SEARCH AND OPTIMIZATION WITH RANDOMNESS IN COMPUTATIONAL ECONOMICS: EQUILIBRIA, PRICING, AND DECISIONS

BY

SHANT BOODAGHIANS

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Assistant Professor Ruta Mehta, Chair
Professor Chandra Chekuri
Professor Sariel Har-Peled
Assistant Professor Yang Cai, Yale University

Abstract

In this thesis we study search and optimization problems from computational economics with primarily stochastic inputs. The results are grouped into two categories: First, we address the smoothed analysis of Nash equilibrium computation. Second, we address two pricing problems in mechanism design, and solve two economically motivated stochastic optimization problems.

Computing Nash equilibria is a central question in the game-theoretic study of economic systems of agent interactions. The worst-case analysis of this problem has been studied in depth, but little was known beyond the worst case. We study this problem in the framework of smoothed analysis, where adversarial inputs are randomly perturbed. We show that computing Nash equilibria is hard for 2-player games even when input perturbations are large. This is despite the existence of approximation algorithms in a similar regime. In doing so, our result disproves a conjecture relating approximation schemes to smoothed analysis. Despite the hardness results in general, we also present a special case of co-operative games, where we show that the natural greedy algorithm for finding equilibria has polynomial smoothed complexity. We also develop reductions which preserve smoothed analysis.

In the second part of the thesis, we consider optimization problems which are motivated by economic applications. We address two stochastic optimization problems. We begin by developing optimal methods to determine the best among binary classifiers, when the objective function is known only through pairwise comparisons, *e.g.* when the objective function is the subjective opinion of a client. Finally, we extend known algorithms in the Pandora's box problem — a classic optimal search problem — to an order-constrained setting which allows for richer modelling.

The remaining chapters address two pricing problems from mechanism design. First, we provide an approximately revenue-optimal pricing scheme for the problem of selling time on a server to jobs whose parameters are sampled *i.i.d.* from an unknown distribution. We then tackle the problem of fairly dividing chores among a collection of economic agents via a competitive equilibrium, which balances assigned tasks with payouts. We give efficient algorithms to compute such an equilibrium.

To my family and friends, who
have stood by me from the start.

Acknowledgments

First, I would like to express my deepest gratitude towards my Ph.D. advisor, Ruta Mehta. Very little of this thesis would have been possible without your diligent efforts, both as an advisor and as a mentor. You have never hesitated to tackle hard problems alongside your students when it was needed, you have introduced me to many new and interesting problems, and you have provided a wealth of opportunities to work with amazing academics. These experiences have been the highlight of my time here.

I am also grateful for the amazing colleagues and friends that I have found in the theory group here, and the welcoming community they form. I am grateful to Mitchell Jones, Sahand Mozaffari, Robert Andrews, Xilin Yu, Bolton Bailey, Patrick Lin, Ian Ludden, Vasilis Livanos, Phil Amortila, Charlie Carlson, Tanvi Bajpai, Meghan Shanks, Zander Kelley, Rucha Kulkarni, Pooja Kulkarni, Chao Xu, Belinda Tzen, Eliot Robson, Manuel Torres, Samir Khan, and Spencer Gordon, for having been my friends and peers in these wonderful years. Additionally, I would like to thank Robert, Sahand, and Mitchell, for always being willing to hear me out when I needed to share what I was working on. Your patience and curiosity is greatly appreciated, and my Ph.D. would not have been the same without it.

I have also had the fortune of working with many brilliant minds all around the world during my Ph.D.: Professor Ruta Mehta, Prof. Yishay Mansour, Gaurush Hiranandani, Prof. Sanmi Koyejo, Rucha Kulkarni, Prof. Karthekeyan Chandrasekaran, Prof. Stefano Leonardi, Federico Fusco, Philip Lazos, Prof. Aviad Rubinstein, Joshua Brakensiek, and Bhaskar Ray Chaudhury, in chronological order. I am very grateful to all of them for the time we have spent together solving these hard problems, and the opportunity to work with other excited minds on great papers. I would also like to thank Prof. Aviad Rubinstein and Prof. Stefano Leonardi for giving me the opportunity to visit them, and to thank Joshua Brakensiek and Federico Fusco, for being constant colleagues on these respective visits. My time spent with both groups was productive and memorable. Furthermore, I would like to thank Prof. Adrian Vetta for having started me along this path.

