Information Processing Systems* 33.

Yuan, Deming, Alexandre Proutiere, and Guodong Shi (2021). "Distributed Online Optimization with Long-Term Constraints". In: *IEEE Transactions on Automatic Control*.

# Appendix A: Details in Chapter 3

## A.1 Notation Summary and Definitions in Chapter 3

The notation we use throughout the chapter is summarized in the following two tables.

Table A.1: Notation related to the graph/network structures.

| Notation | Meaning |
|---|---|
| $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ | The network of agents connected by undirected edges; |
| $\text{dist}(u, v)$ | The graph distance (i.e. the length of the shortest path) between two vertices $u$ and $v$ in $\mathcal{G}$; |
| $N_v^r$ | The $r$-hop neighborhood of vertex/agent $v$ in $\mathcal{G}$, i.e., $N_v^r := \{u \in \mathcal{V} \mid \text{dist}(u, v) \leq r\}$; |
| $\partial N_v^r$ | The boundary of the $r$-hop neighborhood of vertex/agent $v$, i.e., $\partial N_v^r = N_v^r \setminus N_v^{r-1}$; |
| $h(r)$ | The size of the largest $r$-hop boundary in $\mathcal{G}$, i.e., $h(r) := \sup_v \lvert\partial N_v^r\rvert$; |
| $\Delta$ | The maximum degree of any vertex $v$ in $\mathcal{G}$; |
| $\mathcal{E}(S)$ | The set of all edges that have both endpoints in $S$, where $S \subseteq \mathcal{V}$; |
| $S_+$ | The extension of $S$ by 1-hop, i.e., $S_+ = S \cup \{u \mid \exists v \in S \text{ s.t. } (u, v) \in \mathcal{E}\}$; |
| $N_{(t,v)}^{(k,r)}$ | $\{\tau \in \mathbb{Z} \mid t \leq \tau < t + k\} \times N_v^r$, which is a set of (time, vertex) pairs; |
| $\partial N_{(t,v)}^{(k,r)}$ | $N_{(t,v)}^{(k,r)} \setminus N_{(t,v)}^{(k-1,r-1)}$, which is a set of (time, vertex) pairs; |

Table A.2: Notation related to the optimization problems.

| Notation | Meaning |
|---|---|
| $\|\cdot\|$ | The (Euclidean) 2-norm for vectors and the induced 2-norm for matrices; |
| $H$ | The whole horizon length of Networked OCO problem; |
| $[H]$ | The sequence of integers $1, 2, \ldots, H$; |
| $\mathbb{S}^m$ | For any positive integer $m$, $\mathbb{S}^m$ denotes the set of all symmetric real $m \times m$ matrices; |
| $y_{t_1:t_2}$ | The sequence $y_{t_1}, y_{t_1+1}, \ldots, y_{t_2}$, for $t_2 \geq t_1$; |
| $x_t^v$ | The individual action of agent $v$ at time step $t$. It is a vector in $\mathbb{R}^n$; |
| $x_t^S$ | The joint action of all agents in set $S \subseteq \mathcal{V}$ at time $t$, i.e., $x_t^S = \{x_t^v\}_{v \in S}$; |
| $x_t$ | The joint action of all agents in $\mathcal{V}$ at time $t$, i.e., $x_t = \{x_t^v\}_{v \in \mathcal{V}}$. It is a shorthand of $x_t^{\mathcal{V}}$; |
| $x_t^*$ | The offline optimal joint action of all agents at time $t$; |
| $x_{\tau\|t}^*$ | The clairvoyant joint decision of all agents at time $\tau$ given that the joint decision is $x_t$ at time $t$; |
| $f_t^v(x_t^v)$ | The node cost function for agent $v \in \mathcal{V}$ at time step $t$; |
| $c_t^v(x_t^v, x_{t-1}^v)$ | The temporal interaction cost function for agent $v \in \mathcal{V}$ at time step $t$; |
| $s_t^e(x_t^v, x_t^u)$ | The spatial interaction cost for edge $e = (v, u) \in \mathcal{E}$ at time step $t$; |
| $\mu$ | The strong convexity constant of node costs $f_t^v$; |
| $\ell_f, \ell_T, \ell_S$ | The smoothness constant of node costs, temporal interaction costs, and spatial interaction costs; |
| $D_t^v$ | The feasible set of $x_t^v$ for agent $v$ at time $t$. It is a convex subset of $\mathbb{R}^n$; |
| $\theta_t^v$ | The minimizer of node cost function for $v$ at time $t$ subject to $D_t^v$, i.e., $\theta_t^v = \arg\min_{y \in D_t^v} f_t^v(y)$; |
| $f_t^S(x_t^{S_+})$ | The total node costs and spatial interaction costs over a subset $S \subseteq \mathcal{V}$ at time $t$, i.e., |
| | $f_t^S(x_t^S) := \sum_{v \in S} f_t^v(x_t^v) + \sum_{(v,u) \in \mathcal{E}(S)} s_t^{(v,u)}(x_t^v, x_t^u)$; |
| $c_t^S(x_t^S)$ | The total temporal interaction costs over a subset $S \subseteq \mathcal{V}$ at time $t$, i.e., $c_t^S(x_t^S) := \sum_{v \in S} c_t^v(x_t^v, x_{t-1}^v)$; |
| $f_t(x_t)$ | The total node costs and spatial interaction costs over a $\mathcal{V}$ at time $t$. A shorthand of $f_t^{\mathcal{V}}(x_t^{\mathcal{V}})$; |
| $c_t(x_t)$ | The total temporal interaction costs over $\mathcal{V}$ at time $t$. A shorthand of $c_t^{\mathcal{V}}(x_t^{\mathcal{V}})$; |
| $\psi_{(t,v)}^{(k,r)}(\cdot,\cdot)$ | The optimal individual decisions in $N_{(t,v)}^{(k,r)}$ when the decision boundaries formed by $\{t-1\} \times N_v^r$ |
| | and $\partial N_{(t,v)}^{(k,r)}$ are fixed as parameters; |
| $\widetilde{\psi}_t^p(\cdot,\cdot)$ | The optimal global trajectory $x_t, x_{t+1}, \ldots, x_{t+p-2}$ when $x_{t-1}$ and $x_{t+p-1}$ are fixed as parameters; |
| $\widetilde{\psi}_t(\cdot)$ | The optimal global trajectory $x_t, x_{t+1}, \ldots, x_T$ when $x_{t-1}$ is fixed as the parameter; |

In addition to the notation in the tables above, we make use of the concepts of strong convexity and smoothness throughout this paper.

**Definition A.1.1.** *For a fixed dimension $m \in \mathbb{Z}_+$, let $D \subset \mathbb{R}^m$ be a convex set, and suppose*

*function $\widehat{h} : D \to \mathbb{R}$ is a differentiable function. Then, $\widehat{h}$ is called $\ell$-smooth for some constant $\ell \in \mathbb{R}_{\geq 0}$ if*

$$\widehat{h}(y) \leq \widehat{h}(x) + \langle \nabla \widehat{h}(x), y - x \rangle + \frac{\ell}{2} \|y - x\|_2^2, \forall x, y \in \mathbb{R}^m,$$

*and is called $\mu$-strongly convex for some constant $\mu \in \mathbb{R}_{\geq 0}$ if*

$$\widehat{h}(y) \geq \widehat{h}(x) + \langle \nabla \widehat{h}(x), y - x \rangle + \frac{\mu}{2} \|y - x\|_2^2, \forall x, y \in \mathbb{R}^m.$$

*Here $\langle \cdot, \cdot \rangle$ denotes the dot product of vectors.*

## A.2 Perturbation Bounds

This section provides the full proofs of the perturbation bounds stated in Section 3.3.2.

### A.2.1 Proof of Theorem 3.3.1

We begin with a technical lemma. Recall that for any positive integer $m$, $\mathbb{S}^m$ denotes the set of all symmetric $m \times m$ real matrices.

**Lemma A.2.1.** *For a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, suppose $A$ is a positive definite matrix in $\mathbb{S}^{\sum_{i \in \mathcal{V}'} p_i}$ formed by $|\mathcal{V}'| \times |\mathcal{V}'|$ blocks, where the $(i, j)$-th block has dimension $p_i \times p_j$, i.e., $A_{i,j} \in \mathbb{R}^{p_i \times p_j}$. Assume that $A$ is $q$-banded for an even positive integer $q$; i.e.,*

$$A_{i,j} = 0, \forall d_{\mathcal{G}'}(i, j) > q/2.$$

*Let $a_0$ denote the smallest eigenvalue value of $A$, and $b_0$ denote the largest eigenvalue value of $A$. Assume that $b_0 \geq a_0 > 0$. Suppose $D = diag(D_1, \ldots, D_{|\mathcal{V}'|})$, where $D_i \in \mathbb{S}^{p_i}$ is positive semi-definite. Let $M = ((A + D)^{-1})_{S_R, S_C}$, where $S_R, S_C \subseteq \{1, \ldots, |\mathcal{V}'|\}$. Then we have $\|M\| \leq C\gamma^{\widehat{d}}$, where*

$$C = \frac{2}{a_0}, \gamma = \left( \frac{\sqrt{cond(A)} - 1}{\sqrt{cond(A)} + 1} \right)^{2/q}, \widehat{d} = \min_{i \in S_R, j \in S_c} d_{\mathcal{G}'}(i, j).$$

*Here $cond(A) = b_0/a_0$ denotes the condition number of matrix $A$.*

We can show Lemma A.2.1 using the same method as Lemma B.1 in Lin et al., 2021. We only need to note that even when the size of blocks are not identical, the $m$ th power of a $q$-banded matrix is a $qm$-banded matrix for any positive integer $m$.

With the help of Lemma A.2.1, we can proceed to show a local perturbation bound on a general $\mathcal{G}'$ in Theorem A.2.1, where $\mathcal{G}'$ can be different from the network $\mathcal{G}$ of agents in Section 3.2. Compared with Theorem 3.1 in Lin et al., 2021, Theorem A.2.1 is more general because it considers a general network of decision variables while Theorem 3.1 in Lin et al., 2021 only consider the special case of a line graph. Although Theorem A.2.1 does not consider the temporal dimension which features in the local perturbation bound defined in Definition 3.3.1, we will use it to show Theorem 3.3.1 later by redefining the variables from two perspectives.

**Theorem A.2.1.** *For a network $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ with undirected edges, suppose that each node $v \in \mathcal{V}'$ is associated with a decision vector*[1] *$\widehat{x}_v \in \mathbb{R}^{p_v}$ and a cost function $\widehat{f}_v : \mathbb{R}^{p_v} \to \mathbb{R}_{\geq 0}$, and each edge $e = (u, v) \in \mathcal{E}'$ is associated with an edge cost $\widehat{c}_e : \mathbb{R}^{p_v} \times \mathbb{R}^{p_u} \times \mathbb{R}^q \to \mathbb{R}_{\geq 0}$. Assume that $\widehat{f}_v$ is $\mu$-strongly convex for all $v \in \mathcal{V}'$ and $\widehat{c}_e$ is $\ell$-smooth for all $e \in \mathcal{E}'$. For some subset $S \subset \mathcal{V}'$, define*

$$E_0 := \{(u, v) \in \mathcal{E}' \mid u, v \in \mathcal{V}' \setminus S\},$$
$$E_1 := \{(u, v) \in \mathcal{E}' \mid u \in \mathcal{V}' \setminus S, v \in S\}.$$

*For the disturbance vectors*[2] *$w \in \mathbb{R}^{(|E_0|+|E_1|)\times q}$ indexed by $e \in E_0 \cup E_1$ and $y \in \mathbb{R}^{\sum_{v\in S} p_v}$ indexed by $v \in S$, let $\psi(w, y)$ denote the optimal solution of the optimization problem*

$$\psi(w, y) := \operatorname*{arg\,min}_{x\in\mathbb{R}^{|\mathcal{V}'\setminus S|\times d}} \sum_{v\in\mathcal{V}'\setminus S} \widehat{f}_v(\widehat{x}_v) + \sum_{(u,v)\in E_0} \widehat{c}_{(u,v)}(\widehat{x}_u, \widehat{x}_v, w_{(u,v)}) + \sum_{(u,v)\in E_1} \widehat{c}_{(u,v)}(\widehat{x}_u, y_v, w_{(u,v)}).$$

[1] We add a hat over the decision vector $\widehat{x}_v$ to distinguish it with the local action $x_t^v$ and global action $x_t$ defined in Section 3.2. We assume $\widehat{x}_v$ is a $p_v$ dimensional real vector.

[2] We do not consider the disturbance vectors in the exponentially decaying local perturbation bounds defined in Definition 3.3.1, but adding $w$ into the edge costs makes Theorem A.2.1 more general. For each edge $e$, $w_e$ is a $q$-dimensional real vector.

*Then, we have that for any vertex $u_0 \in \mathcal{V}' \setminus S$, the following inequality holds:*

$$\|\psi(w,y)_{u_0} - \psi(w',y')_{u_0}\| \leq C\left(\sum_{e\in E_0\cup E_1} \lambda^{d_{\mathcal{G}'}(h,e)-1}\|w_e - w'_e\| + \sum_{v\in S}\lambda^{d_{\mathcal{G}'}(h,v)-1}\|y_v - y'_v\|\right), \forall w, y, w', y',$$

*where $C \coloneqq (2\ell)/\mu$ and $\lambda \coloneqq 1 - 2\cdot\left(\sqrt{1+(\Delta'\ell/\mu)}+1\right)^{-1}$. Here, $\Delta'$ denote the maximum degree of any vertex $v \in \mathcal{V}'$ in graph $\mathcal{G}'$. For $e = (u,v) \in \mathcal{E}'$, we define $d_{\mathcal{G}'}(u_0, e) \coloneqq \min\{d_{\mathcal{G}'}(u_0,u), d_{\mathcal{G}'}(u_0,v)\}$.*

*Proof of Theorem A.2.1.* Let $e = [\pi^\top, \epsilon^\top]^\top$ be a vector where $\epsilon = \{\epsilon_v\}_{v\in S}$ for $\epsilon_v \in \mathbb{R}^{p_v}$ and $\pi = \{\pi_e\}_{e\in E_0\cup E_1}$, for $\pi_e \in \mathbb{R}^q$. Let $\theta$ be an arbitrary real number. Define function $\widehat{h} : \mathbb{R}^{\sum_{v\in\mathcal{V}'\setminus S} p_v} \times \mathbb{R}^{(|E_0|+|E_1|)\times q} \times \mathbb{R}^{\sum_{v\in S} p_v} \to \mathbb{R}_{\geq 0}$ as

$$\widehat{h}(\widehat{x}, w, y) = \sum_{v\in\mathcal{V}'\setminus S}\widehat{f}_v(\widehat{x}_v) + \sum_{(u,v)\in E_0}\widehat{c}_{(u,v)}(\widehat{x}_u, \widehat{x}_v, w_{(u,v)}) + \sum_{(u,v)\in E_1}\widehat{c}_{(u,v)}(\widehat{x}_u, y_v, w_{(u,v)}).$$

To simplify the notation, we use $\zeta$ to denote the tuple of system parameters, i.e.,

$$\zeta := (w, y).$$

From our construction, we know that $\widehat{h}$ is $\mu$-strongly convex in $x$, so we use the decomposition $\widehat{h} = \widehat{h}_a + \widehat{h}_b$, where

$$\widehat{h}_a(\widehat{x},\zeta) = \sum_{v\in\mathcal{V}'\setminus S}\frac{\mu}{2}\|\widehat{x}_v\|^2 + \sum_{(u,v)\in E_0}\widehat{c}_{(u,v)}(\widehat{x}_u, \widehat{x}_v, w_{(u,v)}) + \sum_{(u,v)\in E_1}\widehat{c}_{(u,v)}(\widehat{x}_u, y_v, w_{(u,v)}),$$
$$\widehat{h}_b(\widehat{x},\zeta) = \sum_{v\in\mathcal{V}'\setminus S}\left(\widehat{f}_v(\widehat{x}_v) - \frac{\mu}{2}\|\widehat{x}_v\|^2\right).$$

Since $\psi(\zeta+\theta e)$ is the minimizer of convex function $\widehat{h}(\cdot, \zeta+\theta e)$, we see that

$$\nabla_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e), \zeta+\theta e) = 0.$$

Taking the derivative with respect to $\theta$ gives that

$$\nabla^2_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)\frac{d}{d\theta}\psi(\zeta+\theta e) = -\sum_{v\in S}\nabla_{y_v}\nabla_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)\epsilon_v - \sum_{e\in E_1\cup E_2}\nabla_{w_e}\nabla_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)\pi_e.$$

To simplify the notation, we define

$$\begin{aligned}
M &:= \nabla^2_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e), \text{which is a } |\mathcal{V}'\setminus S|\times|\mathcal{V}'\setminus S| \text{ block matrix},\\
R^{(v)} &:= -\nabla_{y_v}\nabla_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e), \forall v\in S, \text{which are } |\mathcal{V}'\setminus S|\times 1 \text{ block matrix},\\
K^{(e)} &:= -\nabla_{w_e}\nabla_{\widehat{x}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e), \forall e\in E_0\cup E_1, \text{which are } |\mathcal{V}'\setminus S|\times 1 \text{ block matrices},
\end{aligned}$$

where in $M$, the block size is $p_u\times p_v, \forall(u,v)\in(\mathcal{V}'\setminus S)^2$; in $R^{(v)}$, the block size is $p_u\times p_v, \forall u\in\mathcal{V}'\setminus S$; in $K^{(e)}$, the block size is $p_u\times q, \forall u\in\mathcal{V}'\setminus S$. Hence we can write

$$\frac{d}{d\theta}\psi(\zeta+\theta e) = M^{-1}\left(\sum_{v\in S}R^{(v)}\epsilon_v + \sum_{e\in E_1\cup E_2}K^{(e)}\pi_e\right).$$

Recall that $\{R^{(v)}\}_{v\in S}$ are $|\mathcal{V}'\setminus S|\times 1$ block matrices with block size $p_u\times p_v, \forall u\in\mathcal{V}'\setminus S$; $\{K^{(e)}\}_{e\in E_0\cup E_1}$ are $|\mathcal{V}'\setminus S|\times 1$ block matrices with block size $p_u\times q, \forall u\in\mathcal{V}'\setminus S$. Let $N(v)$ denote the set of neighbors of vertex $v$ on $\mathcal{G}'$. For $R^{(v)}, v\in S$, the $(u,1)$-th block can be non-zero only if $u\in(\mathcal{V}'\setminus S)\cap N(v)$. For $K^{(e)}, e\in E_0\cup E_1$, the $(u,1)$-th block can be non-zero only if $u\in e$ and $u\in\mathcal{V}'\setminus S$. Hence we see that

$$\frac{d}{d\theta}\psi(\zeta+\theta e)_{u_0} = \sum_{v\in S}(M^{-1})_{u_0,(\mathcal{V}'\setminus S)\cap N(v)}R^{(v)}_{(\mathcal{V}'\setminus S)\cap N(v),1}\epsilon_v + \sum_{e\in E_0\cup E_1}(M^{-1})_{u_0,\{u\in e|u\in\mathcal{V}'\setminus S\}}K^{(\tau)}_{\{u\in e|u\in\mathcal{V}'\setminus S\},1}\pi_e.$$

Since the switching costs $c_\tau(\cdot,\cdot,\cdot), \tau=1,\ldots,p$ are $\ell$-strongly smooth, we know that the norms of

$$R^{(v)}_{(\mathcal{V}'\setminus S)\cap N(v),1}, \text{ and } K^{(\tau)}_{\{u\in e|u\in\mathcal{V}'\setminus S\},1}$$

are all upper bounded by $\ell$. Taking norms on both sides gives that

$$\left\| \frac{d}{d\theta} \psi(\zeta + \theta e)_{u_0} \right\| \leq \sum_{v \in S} \ell \left\| (M^{-1})_{u_0, (\mathcal{V}' \setminus S) \cap N(v)} \right\| \|\epsilon_v\| + \sum_{e \in E_0 \cup E_1} \ell \left\| (M^{-1})_{u_0, \{u \in e | u \in \mathcal{V}' \setminus S\}} \right\| \|\pi_e\| . \tag{A.1}$$

Note that $M$ can be decomposed as $M = M_a + M_b$, where

$$M_a := \nabla^2_{\widehat{x}} \widehat{h}_a(\psi(\zeta + \theta e), \zeta + \theta e),$$
$$M_b := \nabla^2_{\widehat{x}} \widehat{h}_b(\psi(\zeta + \theta e), \zeta + \theta e).$$

Since $M_a$ is block tri-diagonal and satisfies $(\mu + \Delta' \ell) I \succeq M_a \succeq \mu I$ , and $M_b$ is block diagonal and satisfies $M_b \succeq 0$, we obtain the following using Lemma A.2.1:

$$\left\| (M^{-1})_{u_0, (\mathcal{V}' \setminus S) \cap N(v)} \right\| \leq \frac{2}{\mu} \lambda^{d_{\mathcal{G}'}(u_0, v) - 1}, \text{ and } \left\| (M^{-1})_{u_0, \{u \in e | u \in \mathcal{V}' \setminus S\}} \right\| \leq \frac{2}{\mu} \lambda^{d_{\mathcal{G}'}(u_0, e) - 1},$$

where $\lambda := (\sqrt{cond(M_a)} - 1)/(\sqrt{cond(M_a)} + 1) = 1 - 2 \cdot \left( \sqrt{1 + (2\ell/\mu)} + 1 \right)^{-1}$.

Substituting this into (A.1), we see that

$$\left\| \frac{d}{d\theta} \psi(\zeta + \theta e)_{u_0} \right\| \leq C \left( \sum_{v \in S} \lambda^{d_{\mathcal{G}'}(u_0, v) - 1} \|\epsilon_v\| + \sum_{e \in E_0 \cup E_1} \lambda^{d_{\mathcal{G}'}(u_0, e) - 1} \|\pi_e\| \right),$$

where $C = (2\ell)/\mu$.

Finally, by integration we can complete the proof

$$\begin{aligned} \|\psi(\zeta)_{u_0} - \psi(\zeta + e)_{u_0}\| &= \left\| \int_0^1 \frac{d}{d\theta} \psi(\zeta + \theta e)_{u_0} d\theta \right\| \\ &\leq \int_0^1 \left\| \frac{d}{d\theta} \psi(\zeta + \theta e)_{u_0} \right\| d\theta \\ &\leq C \left( \sum_{v \in S} \lambda^{d_{\mathcal{G}'}(u_0, v) - 1} \|\epsilon_v\| + \sum_{e \in E_0 \cup E_1} \lambda^{d_{\mathcal{G}'}(u_0, e) - 1} \|\pi_e\| \right). \end{aligned}$$

Q.E.D.

Now we return to the proof of Theorem 3.3.1. For simplicity, we temporarily assume the individual decision points are unconstrained, i.e., $D_t^v = \mathbb{R}^n$. We discuss how to relax this assumption in Appendix A.2.3.

We first consider the case when $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ and $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ only differ at one entry $y_{t-1}^u$ or $z_\tau^u$. If the difference is at $z_\tau^u$, by viewing each subset $\{\tau\} \times N_v^r$ for $\tau \in \{t-1, t, \ldots, t+k\}$ as a vertex in the new graph $\mathcal{G}'$ and applying Theorem A.2.1, we obtain that

$$\left\|x_{t_0}^{v_0} - (x_{t_0}^{v_0})'\right\| \leq C_1^0 \cdot (\rho_T^0)^{|t_0-\tau|} \left\|z_\tau^u - (z_\tau^u)'\right\|, \tag{A.2}$$

where $C_1^0 = (2\ell_T)/\mu$ and $\rho_T^0 = 1 - 2 \cdot \left(\sqrt{1 + (2\ell_T/\mu)} + 1\right)^{-1}$. On the other hand, by viewing each subset $\{\tau \mid t-1 \leq \tau < t+k\} \times \{u\}$ for $u \in N_v^r$ as a vertex in the new graph $\mathcal{G}'$ and applying Theorem A.2.1, we obtain that

$$\left\|x_{t_0}^{v_0} - (x_{t_0}^{v_0})'\right\| \leq C_1^1 \cdot (\rho_S^0)^{\text{dist}(u,v_0)} \left\|z_\tau^u - (z_\tau^u)'\right\|, \tag{A.3}$$

where $C_1^1 = (2\Delta\ell_S)/\mu$ and $\rho_S^0 = 1 - 2 \cdot \left(\sqrt{1 + (2\Delta\ell_S/\mu)}\right)^{-1}$. Combining (A.2) and (A.3) gives that

$$\begin{aligned}
\left\|x_{t_0}^{v_0} - (x_{t_0}^{v_0})'\right\| &\leq \min\{C_1^0 \cdot (\rho_T^0)^{|t_0-\tau|}, C_1^1 \cdot (\rho_S^0)^{\text{dist}(u,v_0)}\} \cdot \left\|z_\tau^u - (z_\tau^u)'\right\| \\
&\leq \sqrt{C_1^0 \cdot C_1^1} \cdot (\rho_T^0)^{|t_0-\tau|/2} \cdot (\rho_S^0)^{\text{dist}(u,v_0)/2} \cdot \left\|z_\tau^u - (z_\tau^u)'\right\| \\
&\leq C_1 \cdot \rho_T^{|t_0-\tau|} \rho_S^{\text{dist}(v_0,u)} \left\|z_\tau^u - (z_\tau^u)'\right\|
\end{aligned} \tag{A.4}$$

when $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ and $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ only differ at one entry $z_\tau^u$ for $(\tau, u) \in \partial N_{(t,v)}^{(k,r)}$.

We can use the same method to show that when $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ and $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ only differ

at one entry $y_{t-1}^u$ for $u \in N_v^r$, we have

$$\left\| x_{t_0}^{v_0} - (x_{t_0}^{v_0})' \right\| \leq C_2 \rho_T^{t_0-(t-1)} \rho_S^{\text{dist}(v_0,u)} \left\| y_{t-1}^u - (y_{t-1}^u)' \right\|. \tag{A.5}$$

In the general case where $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ and $\left(\{y_{t-1}^u\}, \{z_\tau^u\}\right)$ differ not only at one entry, we can perturb the entries of parameters one at a time and apply the triangle inequality. Then, the conclusion of Theorem 3.3.1 follows from (A.4) and (A.5).

### A.2.2 Proof of Theorem 3.3.2

The proof follows a four-step structure outlined in Section 3.4.1.

**Step 1. Establish first order equations**

Given any system parameter $\zeta = (x_{t-1}^{(N_v^r)}, \{z_\tau^u | (\tau, u) \in \partial N_{(t,v)}^{(k,r)}\})$, we can define function $\widehat{h}$ as follows:

$$\widehat{h}(\widehat{x}_{[k-1]}, \zeta) = \sum_{i=1}^{k-1} \sum_{u \in N_v^{r-1}} f_{t-1+i}^u(\widehat{x}_i^u) + \sum_{i=1}^{k-1} \sum_{(u,u') \in \mathcal{E}(N_v^r)} s_{t-1+i}^{(u,u')}(\widehat{x}_i^u, \widehat{x}_i^{u'}) + \sum_{i=1}^{k} \sum_{u \in N_v^r} c_{t-1+i}^u(\widehat{x}_i^u, \widehat{x}_{i-1}^u).$$

$\widehat{x}_0$ coincides with $x_{t-1}$ on every node in $N_v^r$. $\widehat{x}_k$ coincides with $z_{t-1+k}$ on every node in $N_v^r$. For $1 \leq i \leq k-1$, $\widehat{x}_i^u$ coincides with $z_{t-1+i}^u$ on the boundary, i.e., $u \in \partial N_v^r$.

Let perturbation vector $e = [e_0^T, e_1^T, \cdots, e_{k-1}^T, e_k^T]^T$ where $e_0, e_k \in \mathbb{R}^{|N_v^r| \times n}$ and $e_i = \mathbb{R}^{|\partial N_v^r| \times n}$ for $1 \leq i \leq k-1$.

Given $\theta \in \mathbb{R}$, $\psi(\zeta + \theta e)$ is the global minimizer of convex function $\widehat{h}(\cdot, \zeta + \theta e)$, and hence we have

$$\nabla_{\widehat{x}_{1:k-1}^{(N_v^{r-1})}} \widehat{h}(\psi(\zeta + \theta e), \zeta + \theta e) = 0.$$

Taking the derivative with respect to $\theta$, we establish the following set of equations:

$$\begin{aligned}\nabla^2_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)\frac{d}{d\theta}\psi(\zeta+\theta e) = &-\nabla_{\widehat{x}^{(N_v^r)}_0}\nabla_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)e_0\\ &-\nabla_{\widehat{x}^{(N_v^r)}_k}\nabla_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)e_k\\ &-\sum_{\tau=1}^{k-1}\nabla_{\widehat{x}^{(\partial N_v^r)}_\tau}\nabla_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)e_\tau.\end{aligned} \tag{A.6}$$

We adopt the following short-hand notation:

- $M \coloneqq \nabla^2_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)$, which is a hierarchical block matrix with the first level of dimension $(k-1)\times(k-1)$, the second level of dimension $|N_v^{r-1}|\times|N_v^{r-1}|$ and the third level of dimension $n\times n$.
- $R^{(1)} \coloneqq -\nabla_{\widehat{x}^{(N_v^r)}_0}\nabla_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)$, which is also a hierarchical block matrix with the first level of dimension $(k-1)\times 1$, the second level of dimension $|N_v^{r-1}|\times|N_v^r|$ and the third level of dimension $n\times n$.
- $R^{(k-1)} \coloneqq -\nabla_{\widehat{x}^{(N_v^r)}_k}\nabla_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)$, which is also a hierarchical block matrix with the first level of dimension $(k-1)\times 1$, the second level of dimension $|N_v^{r-1}|\times|N_v^r|$ and the third level of dimension $n\times n$.
- $K^{(\tau)} \coloneqq -\nabla_{\widehat{x}^{(\partial N_v^r)}_\tau}\nabla_{\widehat{x}^{(N_v^{r-1})}_{1:k-1}}\widehat{h}(\psi(\zeta+\theta e),\zeta+\theta e)$, which is also a hierarchical block matrix with the first level of dimension $(k-1)\times 1$. the second level of dimension $|N_v^{r-1}|\times|\partial N_v^r|$ and the third level of dimension $n\times n$.

Using the above, we can rewrite (A.6) as follows:

$$\frac{d}{d\theta}\psi(\zeta+\theta e) = M^{-1}\left(R^{(1)}e_0 + R^{(k-1)}e_k + \sum_{\tau=1}^{k-1}K^{(\tau)}y_\tau\right).$$

Due to the structure of temporal interaction cost functions, for $R^{(1)}$(resp. $R^{(k-1)}$), only when the first level index is 1 (resp. $k-1$), the lower level block matrix is non-zero; due to the structure

of spatial interaction cost functions, for $K^{(\tau)}$, only when the first level index is $\tau$, the lower level block matrix is non-zero. Hence, for $1 \leq \tau' \leq k-1$, we have

$$(\frac{d}{d\theta}\psi(\zeta+\theta e))_{\tau'} = M^{-1}_{\tau',1}R^{(1)}_1 e_0 + M^{-1}_{\tau',k-1}R^{(k-1)}_{k-1}e_k + \sum_{\tau=1}^{k-1} M^{-1}_{\tau',\tau}K^{(\tau)}_{\tau}y_\tau, \tag{A.7}$$

where the subscripts on the right hand side denote the first level index of hierarchical block matrices $M$, $R^{(1)}$, $R^{(k-1)}$ and $K^{(\tau)}$.

**Step 2. Exploit the structure of matrix $M$**

We decompose $M$ to block diagonal matrix $D$ and tri-diagonal block matrix $A$ such that $M = D + A$. We denote each diagonal block in $D$ as $D_{i,i}$ for $1 \leq i \leq k-1$. Other blocks in $D$ are zero matrices.

$$D := \begin{bmatrix}
* & 0 & \cdots & * & & & & & & & & & & & \\
0 & * & & 0 & & & & & & & & & & & \\
\vdots & & \ddots & & & & & & & & & & & & \\
* & 0 & \cdots & * & & & & & & & & & & & \\
 & & & & * & 0 & \cdots & * & & & & & & & \\
 & & & & 0 & * & & 0 & & & & & & & \\
 & & & & \vdots & & \ddots & & & & & & & & \\
 & & & & * & 0 & \cdots & * & & & & & & & \\
 & & & & & & & & \ddots & & & & & & \\
 & & & & & & & & & * & 0 & \cdots & * & \\
 & & & & & & & & & 0 & * & & 0 & \\
 & & & & & & & & & \vdots & & \ddots & & \\
 & & & & & & & & & * & 0 & \cdots & * &
\end{bmatrix}$$

Each non-zero block in $A$ is a diagonal block matrix, which captures the Hessian of temporal interaction cost between consecutive time steps. Denote each block as $A_{i,j}$ for $1 \leq i, j \leq k-1$.

$$A \coloneqq \begin{bmatrix}
 & & & & * & 0 & \cdots & 0 & & & & & & & & \\
 & & & & 0 & * & & 0 & & & & & & & & \\
 & & & & \vdots & & \ddots & & & & & & & & & \\
 & & & & 0 & 0 & \cdots & * & & & & & & & & \\
* & 0 & \cdots & 0 & & & & & * & 0 & \cdots & 0 & & & & \\
0 & * & & 0 & & & & & 0 & * & & 0 & & & & \\
\vdots & & \ddots & & & & & & \vdots & & \ddots & & & & & \\
0 & 0 & \cdots & * & & & & & 0 & 0 & \cdots & * & & & & \\
 & & & & & & & & & & & & * & 0 & \cdots & 0 \\
 & & & & & & & & & & & & 0 & * & & 0 \\
 & & & & & & & & & & \ddots & & \vdots & & \ddots & \\
 & & & & & & & & & & & & 0 & 0 & \cdots & * \\
 & & & & & & & & * & 0 & \cdots & 0 & & & & \\
 & & & & & & & & 0 & * & & 0 & & & & \\
 & & & & & & & & \vdots & & \ddots & & & & & \\
 & & & & & & & & 0 & 0 & \cdots & * & & & &
\end{bmatrix}$$

We rewrite the inverse of $M$ as follows:

$$M^{-1} = (D + A)^{-1} = D^{-1}(I + AD^{-1})^{-1} = D^{-1}P^{-1}.$$

Next we show the proof for Lemma 3.4.2.