Finally, I would like to thank the many friends who have been a part of my life outside of the University, and back home. This thesis would not have been possible without the friends and my family who have always been supportive of me. To my parents and grandparents, to my brother Levon, to my friends back home Thomas, Bruce, and Terrence, and to Christine, I owe a debt of gratitude, for being there when I needed them, for teaching me to grow as a person, and for stimulating, enlightening conversations.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The classical study of microeconomics is concerned with the behaviour of rational, selfish agents in economic interactions, with topics ranging from the decision problems of the agents themselves, to the design of systems which incentivize desirable behaviour. The advent of computational resources and optimization techniques has allowed economists to ask these questions in the language of Algorithms and Complexity. This has come to be known as *Computational Economics*, which includes *Algorithmic Game Theory*, the computational study of classical game-theoretic problems such as equilibrium computation.

Algorithmic Game Theory as a field can be broadly divided into two main categories, which are reflected in the organization of this thesis. The first of these is *equilibrium computation*, which has centered around the notion of Nash equilibria in games — defined below — and the computational considerations in finding equilibria, which has led to its own subfield of complexity theory. The second category focuses on the optimization problems that a rational economic agent would consider, such as asking how to behave optimally as a single agent — referred to as *decision theory* — and asking how to set up a system or a market so that others behave in a desirable way — referred to as *mechanism design*.

Games are a fundamental model of interaction between rational, selfish agents, due to being both simple and expressive. The most widely studied solution concept for games is that of Nash equilibria, which capture stable states where no agent is incentivized to change their decisions, and is therefore central to understanding the eventual behaviour of rational agents in games. (See *e.g.* [1] for a survey.) Despite this, a sequence of seminal results established the worst-case complexity for the problem of Nash equilibrium computation, conjecturing that equilibria can not be found in quasi-polynomial time.[1] The complexity class PPAD was introduced to capture these problems [2], and Nash equilibrium computation was shown to be complete for this class [3]. Notably, approximately finding an equilibrium in a game with only two players is also PPAD-complete [4]. For this reason, and due to the importance of Nash equilibria, the community has turned its attention to *beyond-worst-case* analysis, which seeks to theoretically model the "in-practice" performance of algorithms, by arguing that worst-case instances are rare, or pathological. One of the most studied forms of beyond-worst-case analysis is known as *smoothed analysis*, which fixes an algorithm for some problem, and seeks to show that when an adversarially chosen input to this algorithm is randomly perturbed, then with high probability, the running time of the algorithm is a small function of the input size and of parameters of the perturbation's distribution function.

[1] A function is said to be quasipolynomial in n if it is bounded by $n^{O(\log n)}$.

In the first part of this thesis, we present smoothed analysis for Nash equilibrium computation for two game models. For two-player games, we disprove a conjecture relating smoothed complexity to approximation hardness, by giving a smoothed hardness result. However, in a model of multi-player collaboration games, we show that a simple local improvement algorithm has polynomial smoothed complexity. In the former, we show that the hardness-of-approximation results for 2-player implies hardness of equilibrium computation still holds when the parameters of the game are randomly perturbed. Specifically, we show the hardness of equilibrium computation holds even when the size of the random perturbation is in a regime where efficient approximation is possible. To the best of our knowledge, this beyond-worst-case hardness when approximation is possible is novel. Second, we show positive results in a model of many-player co-operative play. We consider the "natural dynamic", where players will greedily make changes that improve their individual outcome, until an equilibrium point is reached. We show that such natural dynamics converge quickly to an equilibrium, with high probability, when the game parameters are perturbed, despite taking exponentially many transitions in the worst case.

When games are implicitly defined, finding the optimal behaviour for a rational agent can also be a computationally challenging task. The computational complexity is even greater when determining the parameters of a game so as to incentivize desirable behaviour in players who optimize their reward. Such computational problems are known as decision problems and mechanism design, respectively. In the second part of the thesis, we study a variety of optimization problems, primarily stochastic, which arise in decision theory and mechanism design, and present efficient algorithms to solve them. These problems combine aspects of standard optimization theory with game-theoretic notions, such as the need to model selfish play in an allocation problem, or to minimize the cost of optimization when there is a financial cost attributed to exploration.

In the following sections, we introduce the historical context, and economic concepts, necessary for the first part of the thesis. We begin by introducing game-theoretic concepts, the specific notion of beyond-worst-case analysis considered, and our beyond-worst-case results in Nash equilibrium computation. We then go on to introduce the background for the second part of the thesis, laying out the economically motivated optimization problems that we consider.