*Proof of Lemma 3.4.2.* We claim the eigenvalues of $P$ are in $\{\lambda \in \mathbb{C} || \lambda - z| \leq R\}$ for some $R \in \mathbb{R}_{>0}$ and $z \in \mathbb{C} \setminus \{0\}$ such that $R < |z|$. We first establish Lemma 3.4.2 based on the claim and then prove the claim.

We follow the argument as in the proof of Thm 4 in Shin, Zavala, and Anitescu, 2020. Since any eigenvalue $\lambda$ of $P$ satisfies $|\lambda - z| \leq R$, $|\lambda/z - 1| \leq R/|z| < 1$. Thus, the eigenvalues of $I - (1/z)P$ lie on $\{\widetilde{\lambda} \in \mathbb{C} : |\widetilde{\lambda}| \leq R/|z|\}$, which guarantees $\rho(I - (1/z)P) < 1$. Therefore,

$$P^{-1} = \frac{1}{z}\left(I - (I - \frac{1}{z}P)\right)^{-1} = \frac{1}{z}\sum_{q\geq 0}(I - \frac{1}{z}P)^q.$$

We let $z = 1$ and $R = \frac{2\ell_T}{\mu}$ and prove the above claim by utilizing Gershgorin circle theorem for block matrices.

By Theorem 1.13.1 and Remark 1.13.2 of Tretter, 2008, the following holds: Consider $\mathcal{A} = (A_{ij}) \in \mathbb{R}^{dn\times dn}$ $(d, n \geq 1)$ where $A_{ij} \in \mathbb{R}^{d\times d}$ and $A_{ii}$ is symmetric. Suppose $\sigma(\cdot)$ is the spectrum of a matrix. Define set

$$G_i \coloneqq \sigma(A_{ii}) \cup \left\{ \cup_{k=1}^{d} B\left(\lambda_k(A_{ii}), \sum_{j\neq i} \|A_{ij}\|\right)\right\}$$

where $B(\cdot,\cdot)$ denotes a disk $B(c, r) = \{\lambda : \|\lambda - c\| \leq r\}$ and $\lambda_k$ is the $k$-th smallest eigenvalues of $A_{ii}$. Then,

$$\sigma(\mathcal{A}) \in \cup_{i=1}^{n} G_i.$$

Next, we use the above fact to find a superset of $\sigma(P)$. Every diagonal block of $P$ is $I$. Moreover, $P_{i,j} = 0$ for $|i - j| > 1$, $P_{i,i-1} = A_{i,i-1}D_{i-1,i-1}^{-1}$, $P_{i,i+1} = A_{i,i+1}D_{i+1,i+1}^{-1}$. Hence we have

$$\begin{aligned}\sum_{j\neq i} \|P_{i,j}\| &\leq \|A_{i,i-1}\| \left\|D_{i-1,i-1}^{-1}\right\| + \|A_{i,i+1}\| \left\|D_{i+1,i+1}^{-1}\right\| \\ &\leq \frac{2\ell_T}{\mu}.\end{aligned}$$

The last inequality is by Assumptions 3.2.1. Therefore, $G_i = B(1, \frac{2\ell_T}{\mu})$. This implies all eigenvalues of $P$ are in $B(1, \frac{2\ell_T}{\mu})$. Q.E.D.

To further simplify the notation in the power series expansion, we define $J \coloneqq AD^{-1} = P - I$. Given any time indices $\tau'$ and $\tau$, we have

$$\begin{aligned}(M^{-1})_{\tau',\tau} &= (D^{-1})_{\tau',\tau'}(P^{-1})_{\tau',\tau} \\ &= (D^{-1})_{\tau',\tau'} \times \sum_{\ell \geq 0} (-J)^{\ell}_{\tau',\tau},\end{aligned} \tag{A.8}$$

where the first equality is since $D^{-1}$ is a diagonal block matrix, the second equality is due to Lemma 3.4.2.

**Step 3: Property for general exponential-decay matrices**

This step simply requires proving Lemma 3.4.3.

*Proof of Lemma 3.4.3.* Under the assumptions, we see that

$$\begin{aligned}\sum_{q} (\frac{1}{\lambda'})^{d_{\mathcal{M}}(u,q)} \|(A_1 A_2 \cdots A_\ell)_{u,q}\| &= \sum_{q} (\frac{1}{\lambda'})^{d_{\mathcal{M}}(u,q)} \left\| \sum_{s_1,\cdots,s_{\ell-1}} (A_1)_{u,s_1}(A_2)_{s_1,s_2} \cdots (A_\ell)_{s_{\ell-1},q} \right\| \\ &\leq \sum_{q} (\frac{1}{\lambda'})^{d_{\mathcal{M}}(u,q)} \sum_{s_1,\cdots,s_{\ell-1}} (C_1 \lambda^{d_{\mathcal{M}}(u,s_1)})(C_2 \lambda^{d_{\mathcal{M}}(s_1,s_2)}) \cdots (C_\ell \lambda^{d_{\mathcal{M}}(s_{\ell-1},q)}) \\ &\leq \sum_{q} \sum_{s_1,\cdots,s_{\ell-1}} \prod_{i=1}^{\ell} C_i (\frac{\lambda}{\lambda'})^{d_{\mathcal{M}}(u,s_1)+d_{\mathcal{M}}(s_1,s_2)+\cdots+d_{\mathcal{M}}(s_{\ell-1},q)} \\ &\leq (\widetilde{a})^{\ell} \prod_{i=1}^{\ell} C_i.\end{aligned} \tag{A.9}$$

Hence, we obtain that

$$\left\| (\prod_{i=1}^{\ell} A_i)_{u,q} \right\| \leq C' (\lambda')^{d_M(u,q)}.$$

Q.E.D.

**Step 4: Establish correlation decay properties of matrix $M$**

In this step, we use the property developed for general exponential-decay matrices on $M$ and derive the perturbation bound in the Theorem 3.3.2.

**Lemma A.2.2.** *For $\ell \geq 1$, time index $i, j \geq 1$, $J^\ell$ has the following properties:*

- $(J^\ell)_{i,j} = 0$ *if* $\ell < |i-j|$ *or* $\ell - |i-j|$ *is odd.*
- $(J^\ell)_{i,j}$ *is a summation of terms* $\prod_{k=1}^{\ell} A_{j_k,i_k} D^{-1}_{i_k,i_k}$ *and the number of such terms is bounded by* $\binom{\ell}{(\ell-|i-j|)/2}$.

Note for integers $m, k \geq 1$, we define $\binom{m}{k/2} = 0$ if $k$ is odd.

*Proof.* Since $J$ is a tri-diagonal banded matrix, $J^\ell_{i,j} = 0$ for $\ell < |i-j|$. We prove the rest of properties of $J$ by induction on $\ell$. When $\ell = 1$,

$$J_{i,i} = 0, \quad J_{i,i-1} = A_{i,i-1} D^{-1}_{i-1,i-1}, \quad J_{i,i+1} = A_{i,i+1} D^{-1}_{i+1,i+1}.$$

Lemma A.2.2 holds for the base case. Suppose Lemma A.2.2 holds for $J^q$ for $q \leq \ell - 1$. Let $q = \ell$, then

$$J^\ell_{i,j} = \sum_k J^{\ell-1}_{i,k} J_{k,j} = J^{\ell-1}_{i,j-1} A_{j-1,j} D^{-1}_{j,j} + J^{\ell-1}_{i,j+1} A_{j+1,j} D^{-1}_{j,j}.$$

By induction hypothesis, $J^{\ell-1}_{i,j}$ is a summation of terms $\prod_{k=1}^{\ell-1} A_{j_k,i_k} D^{-1}_{i_k,i_k}$. Moreover, the number of such terms is bounded by $\binom{\ell-1}{(\ell-1-|i-j-1|)/2} + \binom{\ell-1}{(\ell-1-|i-j+1|)/2}$. Next we will show $\binom{\ell-1}{(\ell-1-|i-j-1|)/2} + \binom{\ell-1}{(\ell-1-|i-j+1|)/2} = \binom{\ell}{(\ell-|i-j|)/2}$ case by case.

**Case 1:** $\ell - |i-j|$ is odd.

If $\ell - |i-j|$ is odd, then $\ell - 1 - |i-j-1|$ and $\ell - 1 - |i-j+1|$ are both odd. Under this case,

$$\binom{\ell-1}{(\ell-1-|i-j-1|)/2} + \binom{\ell-1}{(\ell-1-|i-j+1|)/2} = 0,$$

which is equal to $\binom{\ell}{(\ell-|i-j|)/2}$.

**Case 2:** $\ell - |i-j|$ is even and $i = j$. Under this case, we have

$$\binom{\ell-1}{(\ell-1-|i-j-1|)/2} + \binom{\ell-1}{(\ell-1-|i-j+1|)/2} = \binom{\ell-1}{\ell/2-1} + \binom{\ell-1}{\ell/2-1}.$$

Since $\ell$ is even, $\binom{\ell-1}{\ell/2-1} + \binom{\ell-1}{\ell/2-1} = \binom{\ell}{\ell/2} = \binom{\ell}{(\ell-|i-j|)/2}$.

**Case 3:** $\ell - |i-j|$ is even and $i \neq j$.

If $\ell - |i-j|$ is even, then $\ell - 1 - |i-j-1|$ and $\ell - 1 - |i-j+1|$ are both even. We denote $(\ell - |i-j|)/2$ as $k_0$. By triangle inequality, $(\ell - 1 - |i-j-1|)/2$ and $(\ell - 1 - |i-j+1|)/2$ are in $\{k_0 - 1, k_0\}$. Since $i \neq j$,

$$\binom{\ell-1}{(\ell-1-|i-j-1|)/2} + \binom{\ell-1}{(\ell-1-|i-j+1|)/2} = \binom{\ell-1}{k_0-1} + \binom{\ell-1}{k_0},$$

which sums to $\binom{\ell}{k_0}$ by Pascal's triangle.

Q.E.D.

Next we present the proof of Lemma 3.4.4.

*Proof of Lemma 3.4.4.* By Lemma A.2.2, $(J^\ell)_{i,j}$ is a summation of terms $\prod_{k=1}^{\ell} A_{j_k,i_k} D^{-1}_{i_k,i_k}$ and the number of such terms is bounded by $\binom{\ell}{(\ell-|i-j|)/2}$.

Define $B_k \coloneqq A_{j_k,i_k} D^{-1}_{i_k,i_k}$. Recall $A_{j_k,i_k}$ is a diagonal matrix and $D_{i_k,i_k}$ is a graph-induced banded matrix.

$$\|(B_k)_{u,q}\| = \left\|(A_{j_k,i_k} D^{-1}_{i_k,i_k})_{u,q}\right\| = \left\|(A_{j_k,i_k})_{u,u}(D^{-1}_{i_k,i_k})_{u,q}\right\| \leq \ell_T \left\|(D^{-1}_{i_k,i_k})_{u,v}\right\| \leq \frac{2\ell_T}{\mu}\gamma_S^{\mathsf{dist}(u,v)}.$$

where the last inequality is by using Lemma A.2.1 on $D_{i_k,i_k}$.

Under the condition $b < \infty$, we can use Lemma 3.4.3 to obtain the following bound,

$$\left\| (\prod_{k=1}^{\ell} A_{j_k,i_k} D_{i_k}^{-1})_{u,v} \right\| \leq (b\frac{2\ell_T}{\mu})^{\ell} (\gamma_S')^{\text{dist}(u,v)}.$$

Since the number of such terms is bounded by $\binom{\ell}{(\ell-|i-j|)/2}$, we have

$$\left\| ((J^{\ell})_{i,j})_{u,q} \right\| \leq \binom{\ell}{(\ell-|i-j|)/2} (b\frac{2\ell_T}{\mu})^{\ell} (\gamma_S')^{\text{dist}(u,v)}.$$

Q.E.D.

**Lemma A.2.3.** *Given* $1 \leq \tau', \tau \leq k-1$, $y \in \mathbb{R}^{|\partial N_v^r| \times n}$ *and* $v_0 \in N_v^{r-1}$, *we have*

$$\left\| \left( (M)_{\tau',\tau}^{-1} K_{\tau}^{(\tau)} y \right)_{v_0} \right\| \leq C_1 \rho_T^{|\tau'-\tau|} \sum_{u \in \partial N_v^r} \rho_S^{\text{dist}(v_0,u)-1} \left\| y_u \right\|,$$

*and for* $i \in \{1, k-1\}$, $e \in \mathbb{R}^{|N_v^r| \times n}$,

$$\left\| \left( (M^{-1})_{\tau',i}) R_i^{(i)} e \right)_{v_0} \right\| \leq C_2 \rho_T^{|\tau'-i|+1} \sum_{u \in N_v^r} \rho_S^{\text{dist}(v_0,u)} \left\| e_u \right\|,$$

*where* $\rho_T = \frac{4\widetilde{a}\ell_T}{\mu}$ *and* $\rho_S = (1+b_1+b_2)\gamma_S$. *We let* $C_1 = C_2 = \max\{\frac{a^2}{2\widetilde{a}(1-4\widetilde{a}\ell_T/\mu)}, \frac{2a^2\Delta\ell_S/\mu}{\gamma_S(1+b_1+b_2)(1-4\widetilde{a}\ell_T/\mu)}\}$.

*Proof.* Given $1 \leq \tau, \tau' \leq k-1$ and $v_0 \in N_v^{r-1}$, since $M^{-1} = D^{-1} \sum_{\ell \geq 0} (-J)^{\ell}$, we have

$$\left\| \left( (M)_{\tau',\tau}^{-1} K_{\tau}^{(\tau)} y \right)_{v_0} \right\| = \left\| \left( D_{\tau',\tau'}^{-1} \sum_{\ell \geq 0} (-J)_{\tau',\tau}^{\ell} K_{\tau}^{(\tau)} y \right)_{v_0} \right\|. \tag{A.10}$$

With slight abuse of notation, we use $K$ to denote $K_{\tau}^{(\tau)}$, and $Q^{-1}$ to denote $D_{\tau',\tau'}^{-1}$ in this proof

from now. We can rewrite the right hand side of (A.10) using the new notation as follows:

$$\begin{aligned}\left\|\left(Q^{-1}\sum_{\ell\geq 0}(-J)^{\ell}_{\tau',\tau}Ky\right)_{v_0}\right\| &\leq \sum_{\ell\geq 0}\left\|\left(Q^{-1}(-J)^{\ell}_{\tau',\tau}Ky\right)_{v_0}\right\| \\ &= \sum_{\ell\geq 0}\left\|\sum_{q\in N_v^{r-1}}\left(Q^{-1}(-J)^{\ell}_{\tau',\tau}\right)_{v_0,q}(Ky)_q\right\| \\ &\leq \sum_{\ell\geq 0}\sum_{q\in N_v^{r-1}}\left\|\left(Q^{-1}(-J)^{\ell}_{\tau',\tau}\right)_{v_0,q}\right\|\|(Ky)_q\|\,.\end{aligned} \tag{A.11}$$

For a given $q\in N_v^{r-1}$ and $y\in\mathbb{R}^{|\partial N_v^r|d}$,

$$\|(Ky)_q\| = \left\|\sum_{u\in\partial N_v^r}K_{q,u}y_u\right\| = \left\|\sum_{u\in\partial N_v^r\cap N_q^1}K_{q,u}y_u\right\|.$$

where the last equality is since spacial interaction costs are only among neighboring nodes.

For a given $u\in\partial N_v^r$, since the spacial interaction cost for each edge is $\ell_S$ smooth,

$$\|K_{q,u}y_u\|\leq\|K_{q,u}\|\,\|y_u\|\leq\ell_S\,\|y_u\|\,,$$

which gives

$$\|(Ky)_q\|\leq\sum_{u\in\partial N_v^r\cap N_q^1}\ell_S\,\|y_u\|\,.$$

Therefore,

$$\left\|\left(Q^{-1}\sum_{\ell\geq 0}(-J)^{\ell}_{\tau',\tau}Ky\right)_{v_0}\right\|\leq\ell_S\sum_{\ell\geq 0}\sum_{q\in N_v^{r-1}}\left\|\left(Q^{-1}(-J)^{\ell}_{\tau',\tau}\right)_{v_0,q}\right\|\sum_{u\in\partial N_v^r\cap N_q^1}\|y_u\|\,. \tag{A.12}$$

By Lemma 3.4.4, $(-J)^{\ell}_{\tau',\tau}$ satisfies the following exponential decay properties: for any $u,q\in N_v^{r-1}$,

$$\left\|((J^{\ell})_{\tau',\tau})_{u,q}\right\|\leq\binom{\ell}{(\ell-|\tau'-\tau|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^{\ell}(\gamma_S')^{\text{dist}(u,q)},$$

where we choose $\delta=b_1\cdot\gamma_S$, $\gamma_S'=(1+b_1)\gamma_S$ and $\widetilde{a}=\sum_{\gamma\geq 0}(\frac{1}{1+b_1})^{\gamma}h(\gamma)$.

Moreover, $Q^{-1}$ (which denotes $D^{-1}_{\tau',\tau'}$) is the inverse of a graph-induced banded matrix. $Q^{-1}$ satisfies: for any $u, q \in N_v^{r-1}$,

$$\left\|(Q^{-1})_{u,q}\right\| \leq \frac{2}{\mu}\gamma_S^{\text{dist}(u,q)} < \frac{2}{\mu}(\gamma'_S)^{\text{dist}(u,q)},$$

where the first inequality is again by using Lemma A.2.1 on $D_{\tau',\tau'}$.

Applying Lemma 3.4.3 on $Q^{-1}$ and $\left\|((J^\ell)_{\tau',\tau})\right\|$, we have for any $u, q \in N_v^{r-1}$, and $\ell \geq 1$,

$$\left\|\left(Q^{-1}(-J)^\ell_{\tau',\tau}\right)_{u,q}\right\| \leq a^2\frac{2}{\mu}\binom{\ell}{(\ell-|\tau'-\tau|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^\ell(\lambda')^{\text{dist}(u,q)},$$

where $\lambda' \coloneqq \gamma'_S + b_2 \cdot \gamma_S < 1$ and $a \coloneqq \sum_{\gamma\geq 0}(\frac{1+b_1}{1+b_1+b_2})^\gamma h(\gamma)$. Note that $J^0 \coloneqq I$, it is straightforward to verify that the above inequality holds when $\ell = 0$.

With the exponential decay properties of $Q^{-1}(-J)^\ell_{\tau',\tau}$, we have

$$\begin{aligned}
\left\|\left(Q^{-1}\sum_{\ell\geq 0}(-J)^\ell_{\tau',\tau}Ky\right)_{v_0}\right\| &\leq \ell_S a^2\frac{2}{\mu}\sum_{\ell\geq 0}\binom{\ell}{(\ell-|\tau'-\tau|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^\ell \sum_{q\in N_v^{r-1}}(\lambda')^{\text{dist}(v_0,q)}\sum_{u\in\partial N_v^r\cup N_q^1}\|y_u\| \\
&\leq \ell_S a^2\frac{2}{\mu}\sum_{\ell\geq|\tau'-\tau|}\binom{\ell}{(\ell-|\tau'-\tau|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^\ell \sum_{u\in\partial N_v^r}\Delta(\lambda')^{\text{dist}(v_0,u)-1}\|y_u\| \\
&\leq \Delta\ell_S a^2\frac{2}{\mu}\sum_{\ell\geq|\tau'-\tau|}(\frac{4\widetilde{a}\ell_T}{\mu})^\ell \sum_{u\in\partial N_v^r}(\lambda')^{\text{dist}(v_0,u)-1}\|y_u\| \\
&\leq \frac{2\Delta\ell_S a^2}{\mu-4\widetilde{a}\ell_T}(\frac{4\widetilde{a}\ell_T}{\mu})^{|\tau'-\tau|}\sum_{u\in\partial N_v^r}(\lambda')^{\text{dist}(v_0,u)-1}\|y_u\| \\
&= \frac{2\Delta\ell_S a^2}{\lambda'(\mu-4\widetilde{a}\ell_T)}(\frac{4\widetilde{a}\ell_T}{\mu})^{|\tau'-\tau|}\sum_{u\in\partial N_v^r}(\lambda')^{\text{dist}(v_0,u)}\|y_u\|.
\end{aligned} \tag{A.13}$$

The third inequality uses $\binom{\ell}{(\ell-|\tau'-\tau|)/2} \leq 2^\ell$, which can be proved using the following version of Stirling's approximation: For all $n \geq 1$, $e$ denotes the natural number,

$$\sqrt{2\pi n}(n/e)^n e^{1/(12n+1)} < n! < \sqrt{2\pi n}(n/e)^n e^{1/(12n)}.$$

Similarly, consider $\left\|((M^{-1})_{\tau',i})R_i^{(i)}e)_{v_0}\right\|$ for $i \in \{1, k-1\}$. With slight abuse of notation, in this proof, we use $R$ to denote $R_i^{(i)}$ and use the notation $Q^{-1}$ to denote $D_{\tau',\tau'}^{-1}$. Following the same steps as before, we have

$$\left\|\left((M^{-1})_{\tau',i})R_i^{(i)}e\right)_{v_0}\right\| \leq \sum_{\ell\geq 0}\sum_{q\in N_v^r}\left\|\left(Q^{-1}(-J)_{\tau',i}^{\ell}\right)_{v_0,q}\right\| \|(Re)_q\|. \tag{A.14}$$

Since temporal interactions occurs for the same node under consecutive time steps, $R$ is a diagonal block matrix. Hence,

$$\|(Re)_q\| = \|R_{q,q}e_q\| \leq \ell_T \|e_q\|.$$

Moreover, using the exponential decay properties of $Q^{-1}(-J)_{\tau',i}^{\ell}$, we have for $u, q \in N_v^{r-1}$,

$$\left\|\left(Q^{-1}(-J)_{\tau',i}^{\ell}\right)_{u,q}\right\| \leq a^2\frac{2}{\mu}\binom{\ell}{(\ell-|\tau'-i|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^{\ell}(\lambda')^{\text{dist}(u,q)}.$$

Therefore,

$$\begin{aligned}
\left\|((M^{-1})_{\tau',i})R_i^{(i)}e)_{v_0}\right\| &\leq \sum_{\ell\geq 0}\sum_{q\in N_v^r} a^2\frac{2}{\mu}\binom{\ell}{(\ell-|\tau'-i|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^{\ell}(\lambda')^{\text{dist}(v_0,q)}\ell_T\|e_q\| \\
&\leq \sum_{\ell\geq|\tau'-i|}\sum_{q\in N_v^r} a^2\frac{2}{\mu}\binom{\ell}{(\ell-|\tau'-i|)/2}(\widetilde{a}\frac{2\ell_T}{\mu})^{\ell}(\lambda')^{\text{dist}(v_0,q)}\ell_T\|e_q\| \\
&\leq \frac{2\ell_T a^2}{\mu}\sum_{\ell\geq|\tau'-i|}(\frac{4\widetilde{a}\ell_T}{\mu})^{\ell}\sum_{q\in N_v^r}(\lambda')^{\text{dist}(v_0,q)}\|e_q\| \\
&\leq \frac{2\ell_T a^2}{\mu-4\widetilde{a}\ell_T}(\frac{4\widetilde{a}\ell_T}{\mu})^{|\tau'-i|}\sum_{q\in N_v^r}(\lambda')^{\text{dist}(v_0,q)}\|e_q\| \\
&= \frac{a^2\mu}{2\widetilde{a}(\mu-4\widetilde{a}\ell_T)}(\frac{4\widetilde{a}\ell_T}{\mu})^{|\tau'-i|+1}\sum_{q\in N_v^r}(\lambda')^{\text{dist}(v_0,q)}\|e_q\|.
\end{aligned} \tag{A.15}$$

Q.E.D.

Given time index $1 \leq \tau' \leq k-1$, node $v_0 \in N_v^{r-1}$, and perturbation vector $e = (e_0, e_1, \cdots, e_k)$,

$$\begin{aligned}\left\|(\frac{d}{d\theta}\psi(\zeta+\theta e))_{\tau',v_0}\right\| &\leq \left\|\left(M_{\tau',1}^{-1}R_1^{(1)}e_0\right)_{v_0}\right\| + \left\|\left(M_{\tau',k-1}^{-1}R_{k-1}^{(k-1)}e_k\right)_{v_0}\right\| + \sum_{\tau=1}^{k-1}\left\|\left(M_{\tau',\tau}^{-1}K_\tau^{(\tau)}e_\tau\right)_{v_0}\right\| \\ &\leq \frac{a^2\mu}{2\widetilde{a}(\mu-4\widetilde{a}\ell_T)}\left[\rho_T^{\tau'}\sum_{q\in N_v^r}\rho_S^{\text{dist}(v_0,q)}\left\|(e_0)_q\right\| + \rho_T^{k-\tau'}\sum_{q\in N_v^r}\rho_S^{\text{dist}(v_0,q)}\left\|(e_k)_q\right\|\right] \\ &+ \sum_{\tau=1}^{k-1}\frac{2\Delta\ell_S a^2}{\lambda'(\mu-4\widetilde{a}\ell_T)}\rho_T^{|\tau'-\tau|}\sum_{u\in\partial N_v^r}(\rho_S)^{\text{dist}(v_0,u)}\left\|(e_\tau)_u\right\|\end{aligned}$$

where $\rho_T = \frac{4\widetilde{a}\ell_T}{\mu}$ and $\rho_S = \lambda' = (1+b_1+b_2)\gamma_S$. We let $C = \max\{\frac{a^2}{2\widetilde{a}(1-4\widetilde{a}\ell_T/\mu)}, \frac{2a^2\Delta\ell_S/\mu}{\gamma_S(1+b_1+b_2)(1-4\widetilde{a}\ell_T/\mu)}\}$. Under the condition $\mu \geq \max\{8\widetilde{a}\ell_T, \Delta\ell_S(b_1+b_2)/4\}$, $\rho_T < 1$ and $\rho_S < 1$.

Then,

$$\begin{aligned}&\left\|(\frac{d}{d\theta}\psi(\zeta+\theta e))_{\tau',v_0}\right\| \\ &\leq C\left[\rho_T^{\tau'}\sum_{q\in N_v^r}\rho_S^{\text{dist}(v_0,q)}\left\|(e_0)_q\right\| + \rho_T^{k-\tau'}\sum_{q\in N_v^r}\rho_S^{\text{dist}(v_0,q)}\left\|(e_k)_q\right\| + \sum_{\tau=1}^{k-1}\rho_T^{|\tau'-\tau|}\sum_{u\in\partial N_v^r}(\rho_S)^{\text{dist}(v_0,u)}\left\|(e_\tau)_u\right\|\right].\end{aligned}$$

Finally, let $\zeta = \{y_{t-1}^u, z_\tau^u | (\tau,u) \in \partial N_{(v,t)}^{(k,r)}\}$ and $e = \{(y_{t-1}^u)' - y_{t-1}^u, (z_\tau^u)' - z_\tau^u\}$. By integration,

$$\left\|\psi_{(t,v)}^{(k,r)}\left(\{y_{t-1}^u\},\{z_\tau^u\}\right)_{(t_0,v_0)} - (\psi_{(t,v)}^{(k,r)}\left(\{(y_{t-1}^u)'\},\{(z_\tau^u)'\}\right)_{(t_0,v_0)}\right\| \leq \int_0^1\left\|(\frac{d}{d\theta}\psi(\zeta+\theta e))_{t_0,v_0}\right\|d\theta,$$

which is bounded by

$$C\sum_{u\in N_v^r}\rho_T^{t_0-(t-1)}\rho_S^{\text{dist}(v_0,u)}\left\|y_{t-1}^u - (y_{t-1}^u)'\right\| + C\sum_{(u,\tau)\in\partial N_{(t,v)}^{(k,r)}}\rho_T^{|t_0-\tau|}\rho_S^{\text{dist}(v_0,u)}\left\|z_\tau^u - (z_\tau^u)'\right\|.$$

### A.2.3 Adding Constraints to Perturbation Bounds

Recall that in Appendix A.2.1 and A.2.2, we showed Theorem 3.3.1 and Theorem 3.3.2 under the assumption that the individual decisions are unconstrained to simplify the analysis. In this section, we present a general way to relax this assumption by incorporating logarithm barrier

functions, which also applies for Theorem 3.4.1.

Recall that in Assumption 3.2.1, we assume that $D_t^v$ is convex with a non-empty interior, and can be expressed as

$$D_t^v := \{x_t^v \in \mathbb{R}^n \mid (g_t^v)_i(x_t^v) \leq 0, \forall 1 \leq i \leq m_t^v\},$$

where the $i$ th constraint $(g_t^v)_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function in $C^2$. For any time-vertex pair $(\tau, v)$, we can approximate the individual constraints

$$(g_\tau^v)_i(x_\tau^v) \leq 0, \forall 1 \leq i \leq m_\tau^v,$$

by adding the *logarithmic barrier function* $-\mu \sum_{i=1}^{m_\tau^v} \ln\left(-(g_\tau^v)_i(x_\tau^v)\right)$ to the original node cost function $f_\tau^v$. Here, parameter $\mu$ is a positive real number that controls how "good" the barrier function approximates the indicator function

$$\mathbf{I}_{D_\tau^v}(x_\tau^v) = \begin{cases} 0 & \text{if } (g_\tau^v)_i(x_\tau^v) \leq 0, \forall 1 \leq i \leq m_\tau^v, \\ +\infty & \text{otherwise.} \end{cases}$$

The approximation improves as parameter $\mu$ becomes closer to $0$. Thus, the new node cost function will be

$$B_\tau^v(x_\tau^v; \mu) := f_\tau^v(x_\tau^v) - \mu \sum_{i=1}^{m_\tau^v} \ln\left(-(g_\tau^v)_i(x_\tau^v)\right).$$

As an extension of the original notation, we use $\psi_{(t,v)}^{(k,r)}(\{y_{t-1}^u\}, \{z_\tau^u\}; \mu)$ denote the optimal solution

of the following optimization problem

$$\begin{aligned}\underset{\{x_\tau^u|(\tau,v)\in N_{(t,v)}^{(k-1,r-1)}\}}{\arg\min} & \sum_{\tau=t}^{t+k-1}\left(\sum_{u\in N_v^r} B_\tau^u(x_\tau^u;\mu)+\sum_{u\in N_v^r} c_\tau^u(x_\tau^u,x_{\tau-1}^u)+\sum_{(u,q)\in\mathcal{E}(N_v^r)} g_t^{(u,q)}(x_t^u,x_t^q)\right)\\ \text{s.t. } & x_{t-1}^u=y_{t-1}^u,\forall u\in N_v^r,\\ & x_\tau^u=z_\tau^u,\forall(\tau,u)\in\partial N_{(t,v)}^{(k,r)}.\end{aligned}$$

Compared with $\psi_{(t,v)}^{(k,r)}(\{y_{t-1}^u\},\{z_\tau^u\})$ defined in Section 3.3.1, the constraints $x_\tau^u\in D_\tau^u$ are removed and the node costs $f_\tau^u(x_\tau^u)$ are replaced with $B_\tau^u(x_\tau;\mu)$.

A key observation we need to point out is that the perturbation bounds we have shown in Appendix A.2.1 and A.2.2 do not depend on the smoothness constant $\ell_f$ of node cost functions. That means the perturbation bound

$$\begin{aligned}&\left\|\psi_{(t,v)}^{(k,r)}\left(\{y_{t-1}^u\},\{z_\tau^u\};\mu\right)_{(t_0,v_0)}-\psi_{(t,v)}^{(k,r)}\left(\{(y_{t-1}^u)'\},\{(z_\tau^u)'\};\mu\right)_{(t_0,v_0)}\right\|\\ \le\ & C_1\sum_{(u,\tau)\in\partial N_{(t,v)}^{(k,r)}}\rho_T^{|t_0-\tau|}\rho_S^{\text{dist}(v_0,u)}\left\|z_\tau^u-(z_\tau^u)'\right\|+C_2\sum_{u\in N_v^r}\rho_T^{t_0-(t-1)}\rho_S^{\text{dist}(v_0,u)}\left\|y_{t-1}^u-(y_{t-1}^u)'\right\|\end{aligned}$$

holds for arbitrary $\mu$, where $C_1, C_2, \rho_S, \rho_T$ are specified in Theorem 3.3.1 or Theorem 3.3.2 and are independent of $\mu$. Theorem 3.10 in Forsgren, Gill, and Wright, 2002 guarantees that $\psi_{(t,v)}^{(k,r)}(\{y_{t-1}^u\},\{z_\tau^u\};\mu_k)$ converge to $\psi_{(t,v)}^{(k,r)}(\{y_{t-1}^u\},\{z_\tau^u\})$ for any positive sequence $\{\mu_k\}_{k=1}^\infty$ that tends to zero. Thus the above perturbation bound also holds for $\psi_{(t,v)}^{(k,r)}(\{y_{t-1}^u\},\{z_\tau^u\})$ which includes the constraints on individual decisions.

Note that the argument we present in this section also works for Theorem 3.4.1.

## A.3 Competitive Bounds

This appendix includes the proofs of the competitive bounds presented in Section 3.3.3.

### A.3.1 Proof of Theorem 3.4.2

We first derive an upper bound on the distance between $x_t$ and $x_t^*$.