1.1 GAMES, EQUILIBRIA, AND SMOOTHED ANALYSIS

As discussed above, a *game* is a model that captures the rewards and costs to rational economic decision-makers when they interact [1, 5]. Each *agent* or *player* has a finite collection of strategies to choose from, and depending on everyone's choice of strategy, each player receives some reward, possibly negative. Our goal is to understand how the players in the game are likely to behave, namely which strategies they are likely to choose.

As an example of a 2-player game, consider rock-paper-scissors. Each player has 3 strategies to choose from, namely rock, paper, or scissors, and their reward depends on the choice of both players. We say they both receive 0 if a tie occurs, and if a tie does not occur, the winning player receives 1 and the losing player receives -1. Note that the rewards here do not depend on which particular strategy was chosen that led to a win or a loss, but the model could have been modified to account for this.

Formally, a game $\mathcal{G}$ with n players and m strategies, consists of a collection of n *payoff functions* $u_1, \ldots, u_n$ which determine the payoff to each player. Without loss of generality, each player must choose a strategy from $[m] := \{1, \ldots, m\}$. We denote the simultaneous choice of strategy for each player as a vector $\boldsymbol{\sigma} \in [m]^n$. This is known as the *strategy profile*. The i-th payoff function determines the payoff to player $i \in [n]$ as a function of the strategy profile, and is denoted by $u_i : [m]^n \to \mathbb{R}$.

A player wishes to maximize their payoff, in response to the strategy choice of the other players. It is common to denote the strategy profile of all players except the i-th as the $(n-1)$-dimensional vector $\boldsymbol{\sigma}_{-i}$. Furthermore, writing $\boldsymbol{\sigma} = (\sigma_i, \boldsymbol{\sigma}_{-i})$ highlights the choice of the i-th player in $\boldsymbol{\sigma}$. Thus, player i would prefer to select σ_i over σ_i' as a response to $\boldsymbol{\sigma}_{-i}$ if $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) > u_i(\sigma_i', \boldsymbol{\sigma}_{-i})$.

Players may wish to randomize their strategy, and seek to maximize their expected reward. As an example, in the rock-paper-scissors game introduced above, no pair of deterministic strategies is stable, but if both players uniformly sample between all three options, then it is not in either's interest (in expectation) to deviate. Let Δ_m denote the m-dimensional probability simplex, *i.e.* the space of probability mass functions supported on the integers 1 through m, represented as a vector. Formally, each player chooses a distribution $\pi_i \in \Delta_m$ over the m possible choices. Let $\boldsymbol{\pi}$ denote the simultaneous random strategy choice for all players. We assume that the random strategies of the different players are independently distributed. The notation of $\boldsymbol{\pi}_{-i}$ is defined as in the deterministic case. The payoff to player i is now their *expected* payoff, denoted by $u_i(\boldsymbol{\pi}) := \mathbb{E}_{\boldsymbol{\sigma}\sim\boldsymbol{\pi}}[u_i(\boldsymbol{\sigma})]$, in an abuse of notation.

1.1.1 Nash Equilibria

A strategy profile is at an *equilibrium* if no single player can benefit by changing their strategy, assuming the strategies of the remaining players are kept fixed.[2] An equilibrium is said to be *pure* if none of the players are randomizing. We can assume that rational economic agents will behave according to some equilibrium state of the game, as otherwise at least one player is incentivized to move away from the current state. This effectively models Adam Smith's *invisible hand of the market.* In this sense, the study of equilibria allows us to predict agent behaviour. The economic merits of this definition have been critiqued (see *e.g.* [6, ch. 1]), but the concept has nonetheless seen much interest.

Nash equilibria capture stable states when agents are randomizing over strategies, and wish to maximize their expected returns, as introduced above. John Nash popularized this concept by showing that such equilibria must exist in any game [7], which cemented their importance as a solution concept. This is formalized in the following.

Definition 1.1 (Nash Equilibrium [7])**.** Let $\mathcal{G}$ be a game with n players and m strategies each, with payoff functions $u_i : (\Delta_m)^n \to \mathbb{R}$ for all $1 \leq i \leq n$, the abuse of notation introduced above. Then a *Nash equilibrium* of $\mathcal{G}$ is a (randomized) strategy profile $\boldsymbol{\pi}$ such that no player prefers to deviate when all other players' strategies remain fixed, and thus all players are simultaneously playing optimally in response to the other players. Formally,

$$u_i(\pi_i, \boldsymbol{\pi}_{-i}) \geq u_i(\pi_i', \boldsymbol{\pi}_{-i}) \quad \forall\ \pi_i' \in \Delta_m \qquad \text{for every player } i. \tag{1.1}$$

Nash's original proof of the existence of these equilibrium points consisted of providing a procedure to update the players' strategy distributions to better respond to their adversaries' decisions, and showing that there must exist a fixed-point to this procedure.