Note that for any time step $t$, we have

$$\left\| x_t - \widetilde{\psi}_t \left( x_{t-1} \right)_t \right\| \leq e_t. \tag{A.16}$$

Thus we see that

$$\begin{aligned}
\|x_t - x_t^*\| &= \left\| x_t - \widetilde{\psi}_1 (x_0)_t \right\| \\
&\leq \left\| x_t - \widetilde{\psi}_t (x_{t-1})_t \right\| + \sum_{i=1}^{t-1} \left\| \widetilde{\psi}_{t-i+1} (x_{t-i})_t - \widetilde{\psi}_{t-i} (x_{t-i-1})_t \right\| \\
&\leq \left\| x_t - \widetilde{\psi}_t (x_{t-1})_t \right\| + \sum_{i=1}^{t-1} C_G \rho_G^i \left\| x_{t-i} - \widetilde{\psi}_{t-i} (x_{t-i-1})_{t-i} \right\| && \text{(A.17a)} \\
&\leq \sum_{i=0}^{t-1} C_0 \rho_G^i \left\| x_{t-i} - \widetilde{\psi}_{t-i} (x_{t-i-1})_{t-i} \right\| && \text{(A.17b)} \\
&\leq \sum_{i=1}^{t} C_0 \rho_G^{t-i} e_i, && \text{(A.17c)}
\end{aligned}$$

where in (A.17a), we used Theorem 3.4.1 and the fact that $\widetilde{\psi}_{t-i}(x_{t-i-1})_t$ can be written as

$$\widetilde{\psi}_{t-i}(x_{t-i-1})_t = \widetilde{\psi}_{t-i+1} \left( \widetilde{\psi}_{t-i}(x_{t-i-1})_{t-i} \right)_t.$$

We also used $C_0 = \max\{1, C_G\}$ in (A.17b) and (A.16) in (A.17c).

By (A.17) and the Cauchy-Schwarz Inequality, we see that

$$\|x_t - x_t^*\|^2 \leq C_0^2 \left( \sum_{i=1}^{t} \rho_G^{t-i} e_i \right)^2 \leq C_0^2 \left( \sum_{i=1}^{t} \rho_G^{t-i} \right) \cdot \left( \sum_{i=1}^{t} \rho_G^{t-i} e_i^2 \right) \leq \frac{C_0^2}{1 - \rho_G} \cdot \left( \sum_{i=1}^{t} \rho_G^{t-i} e_i^2 \right).$$

Summing up over $t$ gives that

$$\sum_{t=1}^{H} \|x_t - x_t^*\|^2 \leq \frac{C_0^2}{(1-\rho_G)^2} \cdot \sum_{t=1}^{H} e_t^2.$$

### A.3.2 Proof of Lemma 3.4.5

In this section, we show Lemma 3.4.5 holds with following specific constants:

$$\begin{aligned} e_t^2 :=& \left\|x_t - x_{t|t-1}^*\right\|^2 \\ &\leq 4C_1^2C_0^2 \left( \frac{h(r)^2\rho_G^2}{(1-\rho_T)(1-\rho_G^2\rho_T)} \cdot \rho_S^{2r} + C_3(r)^2 \cdot \rho_T^{2(k-1)} \cdot \rho_G^{2k} \right) \left\|x_{t-1} - x_{t-1}^*\right\|^2 \\ &\quad + \frac{8C_1^2}{\mu} \left( \frac{h(r)^2}{1-\rho_T} \cdot \rho_S^{2r} \sum_{\tau=t}^{t+k-1} \rho_T^{\tau-t} f_\tau(x_\tau^*) + C_3(r)^2 \cdot \rho_T^{2(k-1)} f_{t+k-1}(x_{t+k-1}^*) \right) \end{aligned} \tag{A.18}$$

Note that, by the principle of optimality, we have

$$\begin{aligned} x_t^v &= \psi_{(t,v)}^{(k,r)} \left( \{x_{t-1}^u\}, \{\theta_\tau^u\} \right)_{(t,v)}, \\ (x_{t|t-1}^v)^* &= \psi_{(t,v)}^{(k,r)} \left( \{x_{t-1}^u\}, \{(x_{\tau|t-1}^u)^*\} \right)_{(t,v)}. \end{aligned}$$

Recall that we define the quantity $C_3(r) := \sum_{\gamma=0}^{r} h(\gamma) \cdot \rho_S^\gamma$ to simplify the notation.

Since the exponentially decaying local perturbation bound holds in Definition 3.3.1, we see that

$$\begin{aligned} \left\|x_t^v - (x_{t|t-1}^v)^*\right\| \leq C_1\rho_S^r \sum_{\tau=t}^{t+k-1} \rho_T^{\tau-t} \sum_{u \in \partial N_v^r} \left\|(x_{\tau|t-1}^u)^* - \theta_\tau^u\right\| \\ + C_1\rho_T^{k-1} \sum_{u \in N_v^r} \rho_S^{d_{\mathcal{G}}(u,v)} \left\|(x_{t+k-1|t-1}^u)^* - \theta_{t+k-1}^u\right\|, \end{aligned} \tag{A.19}$$

which implies that

$$\begin{aligned}
\left\|x_t^v - (x_{t|t-1}^v)^*\right\|^2 &\le 2C_1^2\rho_S^{2r}\left(\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\sum_{u\in\partial N_v^r}\left\|(x_{\tau|t-1}^u)^* - \theta_\tau^u\right\|\right)^2 \\
&\quad + 2C_1^2\rho_T^{2(k-1)}\left(\sum_{u\in N_v^r}\rho_S^{d_{\mathcal{G}}(u,v)}\left\|(x_{t+k-1|t-1}^u)^* - \theta_{t+k-1}^u\right\|\right)^2 \qquad \text{(A.20a)} \\
&\le 2C_1^2\rho_S^{2r}\left(\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\sum_{u\in\partial N_v^r}1\right)\left(\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\sum_{u\in\partial N_v^r}\left\|(x_{\tau|t-1}^u)^* - \theta_\tau^u\right\|^2\right) \\
&\quad + 2C_1^2\rho_T^{2(k-1)}\left(\sum_{u\in N_v^r}\rho_S^{d_{\mathcal{G}}(u,v)}\right)\left(\sum_{u\in N_v^r}\rho_S^{d_{\mathcal{G}}(u,v)}\left\|(x_{t+k-1|t-1}^u)^* - \theta_{t+k-1}^u\right\|^2\right) \qquad \text{(A.20b)} \\
&\le \frac{2C_1^2h(r)}{1-\rho_T}\cdot\rho_S^{2r}\left(\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\sum_{u\in\partial N_v^r}\left\|(x_{\tau|t-1}^u)^* - \theta_\tau^u\right\|^2\right) \\
&\quad + 2C_1^2C_3(r)\cdot\rho_T^{2(k-1)}\left(\sum_{u\in N_v^r}\rho_S^{d_{\mathcal{G}}(u,v)}\left\|(x_{t+k-1|t-1}^u)^* - \theta_{t+k-1}^u\right\|^2\right), \qquad \text{(A.20c)}
\end{aligned}$$

where we used the AM-GM Inequality in (A.20a); we used the Cauchy-Schwarz Inequality in (A.20b); we used the definitions of functions $h(r)$ and $C_3(r)$ in (A.20c).

Summing up (A.20) over all $v \in \mathcal{V}$ and reorganizing terms gives

$$\begin{aligned}
&\sum_{v\in\mathcal{V}}\left\|x_t^v - (x_{t|t-1}^v)^*\right\|^2 \\
&\le \frac{2C_1^2h(r)}{1-\rho_T}\cdot\rho_S^{2r}\sum_{v\in\mathcal{V}}\left(\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\sum_{u\in\partial N_v^r}\left\|(x_{\tau|t-1}^u)^* - \theta_\tau^u\right\|^2\right) \\
&\quad + 2C_1^2C_3(r)\cdot\rho_T^{2(k-1)}\sum_{v\in\mathcal{V}}\left(\sum_{u\in N_v^r}\rho_S^{d_{\mathcal{G}}(u,v)}\left\|(x_{t+k-1|t-1}^u)^* - \theta_{t+k-1}^u\right\|^2\right) \\
&\le \frac{2C_1^2h(r)^2}{1-\rho_T}\cdot\rho_S^{2r}\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\left\|x_{\tau|t-1}^* - \theta_\tau\right\|^2 + 2C_1^2C_3(r)^2\cdot\rho_T^{2(k-1)}\left\|x_{t+k-1|t-1}^* - \theta_{t+k-1}\right\|^2,
\end{aligned} \qquad \text{(A.21)}$$

where we used the facts that

$$\sum_{v\in\mathcal{V}}\sum_{u\in\partial N_v^r}\left\|(x_{\tau|t-1}^u)^*-\theta_\tau^u\right\|^2\le h(r)\sum_{v\in\mathcal{V}}\left\|(x_{\tau|t-1}^v)^*-\theta_\tau^v\right\|^2=h(r)\cdot\left\|x_{\tau|t-1}^*-\theta_\tau\right\|^2,$$

and

$$\begin{aligned}\sum_{v\in\mathcal{V}}\sum_{u\in\partial N_v^r}\rho_S^{d_{\mathcal{G}}(u,v)}\left\|(x_{t+k-1|t-1}^u)^*-\theta_{t+k-1}^u\right\|^2&\le C_3(r)\sum_{v\in\mathcal{V}}\left\|(x_{t+k-1|t-1}^v)^*-\theta_{t+k-1}^v\right\|^2\\&=C_3(r)\cdot\left\|x_{t+k-1|t-1}^*-\theta_{t+k-1}\right\|^2.\end{aligned}$$

We also note that by the principle of optimality, the following equations hold for all $\tau\ge t$:

$$\begin{aligned}x_{\tau|t-1}^*&=\widetilde{\psi}_t\left(x_{t-1}\right)_\tau,\\x_\tau^*&=\widetilde{\psi}_t\left(x_{t-1}^*\right)_\tau.\end{aligned}$$

Recall that $C_0:=\max\{1,C_G\}$. By Theorem 3.4.1, we see that

$$\left\|x_{\tau|t-1}^*-x_\tau^*\right\|\le C_0\rho_G^{\tau-t+1}\left\|x_{t-1}-x_{t-1}^*\right\|,\tag{A.22}$$

which implies

$$\begin{aligned}\left\|x_{\tau|t-1}^*-\theta_\tau\right\|^2&\le2\left\|x_{\tau|t-1}^*-x_\tau^*\right\|^2+2\left\|x_\tau^*-\theta_\tau\right\|^2&\text{(A.23a)}\\&\le2C_0^2\rho_G^{2(\tau-t+1)}\left\|x_{t-1}-x_{t-1}^*\right\|^2+2\left\|x_\tau^*-\theta_\tau\right\|^2,&\text{(A.23b)}\end{aligned}$$

where we used the triangle inequality and the AM-GM inequality in (A.23a); we used (A.22) in (A.23b).

Substituting (A.23) into (A.21) gives

$$\begin{aligned}
&\sum_{v\in\mathcal{V}} \left\|x_t^v - (x_{t|t-1}^v)^*\right\|^2 \\
&\le 4C_1^2C_0^2\left(\frac{h(r)^2\rho_G^2}{(1-\rho_T)(1-\rho_G^2\rho_T)}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\cdot\rho_G^{2k}\right)\left\|x_{t-1}-x_{t-1}^*\right\|^2 \\
&\quad + 4C_1^2\left(\frac{h(r)^2}{1-\rho_T}\cdot\rho_S^{2r}\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}\left\|x_\tau^*-\theta_\tau\right\|^2 + C_3(r)^2\cdot\rho_T^{2(k-1)}\left\|x_{t+k-1}^*-\theta_{t+k-1}\right\|^2\right) \\
&\le 4C_1^2C_0^2\left(\frac{h(r)^2\rho_G^2}{(1-\rho_T)(1-\rho_G^2\rho_T)}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\cdot\rho_G^{2k}\right)\left\|x_{t-1}-x_{t-1}^*\right\|^2 \\
&\quad + \frac{8C_1^2}{\mu}\left(\frac{h(r)^2}{1-\rho_T}\cdot\rho_S^{2r}\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}f_\tau(x_\tau^*) + C_3(r)^2\cdot\rho_T^{2(k-1)}f_{t+k-1}(x_{t+k-1}^*)\right),
\end{aligned} \tag{A.24}$$

where we used the fact that the node cost function $f_\tau^v$ is non-negative and $\mu$-strongly convex for all $\tau, v$, thus

$$f_\tau(x_\tau^*) \ge \sum_{v\in\mathcal{V}} f_\tau^v((x_\tau^v)^*) \ge \frac{\mu}{2}\sum_{v\in\mathcal{V}}\|(x_\tau^v)^*-\theta_\tau^v\|^2 = \frac{\mu}{2}\left\|x_\tau^*-\theta_\tau\right\|^2.$$

Note that $\sum_{v\in\mathcal{V}}\left\|x_t^v-(x_{t|t-1}^v)^*\right\|^2 = \left\|x_t-x_{t|t-1}^*\right\|^2 = e_t^2$. Thus we have finished the proof of (A.18).

### A.3.3 Proof of Theorem 3.3.3

In this section, we show Theorem 3.3.3 holds with the following specific constants:

$$1+\left(1+\frac{32C_0^2C_1^2(\ell_f+\Delta\ell_S+2\ell_T)\cdot h(r)^2}{\mu(1-\rho_G)^2(1-\rho_T)^2}\right)\cdot\rho_S^r + \left(1+\frac{32C_0^2C_1^2(\ell_f+\Delta\ell_S+2\ell_T)C_3(r)^2}{\mu(1-\rho_G)^2}\right)\rho_T^{k-1}. \tag{A.25}$$

under the assumption that

$$\frac{4C_1^2C_0^4}{(1-\rho_G)^2}\left(\frac{h(r)^2\rho_G^2}{(1-\rho_T)(1-\rho_G^2\rho_T)}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\cdot\rho_G^{2k}\right) \le \frac{1}{2}. \tag{A.26}$$

Recall that $C_0$ is defined in Theorem 3.4.2. Note that Theorem 3.3.1 and Theorem 3.4.1 hold

under Assumption 3.2.1. One can check that $C_0, C_1, (1-\rho_G)^{-1}$, and $(1-\rho_T)^{-1}$ are bounded by polynomials of $\ell_f/\mu, \ell_T/\mu$, and $(\Delta\ell_S)/\mu$.

In the proof, we need to use Lemma F.2 in Lin et al., 2021 to bound LPC's total cost by a weighted sum of the offline optimal cost and the sum of squared distances between their trajectories. For completeness, we present Lemma F.2 in Lin et al., 2021 below:

**Lemma A.3.1.** *For a fixed dimension $m \in \mathbb{Z}_+$, assume a function $h : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is convex, $\ell$-smooth and continuously differentiable. For all $x, y \in \mathbb{R}^m$, for all $\eta > 0$, we have*

$$h(x) \leq (1+\eta)h(y) + \frac{\ell}{2}\left(1+\frac{1}{\eta}\right)\|x-y\|^2.$$

Now we come back to the proof of Theorem 3.3.3. We first bound the sum of squared distances between LPC's trajectory and the offline optimal trajectory:

$$\sum_{t=1}^{H}\|x_t - x_t^*\|^2 \leq \frac{C_0^2}{(1-\rho_G)^2}\sum_{t=1}^{H} e_t^2 \tag{A.27a}$$

$$\leq \frac{4C_1^2C_0^4}{(1-\rho_G)^2}\left(\frac{h(r)^2\rho_G^2}{(1-\rho_T)(1-\rho_G^2\rho_T)}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\cdot\rho_G^{2k}\right)\sum_{t=1}^{H}\left\|x_{t-1}-x_{t-1}^*\right\|^2$$
$$+ \frac{8C_0^2C_1^2}{\mu(1-\rho_G)^2}\sum_{t=1}^{H}\left(\frac{h(r)^2}{1-\rho_T}\cdot\rho_S^{2r}\sum_{\tau=t}^{t+k-1}\rho_T^{\tau-t}f_\tau(x_\tau^*) + C_3(r)^2\cdot\rho_T^{2(k-1)}f_{t+k-1}(x_{t+k-1}^*)\right), \tag{A.27b}$$

where we used Theorem 3.4.2 in (A.27a); we used Lemma 3.4.5 with the specific constants given in Appendix A.3.2 in (A.27b).

Recall that in (A.26), we assume $r$ and $k$ are sufficient large so that the coefficient of the first term in (A.27) satisfies

$$\frac{4C_1^2C_0^4}{(1-\rho_G)^2}\left(\frac{h(r)^2\rho_G^2}{(1-\rho_T)(1-\rho_G^2\rho_T)}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\cdot\rho_G^{2k}\right) \leq \frac{1}{2}.$$

Substituting this bound into (A.27) gives that

$$\sum_{t=1}^{H} \|x_t - x_t^*\|^2 \leq \frac{16C_0^2C_1^2}{\mu(1-\rho_G)^2}\left(\frac{h(r)^2}{(1-\rho_T)^2}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\right)\cdot\sum_{t=1}^{H} f_t(x_t^*). \tag{A.28}$$

By Lemma A.3.1, since $f_t$ is $(\ell_f + \Delta\ell_S)$-smooth, convex, and non-negative on $\mathbb{R}^n$, and $c_t$ is $\ell_T$-smooth, convex, and non-negative on $\mathbb{R}^n \times \mathbb{R}^n$, we know that

$$\begin{aligned} f_t(x_t) &\leq (1+\eta) f_t(x_t^*) + \frac{\ell_f + \Delta\ell_S}{2}\left(1+\frac{1}{\eta}\right)\|x_t - x_t^*\|^2 \\ c_t(x_t, x_{t-1}) &\leq (1+\eta) c_t(x_t^*, x_{t-1}^*) + \frac{\ell_T}{2}\left(1+\frac{1}{\eta}\right)\left(\|x_t - x_t^*\|^2 + \left\|x_{t-1} - x_{t-1}^*\right\|^2\right) \end{aligned} \tag{A.29}$$

holds for any $\eta > 0$. Summing the above inequality over $t$ gives

$$\begin{aligned} &\sum_{t=1}^{H}\left(f_t(x_t) + c_t(x_t, x_{t-1})\right) \\ &\leq (1+\eta)\sum_{t=1}^{H}\left(f_t(x_t^*) + c_t(x_t^*, x_{t-1}^*)\right) + \frac{(\ell_f + \Delta\ell_S + 2\ell_T)}{2}\left(1+\frac{1}{\eta}\right)\sum_{t=1}^{H}\|x_t - x_t^*\|^2 \\ &\leq (1+\eta)\mathit{cost}(OPT) \qquad\qquad (A.30) \\ &\quad + \left(1+\frac{1}{\eta}\right)\frac{16C_0^2C_1^2(\ell_f + \Delta\ell_S + 2\ell_T)}{\mu(1-\rho_G)^2}\left(\frac{h(r)^2}{(1-\rho_T)^2}\cdot\rho_S^{2r} + C_3(r)^2\cdot\rho_T^{2(k-1)}\right)\cdot\mathit{cost}(OPT), \qquad\qquad (A.31) \end{aligned}$$

where we used (A.28) and $\sum_{t=1}^{H} f_t(x_t^*) \leq \mathit{cost}(OPT)$ in the last inequality. Setting $\eta = \rho_S^r + \rho_T^{k-1}$ in (A.30) finishes the proof of (A.25).

As a remark, we require the local cost function $(f_t^v, c_t^v, s_t^e)$ to be non-negative, convex, and smooth in the whole Euclidean spaces $(\mathbb{R}^n, \mathbb{R}^n \times \mathbb{R}^n, \mathbb{R}^n \times \mathbb{R}^n)$ in Assumption 3.2.1 because we want to apply Lemma A.3.1 in (A.29).

### A.3.4 Proof of Corollary 3.3.4

We first show $\Delta^2\rho_S \leq \sqrt{\rho_S}$ holds under Assumption 3.2.1 and the assumptions that $\frac{\ell_S}{\mu} \leq \frac{1}{\Delta^7}, \frac{\ell_T}{\mu} \leq \frac{1}{16}$. To see this, note that as we discussed in Section 3.3.2, by setting $b_1 = 2\Delta - 1$ and $b_2 = 4\Delta^2 - 2\Delta$, Theorem 3.3.3 holds with

$$\rho_S = \frac{4\Delta^2(\sqrt{1+\Delta\ell_S/\mu} - 1)}{\sqrt{1+\Delta\ell_S/\mu} + 1}.$$

Hence we see that

$$\Delta^2\sqrt{\rho_S} = 2\Delta^3\left(\frac{\sqrt{1+(\Delta\ell_S/\mu)} - 1}{\sqrt{1+(\Delta\ell_S/\mu)} + 1}\right)^{\frac{1}{2}} \leq 2\Delta^3\left(\frac{\sqrt{1+\Delta^{-6}} - 1}{2}\right)^{\frac{1}{2}} \leq 1,$$

which implies that

$$\Delta^2\rho_S \leq \sqrt{\rho_S}. \tag{A.32}$$

Recall that function $C_3(r) := \sum_{\gamma=0}^{r} h(\gamma) \cdot \rho_S^\gamma$. Hence we see that

$$C_3(r) \leq \sum_{\gamma=0}^{r} \Delta^\gamma \cdot \rho_S^\gamma \leq \sum_{\gamma=0}^{r}\left(\frac{\sqrt{\rho_S}}{\Delta}\right)^\gamma \leq \frac{\Delta}{\Delta - \sqrt{\rho_S}}. \tag{A.33}$$

Substituting (A.32) and (A.33) into the competitive ratio bound in (A.25) shows that the competitive ratio of LPC is upper bound by

$$1 + \left(1 + \frac{32C_0^2C_1^2(\ell_f + \Delta\ell_S + 2\ell_c)}{\mu(1-\rho_G)^2(1-\rho_T)^2}\right) \cdot \rho_S^{\frac{r}{2}} + \left(1 + \frac{32C_0^2C_1^2(\ell_f + \Delta\ell_S + 2\ell_c)\Delta^2}{\mu(1-\rho_G)^2(\Delta - \sqrt{\rho_S})^2}\right)\rho_T^{k-1}.$$

## A.4 Proof of Theorem 3.3.5

In this appendix we prove a lower bound on the competitive ratio of any online algorithm. Our proof focuses on temporal and spatial lower bounds separately first, and then combines them.

**Step 1: Temporal Lower Bounds**

We first show that the competitive ratio of any online algorithm with $k$ steps of future predictions is lower bounded by $1+\Omega(\lambda_T^k)$. To show this, we consider the special case when there are no spatial interaction costs (i.e., $s_t^e \equiv 0$ for all $t$ and $e$). In this case, since all agents are independent with each other, it suffices to assume there is only one agent in the network $\mathcal{G}$. Thus we will drop the agent index in the following analysis. To further simplify the problem, we assume dimension $n=1$, $c_t(x_t, x_{t-1}) = \frac{\ell_T}{2}(x_t - x_{t-1})^2$, and the feasible set is $D_t \equiv D = [0,1]$ for all $t$. Let $R$ denote the diameter of $D$, i.e., $R = \sup_{x,y\in D}|x-y| = 1$.

By Theorem 2 in Li, Qu, and Li, 2020 and Case 1 in its proof, we know that for any online algorithm $ALG$ with $k$ steps of future predictions and $L_T \in (2R, RH)$, there exists a problem instance with quadratic functions $f_1, f_2, \ldots, f_H$ that have the form $f_t(x_t) = \frac{\mu}{2}(x_t - \theta_t)^2, \theta_t \in D$ such that

$$cost(ALG) - cost(OPT) \geq \frac{\mu^3(1-\sqrt{\lambda_T})^2}{96(\mu+1)^2} \cdot \lambda_T^k \cdot R \cdot L_H, \tag{A.34}$$

where $L_H \geq \sum_{t=1}^H |\theta_t - \theta_{t-1}|$. Note that

$$\begin{aligned} R \cdot L_T &\geq \sum_{t=1}^H |v_t - v_{t-1}|^2 \\ &= \frac{2}{\ell_T} \cdot \sum_{t=1}^H \left(f_t(v_t) + c_t(v_t, v_{t-1})\right) \\ &\geq \frac{2}{\ell_T} \cdot cost(OPT). \end{aligned}$$

Substituting this into (A.34) gives

$$cost(ALG) \geq \left(1 + \frac{\mu^3(1-\sqrt{\lambda_T})^2}{48(\mu+1)^2\ell_T} \cdot \lambda_T^k\right) \cdot cost(OPT). \tag{A.35}$$

Note that (A.34) implies $cost(ALG) > 0$, hence the competitive ratio can be unbounded if

$cost(OPT) = 0$.

**Step 2: Spatial Lower Bounds**

We next show that the competitive ratio of any online algorithm that can communicate within $r$-hop neighborhood according to the scheme defined in Section 3.2.1 is lower bounded by $1+\Omega(\lambda_S^r)$. To show this, we will construct a special Networked OCO instance with random cost functions and show there exists a realization that achieves the lower bound by probabilistic methods.

**Theorem A.4.1.** *Under the assumption that $\Delta \geq 3$, the competitive ratio of any decentralized online algorithm $ALG$ with communication radius $r$ is lower bounded by $1 + \Omega(\lambda_S^r)$, where $\Omega(\cdot)$ notation hides factors that depend polynomially on $1/\mu, \ell_T, \ell_S$, and $\Delta$, and*

$$\lambda_S = \begin{cases} \frac{(\Delta\ell_S/\mu)}{3+3(\Delta\ell_S/\mu)} & \text{if } \Delta\ell_S/\mu < 48, \\ \max\left(\frac{(\Delta\ell_S/\mu)}{3+3(\Delta\ell_S/\mu)}, \left(1 - 4\sqrt{3}\cdot(\Delta\ell_S/\mu)^{-\frac{1}{2}}\right)^2\right) & \text{otherwise.} \end{cases} \tag{A.36}$$

*Proof of Theorem A.4.1.* In the proof, we assume the online game only lasts one time step before it ends, i.e., $H = 1$. Note that when $H > 1$, the same counterexample can be constructed repeatedly by letting the temporal interaction costs $c_t^v \equiv 0$ for every agent $v$ and time step $t$. To simplify the notation, we define $\ell := \ell_S/\mu$ and $d := [\Delta/2]$. Without the loss of generality, we assume $\mathcal{V} = \{1, 2, \cdots, n\}$ so that each agent has a positive integer index.

We consider the case where the node cost function for each agent $i$ is $(x_i + w_i)^2$ and the spatial interaction cost between two neighboring agents $i$ and $j$ is $\ell(x_i - x_j)^2$. Here, $x_i \in \mathbb{R}$ is the scalar action of agent $i$, and parameter $w_i \in \mathbb{R}$ is a local information that corresponds to agent $i$. The parameters $\{w_i\}_{i=1}^n$ are sampled i.i.d. from some distribution $\mathcal{D}$, which we will discuss later.

For a general graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of agents, let $L$ denote its graph Laplacian matrix. Recall that

the graph Laplacian matrix $L \in \mathcal{V} \times \mathcal{V}$ is a symmetric $n \times n$ matrix and it is defined as

$$L_{i,j} = \begin{cases} deg(i) & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise,} \end{cases}$$

for agents $i, j \in \mathcal{V}$. Here $deg(\cdot)$ denotes the degree of an agent in graph $\mathcal{G}$. We know that $L$ is a symmetric semi-definite positive semi-definite and has bandwidth $1$ w.r.t. to $\mathcal{G}$. The centralized optimization problem can be expressed as

$$\begin{aligned} cost(OPT) &= \min_{x \in \mathbb{R}^n} (x+w)^\top (x+w) + \ell \cdot x^\top L x \\ &= \min_{x \in \mathbb{R}^n} \left\| (I + \ell \cdot L)^{\frac{1}{2}} x + (I + \ell \cdot L)^{-\frac{1}{2}} w \right\|^2 + w^\top (I - (I + \ell \cdot L)^{-1}) w \\ &= w^\top (I - (I + \ell \cdot L)^{-1}) w, \end{aligned}$$

where the minimum is attained at $x^* = (I + \ell \cdot L)^{-1} w$.

When each agent $i$ only has communication radius $r$, it can only observe the part of $w$ that is within $N_i^r$. To simplify the notation, we define the mask operator $\phi_S : \mathbb{R}^n \to \mathbb{R}^n$ w.r.t. a set $S \subseteq \mathcal{V}$ as

$$\phi_S(w)_i = \begin{cases} w_i & \text{if } i \in S, \\ 0 & \text{otherwise,} \end{cases}$$

for $i \in \mathcal{V}$. The local policy of agent $i$ (denote as $\pi_i$) is a mapping from $w_{N_i^r}$ to the local decision $x_i$.

Suppose the distribution $\mathcal{D}$ of each local parameters $w_i$ is a mean-zero distribution with support

on $\mathbb{R}$. For every agent $i \in \mathcal{V}$, we see that

$$\begin{aligned}
\mathbb{E}_w \left| x_i(w) - x_i^*(w) \right|^2 &= \min_{\pi_i} \mathbb{E}_w \left| \pi_i(w_{N_i^r}) - x_i^*(w) \right|^2 \\
&\geq \mathbb{E}_w \left| \mathbb{E}[x_i^*(w) \mid w_{N_i^r}] - x_i^*(w) \right|^2 && \text{(A.37a)} \\
&= \mathbb{E}_w \left| \mathbb{E}[\left((I + \ell \cdot L)^{-1} w\right)_i \mid w_{N_i^r}] - \left((I + \ell \cdot L)^{-1} w\right)_i \right|^2 \\
&= \mathbb{E}_w \left| \left((I + \ell \cdot L)^{-1} \phi_{N_i^r}(w)\right)_i - \left((I + \ell \cdot L)^{-1} w\right)_i \right|^2 && \text{(A.37b)} \\
&= \mathbb{E}_w \left| \left((I + \ell \cdot L)^{-1} \phi_{N_{-i}^r}(w)\right)_i \right|^2, && \text{(A.37c)}
\end{aligned}$$

where we use the fact that conditional expectations minimize the mean square prediction error in (A.37a); we use the requirement that the distribution of $w$ is mean-zero in (A.37b).

To bound the variance term in (A.37c), we need the following lemma to lower bound the magnitude of every entry in the exponential decaying matrix $(I + \ell \cdot L)^{-1}$:

**Lemma A.4.1.** *There exists a finite graph $\mathcal{G}$ with maximum degree $2d$ that satisfies the following conditions: For any two vertices $i, j$ such that $d_{\mathcal{G}}(i, j) \geq 3$, the following inequality holds:*

$$\left((I + \ell \cdot L)^{-1}\right)_{ij} \geq \frac{d_{\mathcal{G}}(i,j)}{d^2(2d\ell + 1)} \cdot \left(\frac{d\ell}{2d\ell + 1}\right)^{d_{\mathcal{G}}(i,j)}.$$

*If we make the additional assumption that $\ell > \frac{16}{d}$, we have that*

$$\left((I + \ell \cdot L)^{-1}\right)_{ij} \geq \frac{1}{4\sqrt{\pi \cdot d_{\mathcal{G}}(i,j) \cdot \sqrt{d\ell}} \cdot d^2(2d\ell + 1)} \cdot \left(1 - 4(d\ell)^{-\frac{1}{2}}\right)^{d_{\mathcal{G}}(i,j)}.$$

We defer the proof of Lemma A.4.1 to the end of this section. Note that Lemma A.4.1 implies that there exists a graph $\mathcal{G}$ that satisfies $\left((I + \ell \cdot L)^{-1}\right)_{i,j} = \Omega\left(\lambda_S^r\right)$, where $\Omega(\cdot)$ notation hides factors that depend polynomially on $1/\mu, \ell_T, \ell_S$, and $\Delta$, and $\lambda_S$ is as defined in (A.36). We assume the agents are located in this graph $\mathcal{G}$ for the rest of the proof.

Using Lemma A.4.1, we can derive the following lower bound of the variance term in (A.37b):

$$\begin{aligned}\mathbb{E}_w \left| \left( (I + \ell \cdot L)^{-1} \phi_{N^r_{-i}}(w) \right)_i \right|^2 &= \mathbb{E}_w \left( \sum_{j \in N^r_{-i}} \left( (I + \ell \cdot L)^{-1} \right)_{ij} w_j \right)^2 \\ &= \sum_{j \in N^r_{-i}} \left( (I + \ell \cdot L)^{-1} \right)^2_{ij} Var(w_j) \\ &\geq \sum_{j \in \partial N^{r+1}_i} \left( (I + \ell \cdot L)^{-1} \right)^2_{ij} Var(w_j) \\ &\geq \Theta\left( \lambda_S^r \cdot Var(w_i) \right). \end{aligned} \tag{A.38}$$

Substituting (A.38) into (A.37) and summing over all vertices $i$, we obtain that

$$\mathbb{E}_w \left\| x(w) - x^*(w) \right\|^2 \geq \sum_{i=1}^n \mathbb{E}_w \left| x_i(w) - x_i^*(w) \right|^2 \geq \Theta\left( n \cdot \lambda_S^r \cdot Var(w_i) \right).$$

We also see that

$$\mathbb{E}_w[cost(OPT)] = \mathbb{E}_w \left[ w^\top (I - (I + \ell \cdot L)^{-1}) w \right] = O(n \cdot Var(w_i)). \tag{A.39}$$

Note that the global objective function $(x+w)^\top (x+w) + \ell \cdot x^\top L x$ is 1-strongly convex, and $x^*(w)$ is minimizer of this function. Thus, we have that for any outcome of $w$,

$$cost(ALG) - cost(OPT) \geq \frac{1}{2} \left\| x(w) - x^*(w) \right\|^2.$$

Taking expectations on both sides w.r.t. $w$ gives that

$$\mathbb{E}_w cost(ALG) - \mathbb{E}_w cost(OPT) \geq \frac{1}{2} \mathbb{E}_w \left\| x(w) - x^*(w) \right\|^2 \geq \Theta\left( n \cdot \lambda_S^r \cdot Var(w_i) \right). \tag{A.40}$$

Dividing (A.40) by (A.39), we obtain that

$$\frac{\mathbb{E}_w cost(ALG)}{\mathbb{E}_w cost(OPT)} \geq 1 + \Omega\left( \lambda_S^r \right).$$

Note that $\mathbb{P}_w\left[cost(OPT) = 0\right] = 0$. Thus, there must exist an instance of $w$ such that $cost(OPT) > 0$ and

$$\frac{cost(ALG)}{cost(OPT)} \geq 1 + \Omega\left(\lambda_S^r\right).$$

Q.E.D.