One may also choose to weaken this notion by requiring only approximately optimal play. Here, we want a profile where no player can benefit *too much* by deviating.

Definition 1.2 (ϵ-approximate Nash Equilibrium)**.** Let m, n, and the $u_i : (\Delta_m)^n \to \mathbb{R}$ be as above. Then, an *ϵ-approximate Nash Equilibrium* (ϵ-NE) is a (randomized) strategy profile $\boldsymbol{\pi}$ such that no player can benefit by more than ϵ by deviating when all other players' strategies remain fixed. Formally,

$$u_i(\pi_i, \boldsymbol{\pi}_{-i}) \geq u_i(\pi_i', \boldsymbol{\pi}_{-i}) - \epsilon \quad \forall\ \pi_i' \in \Delta_m \qquad \text{for every player } i. \tag{1.2}$$

[2] More general notions of equilibria exist, which require that no group of players be able to benefit as a coalition, but they are not relevant to the work presented in this thesis.

It is worth noting that the payoff from any randomized strategy is a convex combination of payoffs from deterministic strategies, and therefore the requirement to perform better than any randomized strategy can be weakened to comparing exclusively to deterministic strategies. Formally,

$$\begin{aligned} & u_i(\pi_i, \boldsymbol{\pi}_{-i}) \geq u_i(\pi_i', \boldsymbol{\pi}_{-i}) - \epsilon \quad \forall\ \pi_i' \in \Delta_m \\ \iff & u_i(\pi_i, \boldsymbol{\pi}_{-i}) \geq u_i(\sigma_i', \boldsymbol{\pi}_{-i}) - \epsilon \quad \forall\ \sigma_i \in [m] \end{aligned} \tag{1.3}$$

1.1.2 Bilinear Notation for 2-Player Games

For clarity of presentation, we introduce notation specific to the case of 2-player games. This case is of interest as it is conceptually simpler while still being computationally hard. Let $\mathcal{G}$ be a game with 2 players choosing from m strategies each, then the payoff functions u_1 and u_2 are represented by $m \times m$ matrices A and B respectively. Let $A_{i,j} := u_1\big((i,j)\big)$ and $B_{i,j} := u_2\big((i,j)\big)$, where the arguments are pure (deterministic) strategy pairs. Note that, effectively, the first player selects the row of the payoff matrices that the players observe, and the second player selects the column, making them the *row player* and *column player*, respectively.

When the players randomize, let $\boldsymbol{x} \in \Delta_m$ denote the distribution over the strategies for the row player, and $\boldsymbol{y} \in \Delta_m$ for the column player. The expected payoff to the row and column players are respectively given by

$$\sum_{i=1}^{m}\sum_{j=1}^{m} x_i y_j A_{i,j} = \boldsymbol{x}^\top A \boldsymbol{y} \qquad \text{and} \qquad \sum_{i=1}^{m}\sum_{j=1}^{m} x_i y_j B_{i,j} = \boldsymbol{x}^\top B \boldsymbol{y}\ . \tag{1.4}$$

Thus, the pair $(\boldsymbol{x}, \boldsymbol{y})$ is an ϵ-approximate Nash Equilibrium if

$$\boldsymbol{x}^\top A \boldsymbol{y} \geq \boldsymbol{e}_i{}^\top A \boldsymbol{y} - \epsilon \qquad \forall\ 1 \leq i \leq m\ , \tag{1.5a}$$

$$\boldsymbol{x}^\top B \boldsymbol{y} \geq \boldsymbol{x}^\top B \boldsymbol{e}_j - \epsilon \qquad \forall\ 1 \leq j \leq m\ , \tag{1.5b}$$

where $\boldsymbol{e}_i$ denotes the i-th standard basis vector.

1.1.3 Nash Equilibrium Computation and the class PPAD

As Nash equilibria are helpful in understanding the mechanics of a game, the computational task of finding the equilibrium points became of interest. We assume here that the utility functions are given explicitly as lookup tables for every deterministic input. In this case, we say that the game is given in *normal form.* A 2-player game is in normal form if the payoff matrices A and B defined above are given in the input.