Before we present the proof of Lemma A.4.1, we first need to introduce two technical lemmas that will be used in the proof of Lemma A.4.1. The first lemma (Lemma A.4.2) provides a lower bound for binomial coefficient $\binom{(2+\epsilon)m}{m}$.

**Lemma A.4.2.** *For any positive integer $m$ and $\epsilon \in \mathbb{R}_{\geq 0}$ such that $\epsilon m$ is an integer, the following inequality holds:*

$$\binom{(2+\epsilon)m}{m} > \frac{1}{\sqrt{2\pi}} m^{-\frac{1}{2}} \cdot \frac{(2+\epsilon)^{(2+\epsilon)m+\frac{1}{2}}}{(1+\epsilon)^{(1+\epsilon)m+\frac{1}{2}}} \cdot e^{-\frac{1}{6m}}.$$

*Proof of Lemma A.4.2.* By Lemma 2.1 in Stanica, 2001, we know for any $n \in \mathbb{Z}_+$,

$$n! = \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n+r(n)},$$

where $r(n)$ satisfies $\frac{1}{12n+1} < r(n) < \frac{1}{12n}$. Thus we see that

$$\sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n+\frac{1}{12n+1}} < n! < \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n+\frac{1}{12n}}, \forall n \in \mathbb{Z}_+.$$

Therefore, we can lower bound $\binom{(2+\epsilon)m}{m}$ by

$$\begin{aligned}
\binom{(2+\epsilon)m}{m} &= \frac{((2+\epsilon)m)!}{m! \cdot ((1+\epsilon)m)!} \\
&> \frac{\sqrt{2\pi}((2+\epsilon)m)^{(2+\epsilon)m+\frac{1}{2}} e^{-(2+\epsilon)m+\frac{1}{12(2+\epsilon)m+1}}}{\sqrt{2\pi} m^{m+\frac{1}{2}} e^{-m+\frac{1}{12m}} \cdot \sqrt{2\pi}((1+\epsilon)m)^{(1+\epsilon)m+\frac{1}{2}} e^{-(1+\epsilon)m+\frac{1}{12(1+\epsilon)m}}} \\
&= \frac{1}{\sqrt{2\pi}} m^{-\frac{1}{2}} \cdot \frac{(2+\epsilon)^{(2+\epsilon)m+\frac{1}{2}}}{(1+\epsilon)^{(1+\epsilon)m+\frac{1}{2}}} \cdot e^{\frac{1}{12(2+\epsilon)m+1} - \frac{1}{12m} - \frac{1}{12(1+\epsilon)m}} \\
&> \frac{1}{\sqrt{2\pi}} m^{-\frac{1}{2}} \cdot \frac{(2+\epsilon)^{(2+\epsilon)m+\frac{1}{2}}}{(1+\epsilon)^{(1+\epsilon)m+\frac{1}{2}}} \cdot e^{-\frac{1}{6m}}.
\end{aligned}$$

Q.E.D.

The second technical lemma (Lemma A.4.3) will be used to simplify the decay factor in the proof of Lemma A.4.1.

**Lemma A.4.3.** *For all $\epsilon \in [0, \sqrt{2})$, the following inequality holds*

$$\frac{2+\epsilon}{2 \cdot (1+\epsilon)^{\frac{1+\epsilon}{2+\epsilon}}} \geq 1 - \frac{\epsilon^2}{2}.$$

*Proof of Lemma A.4.3.* By taking logarithm on both sides, we see the original inequality is equivalent to

$$\ln\left(1+\frac{\epsilon}{2}\right) - \frac{1+\epsilon}{2+\epsilon}\ln(1+\epsilon) \geq \ln\left(1-\frac{1}{2}\epsilon^2\right),$$

which is further equivalent to

$$\ln\left(1+\frac{\epsilon}{2}\right) - \frac{1+\epsilon}{2+\epsilon}\ln(1+\epsilon) - \ln\left(1-\frac{1}{2}\epsilon^2\right) \geq 0. \tag{A.41}$$

Note that the LHS can be lower bounded by

$$\ln\left(1+\frac{\epsilon}{2}\right) - \frac{1+\epsilon}{2+\epsilon}\ln(1+\epsilon) - \ln\left(1-\frac{1}{2}\epsilon^2\right) \geq \ln\left(1+\frac{\epsilon}{2}\right) - \frac{1+\epsilon}{2}\ln(1+\epsilon) - \ln\left(1-\frac{1}{2}\epsilon^2\right) =: g(\epsilon).$$

Function $g$ satisfies that $g(0) = 0$, and its derivative is

$$\begin{aligned} g'(\epsilon) &= \frac{1}{2+\epsilon} - \frac{1}{2} - \frac{1}{2}\ln(1+\epsilon) + \frac{\epsilon}{1-\frac{1}{2}\epsilon^2} \\ &\geq \frac{1}{2+\epsilon} - \frac{1}{2} - \frac{\epsilon}{2} + \epsilon \\ &= \frac{2-(2+\epsilon)(1-\epsilon)}{2(2+\epsilon)} \\ &= \frac{\epsilon+\epsilon^2}{2(2+\epsilon)} \geq 0. \end{aligned}$$

Thus, $g(\epsilon) \geq 0$ for all $\epsilon \in [0, \sqrt{2})$. Hence (A.41) holds for all $\epsilon \in [0, \sqrt{2})$. Q.E.D.

Now we are ready to present the proof of Lemma A.4.1.

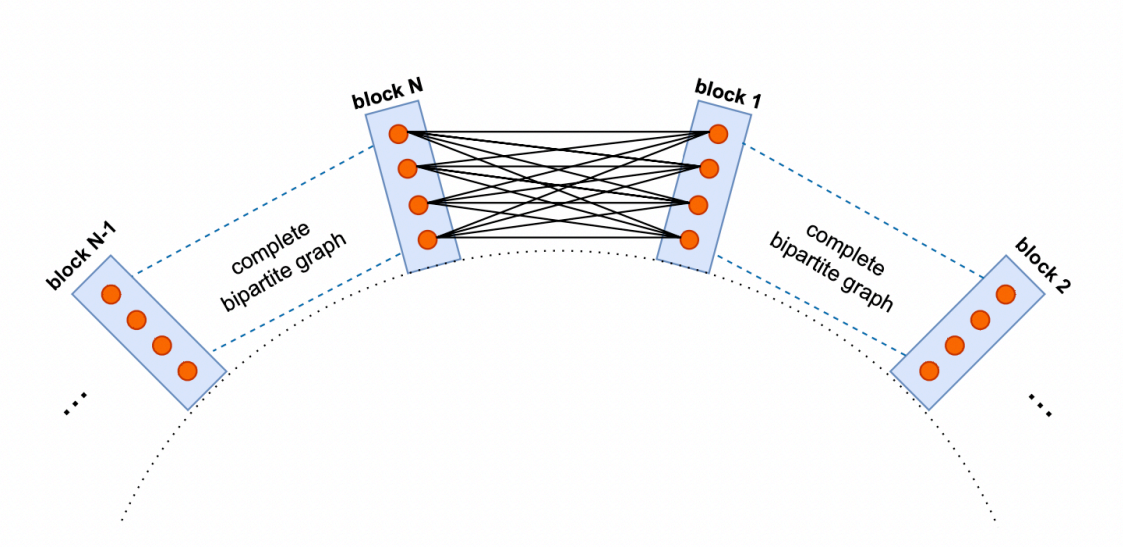


Figure A.1: Graph structure of $\mathcal{G}$ to obtain the lower bound: $N$ blocks form a ring. Each block contains $d$ vertices.

*Proof of Lemma A.4.1.* Consider the graph $\mathcal{G}$ constructed as Figure A.1: Let $N$ be a positive integer that is sufficiently large. $N$ blocks form a ring, where each block contains $d$ nodes. Every pair of blocks are connected by a complete bipartite graph. The graph Laplacian of $\mathcal{G}$ can be decomposed as $L = 2dI - M$, where $M$ is the adjacency matrix of $\mathcal{G}$. We see that

$$\begin{aligned}(I + \ell \cdot L)^{-1} &= ((2d\ell + 1)I - \ell \cdot M)^{-1} \\ &= \frac{1}{2d\ell + 1}\left(I - \frac{\ell}{2d\ell + 1}M\right)^{-1} \\ &= \frac{1}{2d\ell + 1}\sum_{t=0}^{\infty}\frac{\ell^t}{(2d\ell + 1)^t}M^t\end{aligned}$$

Fix two vertices $i$ and $j$ and denote $\kappa := d_{\mathcal{G}}(i,j)$ and assume $\kappa \geq 3$. Without the loss of generality, we can assume $j$ is on the clockwise direction of $i$. We see that

$$\begin{aligned}\left((I+\ell\cdot L)^{-1}\right)_{ij} &= \frac{1}{2d\ell+1}\sum_{t=0}^{\infty}\frac{\ell^t}{(2d\ell+1)^t}(M^t)_{ij} \\ &= \frac{\ell^\kappa}{(2d\ell+1)^{\kappa+1}}\sum_{m=0}^{\infty}\frac{\ell^{2m}}{(2d\ell+1)^{2m}}(M^{\kappa+2m})_{ij}. \end{aligned} \tag{A.42}$$

Note that $(M^{\kappa+2m})_{ij}$ denotes the number of paths from $i$ to $j$ with length $\kappa+2m$ in graph $\mathcal{G}$. Note that the shortest paths from $i$ to $j$ have length $\kappa$. To pick a path with length $(\kappa+2m)$ from $i$ to $j$, we can first pick a path on the level of blocks: The number of possible block-level paths is lower bounded by $\binom{\kappa+2m}{m}$ because we can choose $m$ in $(\kappa+2m)$ steps to go in the counter clockwise direction. After a block-level path is fixed, we can choose which specific vertices in the blocks we want to land at, and there are $d^{\kappa+2m-2}$ choices. Thus we see that

$$(M^{\kappa+2m})_{ij} \geq \binom{\kappa+2m}{m} d^{\kappa+2m-2}.$$

Substituting this into (A.42) gives

$$\left((I+\ell\cdot L)^{-1}\right)_{ij} \geq \frac{\ell^\kappa d^{\kappa-2}}{(2d\ell+1)^{\kappa+1}}\sum_{m=0}^{\infty}\frac{\ell^{2m}d^{2m}}{(2d\ell+1)^{2m}}\binom{\kappa+2m}{m}. \tag{A.43}$$

Let $m=0$ will give that $((I+\ell\cdot L)^{-1})_{ij} \geq \frac{\kappa}{d^2(2d\ell+1)}\cdot\left(\frac{d\ell}{2d\ell+1}\right)^\kappa$, which shows the first claim of Lemma A.4.1. Now we proceed to show the second claim of Lemma A.4.1.

By Lemma A.4.2, we know that when $\kappa=\epsilon m$, we have that

$$\begin{aligned}\binom{\kappa+2m}{m} &= \binom{(2+\epsilon)m}{m} \\ &> \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{6m}}m^{-\frac{1}{2}}\frac{(2+\epsilon)^{(2+\epsilon)m}}{(1+\epsilon)^{(1+\epsilon)m}}\cdot\left(\frac{2+\epsilon}{1+\epsilon}\right)^{\frac{1}{2}} \\ &\geq \frac{1}{2\sqrt{2\pi}}m^{-\frac{1}{2}}\left(\frac{2+\epsilon}{(1+\epsilon)^{\frac{1+\epsilon}{2+\epsilon}}}\right)^{(2+\epsilon)m}.\end{aligned}$$

For any $m > \kappa$, the inequality we just showed can help us bound a term in the summation of (A.43) below:

$$\begin{aligned}
&\frac{\ell^{\kappa} d^{\kappa-2}}{(2d\ell+1)^{\kappa+1}} \cdot \frac{\ell^{2m} d^{2m}}{(2d\ell+1)^{2m}} \binom{\kappa+2m}{m} \\
&\geq \frac{1}{d^2(2d\ell+1)} \cdot \frac{1}{2^{\kappa+2m}} \binom{\kappa+2m}{m} \cdot \left(1 - \frac{1}{2d\ell+1}\right)^{\kappa+2m} \\
&\geq \frac{1}{2\sqrt{2\pi} \cdot \sqrt{\frac{\kappa}{\epsilon}} \cdot d^2(2d\ell+1)} \cdot \left(\frac{2+\epsilon}{2 \cdot (1+\epsilon)^{\frac{1+\epsilon}{2+\epsilon}}}\right)^{(2+\epsilon)\kappa/\epsilon} \cdot \left(1 - \frac{1}{2d\ell+1}\right)^{(1+\frac{2}{\epsilon})\kappa} \\
&\geq \frac{1}{2\sqrt{2\pi} \cdot \sqrt{\frac{\kappa}{\epsilon}} \cdot d^2(2d\ell+1)} \cdot \left(\left(1 - \frac{\epsilon^2}{2}\right)^{\frac{1}{\epsilon}} \cdot \left(1 - \frac{1}{2d\ell+1}\right)^{\frac{1}{\epsilon}}\right)^{(2+\epsilon)\kappa},
\end{aligned}$$

where the last line follows from Lemma A.4.3.

Thus, we obtain that the following inequality holds for arbitrary $\epsilon \in (0,1)$:

$$\left((I+L)^{-1}\right)_{ij} \geq \frac{1}{2\sqrt{2\pi} \cdot \sqrt{\frac{\kappa}{\epsilon}} \cdot d^2(2d\ell+1)} \cdot \left(\left(1 - \frac{\epsilon^2}{2}\right)^{\frac{2}{\epsilon}+1} \cdot \left(1 - \frac{1}{2d\ell+1}\right)^{\frac{2}{\epsilon}+1}\right)^{\kappa}. \quad \text{(A.44)}$$

By setting $\epsilon$ such that $1/\epsilon = \left\lceil 2(d\ell)^{\frac{1}{2}} \right\rceil$ in (A.44), we obtain that:

$$\begin{aligned}
\left((I+\ell \cdot L)^{-1}\right)_{ij} &\geq \frac{1}{2\sqrt{2\pi} \cdot \sqrt{2\kappa \cdot \sqrt{d\ell}} \cdot d^2(2d\ell+1)} \cdot \left(\left(1 - \frac{1}{2d\ell}\right)^{4\sqrt{d\ell}+1} \cdot \left(1 - \frac{1}{2d\ell+1}\right)^{4\sqrt{d\ell}+1}\right)^{\kappa} \\
&\geq \frac{1}{4\sqrt{\pi \cdot \kappa \cdot \sqrt{d\ell}} \cdot d^2(2d\ell+1)} \cdot \left(\left(1 - \frac{4\sqrt{d\ell}+1}{2d\ell}\right) \cdot \left(1 - \frac{4\sqrt{d\ell}+1}{2d\ell+1}\right)\right)^{\kappa} \\
&\geq \frac{1}{4\sqrt{\pi \cdot \kappa \cdot \sqrt{d\ell}} \cdot d^2(2d\ell+1)} \cdot \left(1 - \frac{4}{\sqrt{d\ell}}\right)^{\kappa}. \quad \text{(A.45)}
\end{aligned}$$

Q.E.D.

**Step 3: Combine Temporal and Spatial Lower Bounds**

Combining the results of Steps 1 and 2 together, we know that the competitive ratio of any decentralized online algorithm is lower bounded by

$$\max\{1 + \frac{\mu^3(1-\sqrt{\lambda_T})^2}{48(\mu+1)^2\ell_T} \cdot \lambda_T^k, 1 + \Omega(\lambda_S^r)\} = 1 + \Omega(\lambda^k) + \Omega(\lambda_S^r).$$

## A.5 Proof of Corollary 3.3.6

In this appendix we prove a resource augmentation bound for LPC. To simplify the notation, we define the shorthand $a_T := \ell_T/\mu$ and $a_S := \ell_S/\mu$. $a_T$ and $a_S$ are positive real numbers. We first show two lemmas about the relationships between the decay factors $\rho_T$ and $\lambda_T$, and $\rho_S$ and $\lambda_S$.

**Lemma A.5.1.** *Under the assumptions of Theorem 3.3.1, we have $\rho_T^4 \leq \lambda_T \leq \rho_T^2$.*

*Proof of Lemma A.5.1.* Recall that $\rho_T$ is given by

$$\rho_T = \sqrt{1 - \frac{2}{\sqrt{1+2a_T}+1}}$$

in Theorem 3.3.1. Thus we see that

$$\rho_T^4 = \left(1 - \frac{2}{\sqrt{1+2a_T}+1}\right)^2 \leq \left(1 - \frac{2}{\sqrt{1+4a_T}+1}\right)^2 = \lambda_T.$$

On the other hand, we have that

$$\lambda_T - \rho_T^2 = \left(1 - \frac{2}{\sqrt{1+4a_T}+1}\right)^2 - 1 + \frac{2}{\sqrt{1+2a_T}+1} = \frac{4\sqrt{(1+2a_T)}\left(\sqrt{1+2a_T} - \sqrt{1+4a_T}\right)}{\left(\sqrt{1+2a_T}+1\right)\left(\sqrt{1+4a_T}+1\right)^2} \leq 0.$$

Q.E.D.

**Lemma A.5.2.** *Under the assumptions of Theorem 3.3.1, we have $\rho_S^{32} \leq \lambda_S$.*

*Proof of Lemma A.5.2.* Recall that $\rho_T$ is given by

$$\rho_S = \sqrt{1 - \frac{2}{\sqrt{1+\Delta a_S}+1}}$$

in Theorem 3.3.1. We consider the following three cases separately.

**Case 1**: $\Delta a_S \geq 224$. We have $\rho_S^{32} \leq \lambda_S$ in this case.

We first show that the following inequality holds for any positive integer $n_0$ and $x \in [0, 1/(2n_0)]$:

$$(1-x)^{2n_0} \leq 1 - n_0 x. \tag{A.46}$$

To see this, define function $g(x) = (1-x)^{2n_0} + n_0 x - 1$. Note that $g$ is a convex function with $g(0) = 0$ and

$$g\left(\frac{1}{2n_0}\right) = \left(1 - \frac{1}{2n_0}\right)^{2n_0} - \frac{1}{2} \leq e^{-1} - \frac{1}{2} < 0.$$

Thus, we see that $g(x) \leq 0$ holds for all $x \in [0, 1/(2n_0)]$. Hence (A.46) holds.

By (A.46), we see that

$$\begin{aligned}
\rho_S^{16} &= \left(1 - \frac{2}{\sqrt{1+\Delta a_S}+1}\right)^8 \\
&\leq 1 - \frac{8}{\sqrt{1+\Delta a_S}+1} \\
&\leq 1 - \frac{4\sqrt{3}}{\sqrt{\Delta a_S}} \\
&= \sqrt{\lambda_S}.
\end{aligned}$$

**Case 2**: $1 \leq \Delta a_S < 224$. We have $\rho_S^{28} \leq \lambda_S$ in this case.

To see this, note that $\rho_S^2 \leq 1 - \frac{2}{\sqrt{1+224}+1} = \frac{7}{8}$, and by Theorem 3.3.5, $\lambda_S \geq \frac{1}{3+3\cdot 1} = \frac{1}{6}$. Therefore, we see that

$$\rho_S^{28} \leq \left(\frac{7}{8}\right)^{14} \leq \frac{1}{6} \leq \lambda_S.$$

**Case 3**: $\Delta a_S < 1$. We have $\rho_S^4 \leq \lambda_S$ in this case.

To see this, note that

$$\rho_S^2 = \frac{\sqrt{1+\Delta a_S}-1}{\sqrt{1+\Delta a_S}+1} \leq \frac{\sqrt{\Delta a_S}}{4}.$$

Thus we see that

$$\rho_S^4 \leq \frac{\Delta a_S}{16} \leq \frac{\Delta a_S}{3+3\Delta a_S} = \lambda_S.$$

Q.E.D.

Now we come back to the proof of Corollary 3.3.6. By Theorem 3.3.3 and Theorem 3.3.5, we know that the optimal competitive ratio is lower bounded by

$$c(k^*, r^*) \geq 1 + C_\lambda \left(\lambda_T^{k^*} + \lambda_S^{r^*}\right)$$

and LPC's competitive ratio is upper bounded by

$$c_{LPC}(k, r) := 1 + C_\rho \left(C_3(r)^2 \cdot \rho_T^k + h(r)^2 \cdot \rho_S^r\right),$$

where $C_\lambda$ and $C_\rho$ are some positive constants. To achieve $c_{LPC}(k, r) \leq c(k^*, r^*)$, it suffices to guarantee that

$$C_\rho \cdot C_3(r)^2 \cdot \rho_T^k \leq C_\lambda \lambda_T^{k^*} \text{ and } C_\rho \cdot h(r)^2 \cdot \rho_S^r \leq C_\lambda \lambda_S^{r^*}.$$

Note that $C_3(r)$ can be upper bounded by some constant and $h(r)^2 \leq poly(r) \cdot \rho_S^{-\frac{r}{2}}$ under our assumptions. Applying Lemma A.5.1 and Lemma A.5.2 finishes the proof.

## A.6 Details in the multiproduct pricing application

### A.6.1 Proof of Lemma 3.5.1

The revenue maximization problem Equation (3.7) is equivalent to the following:

$$\begin{aligned} \min \quad & -\sum_{t=1}^{H}\sum_{v\in\mathcal{V}} x_t^v d_t^v = \sum_{t=1}^{H}\sum_{v\in\mathcal{V}}\left[x_t^v(-a_t^v + k_t^v x_t^v - \sum_{u\in N_v^1\setminus\{v\}} \eta_t^{(u\to v)} x_t^u - b_t^v x_{t-1}^v)\right] \\ \text{s.t.} \quad & 0 \le x_t^v \le \overline{p}_t^v \end{aligned} \tag{A.47}$$

Recall the definition of node, spatial, and temporal costs:

$$\begin{aligned} f_t^v(x_t^v) &:= \xi_t^v\left(x_t^v - \frac{a_t^v}{2\xi_t^v}\right)^2, \\ s_t^{(u,v)}(x_t^u, x_t^v) &:= |\gamma_t^{(u,v)}|\left(x_t^u - sgn\left(\gamma_t^{(u,v)}\right)\cdot x_t^v\right)^2, \\ c_t^v(x_t^v, x_{t-1}^v) &:= \frac{b_t^v}{2}\left(x_t^v - x_{t-1}^v\right)^2. \end{aligned}$$

Note that $f_t^v(x_t^v)$ is $\xi_t^v(x_t^v)^2 - a_t^v x_t^v$ plus a constant $(a_t^v)^2/(4\xi_t^v)$, and the interaction functions can be rewritten as

$$s_t^{(u,v)}(x_t^u, x_t^v) = |\gamma_t^{(u,v)}|\left((x_t^u)^2 + (x_t^v)^2\right) + 2\gamma_t^{(u,v)} x_t^v x_t^u, \quad c_t^v(x_t^v, x_{t-1}^v) = \frac{b_t^v}{2}\left((x_t^v)^2 + (x_{t-1}^v)^2\right) - b_t x_{t-1}^v x_t^v.$$

Summing, we see that

$$\sum_{t=1}^{H}\sum_{v\in\mathcal{V}}(f_t^v(x_t^v) + c_t^v(x_t^v, x_{t-1}^v)) + \sum_{t=1}^{H}\sum_{e\in\mathcal{E}} s_t^e(x_t^u, x_t^v) = \text{(Objective in (A.47))} + \sum_{t=1}^{H}\sum_{v\in\mathcal{V}}(a_t^v)^2/(4\xi_t^v)\,. \tag{A.48}$$

Hence the optimal solution of (A.47) is the same as the following problem:

$$\begin{aligned} \min \quad & \sum_{t=1}^{H} \sum_{v \in \mathcal{V}} f_t^v(x_t^v) + c_t^v(x_t^v, x_{t-1}^v) + \sum_{t=1}^{H} \sum_{e \in \mathcal{E}} s_t^e(x_t^u, x_t^v) \\ \text{s.t.} \quad & 0 \le x_t^v \le \overline{p}_t^v \end{aligned} \tag{A.49}$$

### A.6.2 Lemma A.6.1 and its proof

**Lemma A.6.1.** *Suppose the competitive ratio of our general cost minimization problem is $CR(k, r)$, which is a function of prediction horizon $k$ and communication radius $r$. Suppose $\sup_{(u,v)\in\mathcal{E},t\in[H]} a_t^u/a_t^v \le \widetilde{b}$, $\sup_{v\in\mathcal{V},t\in[H]} \frac{a_t^v}{\overline{p}_t^v} \le \widetilde{c}$, then the competitive ratio for the corresponding revenue maximization problem, defined as $rev(ALG)/rev(OPT)$, is at least $1-\frac{\eta}{2}(CR(k,r)-1)$, where $\Delta$ denotes the degree of the product network and $\eta \coloneqq \max\{2(\ell_f + \Delta\widetilde{b}\gamma)/\mu, \widetilde{c}/\mu\}$.*

We define $C \coloneqq \sum_{t,v}(a_t^v)^2/(4\xi_t^v)$. Suppose

$$CR(k,r) \cdot cost(OPT) \ge cost(ALG),$$

then

$$CR(k,r) \cdot (-rev(OPT) + C) \ge (-rev(ALG) + C).$$

Rearranging the terms yields

$$(CR(k,r) - 1) \cdot C \ge CR(k,r) rev(OPT) - rev(ALG). \tag{A.50}$$

To find a lower bound on $rev(OPT)$, we choose a pricing strategy such that $x_t^v = \frac{a_t^v}{\eta\mu} \ge 0$ where $\eta = \max\{2(\ell_f + \Delta\widetilde{b}\gamma)/\mu, \widetilde{c}/\mu\}$. We first check that the demand is always nonnegative

under this strategy:

$$\begin{aligned} a_t^v - k_t^v \frac{a_t^v}{\eta\mu} - \sum_{u \in N_v^1 \setminus \{v\}} \eta_t^{(u \to v)} \frac{a_t^u}{\eta\mu} + b_t^v \frac{a_{t-1}^v}{\eta\mu} &\geq a_t^v - k_t^v \frac{a_t^v}{\eta\mu} - \sum_{u \in N_v^1 \setminus \{v\}} \eta_t^{(u \to v)} \frac{a_t^u}{\eta\mu} \\ &\geq a_t^v - k_t^v \frac{a_t^v}{\eta\mu} - \sum_{u \in N_v^1 \setminus \{v\}} \widetilde{b}\gamma \frac{a_t^v}{\eta\mu} \\ &\geq a_t^v (1 - \frac{\ell_f + \Delta\widetilde{b}\gamma}{\eta\mu}) \\ &\geq \frac{a_t^v}{2}. \end{aligned}$$

Moreover,

$$x_t^v \leq a_t^v / \widetilde{c} \leq \overline{p}_t^v.$$

Hence this is a feasible price strategy.

We lower bound the optimal revenue:

$$\begin{aligned} rev(OPT) &\geq \sum_{t=1}^{H} \sum_{v \in \mathcal{V}} \frac{a_t^v}{\eta\mu} (a_t^v - k_t^v \frac{a_t^v}{\eta\mu} - \sum_{u \in N_v^1 \setminus \{v\}} \eta_t^{(u \to v)} \frac{a_t^u}{\eta\mu} + b_t^v \frac{a_{t-1}^v}{\eta\mu}) \\ &\geq \sum_{t=1}^{H} \sum_{v \in \mathcal{V}} \frac{a_t^v}{\eta\mu} \frac{a_t^v}{2} \\ &\geq \frac{2}{\eta} C. \end{aligned}$$

We further divide Equation (A.50) by $rev(OPT)$ to obtain

$$(CR(k,r) - 1) \frac{C}{rev(OPT)} \geq CR(k,r) - rev(ALG)/rev(OPT).$$

Since $CR(k,r) \geq 1$ for the cost minimization problem,

$$rev(ALG)/rev(OPT) \geq 1 - (CR(k,r) - 1) \frac{C}{rev(OPT)}.$$

This allows us to complete the proof as follows

$$rev(ALG)/rev(OPT) \geq 1 - \frac{\eta}{2}(CR(k,r) - 1).$$

### A.6.3 Additional plots

In Figure A.2, we present the computation time using different pricing policies across 30 simulated samples. Since the heuristics are closed-form solutions, they enjoy a much smaller computation time.

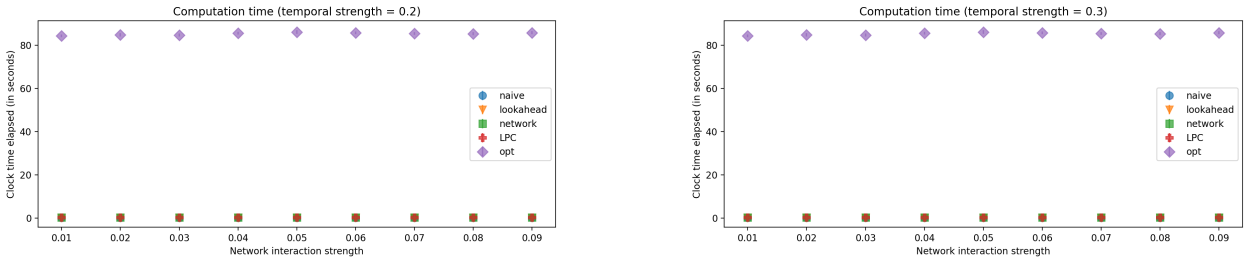


Figure A.2: Clock time elapsed for using different pricing policies.

In Figure A.3, we track price trajectories under different policies for randomly sampled products from our entire product set. We select a random instance and output the corresponding trajectories for 6 weeks. B00D2C6IFO product has a degree 7, B008H86SNA product has a degree 5 and B003KG9Z9S product has a degree 6. In this figure, the temporal interaction strength is 0.2 and the network interaction strength is 0.03. The right panel is the corresponding trajectory of the intercept $a_t^v$'s which can be viewed as the base demand of the product $v$.

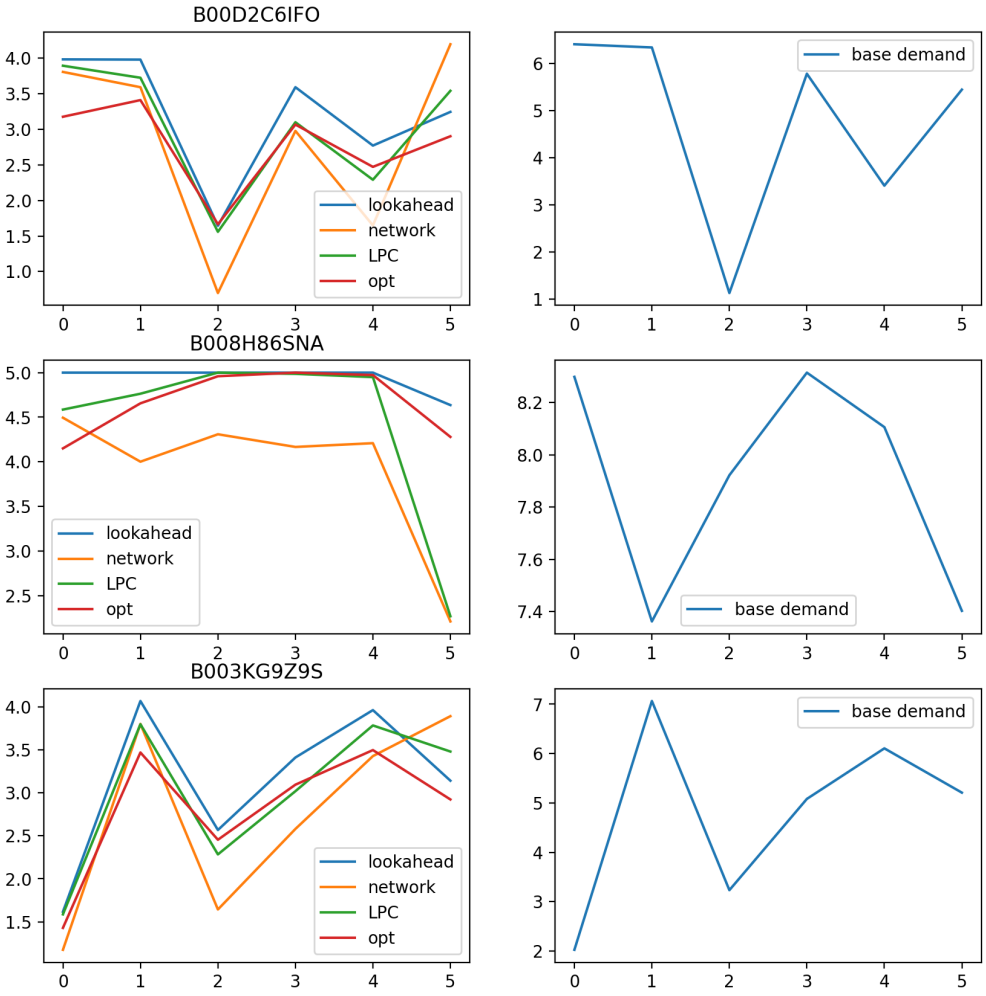


Figure A.3: Weekly pricing trajectories and base demands