The first algorithm for finding equilibria in normal-form 2-player games was given by Lemke and Howson [8]. The algorithm walks along the vertices of an appropriately defined polytope and terminates at an equilibrium. The procedure is guaranteed to terminate, but may walk through exponentially many vertices. It was hoped that due to its similarity to the simplex algorithm for linear programming, this algorithm would eventually give way to more efficient algorithms for equilibrium computation. However, no sub-exponential algorithms were found, and the problem was conjectured to be hard. This led to the introduction of the complexity class PPAD [2], for which Nash equilibrium computation in normal-form games was shown to be complete in the many-player setting [3]. The class PPAD captures search problems whose solution is guaranteed to exist via a characterization as the fixed-point of an iterative procedure [2]. The algorithm of Lemke and Howson [8] is an example of this, as the rule for walking along the polytope's vertices is an update function that reaches a fixed-point at an equilibrium. Notably, Nash's original proof of the existence of equilibria [7] goes through the Brouwer Fixed Point Theorem, and was updated to use the stronger Kakutani fixed-point theorem at the time of publication.

Approximation Algorithms. In the multi-player setting, a first sub-exponential-time algorithm for finding approximate equilibria was shown by Lipton, Marakakis, and Mehta [9]. They give a quasi-polynomial approximation scheme, showing that ϵ-approximate equilibria can be found in $n^{O(\log n/\epsilon^2)}$ time. This running time is conjectured to be tight for constant ϵ, despite the algorithm being straightforward. Furthermore, assuming the exponential time hypothesis for PPAD, the running time is indeed tight for constant ϵ [10]. Subsequent work has given efficient methods for finding equilibria for special cases of games — mostly approximate equilibria — such as sparse games [11], low-rank games [12, 13], positive-semidefinite games [14], anonymous games [15, 16], and tree games [17, 18, 19].

Worst-Case Hardness of Approximation. In general, 2-player, m-strategy games, Nash equilibrium computation remains PPAD-complete even computing an ϵ-approximate equilibrium with $\epsilon = m^{-c}$ for some constant $c > 0$ [4]. Note that finding approximate equilibria allows us to work with finite-precision arithmetic. Since the publication of this first hardness result, many special cases have been shown to be hard as well, including sparse games [20, 21], low-rank games [22], anonymous games [23], and tree games [24].

1.1.4 Beyond the Worst Case: Smoothed Analysis

The mostly-negative results discussed above for the worst-case complexity of Nash equilibrium computation, and the economic importance of the concept, have led to the following question: *Can Nash equilibria be found more efficiently beyond the worst case?* In the average case, when payoff entries are sampled *i.i.d.*, exact equilibria can be found in quasi-polynomial time with high probability [25]. In this thesis, we consider this question under the framework known as *smoothed analysis*. Before stating our results on Nash equilibrium computation, we first introduce the framework, its relevance, and importance.

Smoothed analysis is one of the most important techniques for beyond-worst-case analysis in the literature. It was introduced by Spielman and Teng [26] to explain why the Simplex algorithm for solving linear programs ($\max \langle \boldsymbol{c}, \boldsymbol{x} \rangle$ s.t. $A\boldsymbol{x} \geq \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0}$) performs well in practice despite worst-case instances requiring exponential time [27]. In their model, Gaussian $\mathcal{N}(0, \sigma^2)$ noise is added to each of the entries of the constraints $(A, \boldsymbol{b})$ and cost vector $\boldsymbol{c}$, *i.i.d.*, and the performance of the simplex algorithm is analyzed probabilistically.

As a general framework, smoothed analysis consists of showing that randomly perturbed inputs have efficient algorithms with running time guarantees that depend on "size parameters" of the input distributions, such as variance, or density upper bounds. Thus, an adversary which chooses a point-mass on a worst-case input (zero variance, infinite density) will get worst-case performance, and an adversary which samples the input from a well-spread distribution (maximized variance, minimized density) will approximate average-case performance. In this sense, smoothed analysis interpolates between worst-case and average-case analysis. The goal is to show that when the smoothing noise is "small", algorithms are efficient even near "bad" instances. Formally, we wish to show that with high probability, the running time of the algorithm is inversely polynomial in a spread-parameter for the smoothing distribution, such as variance.

Intuitively, such performance guarantees show that bad instances are "scattered", in a probabilistic sense. This would suggest that for any "real-world" problems, they are unlikely to fall into a bad instance, and may even benefit from measurement noise or from modelling imprecision — at least in running time. Notice that this framework is oblivious and does not require the problem to lie in a class of "reasonable" instances. It also directly bounds the fraction of instances which are "bad", showing that they are both few and also far between.

Smoothed analysis has been applied widely since its inception, often used to analyze heuristics used in practice. Among these are TSP heuristics [28], k-means methods [29], edit distance [30], local search methods for cuts [31, 32], and many more. The smoothed analysis of certain special cases of integer linear programming was also characterized in terms of its worst-case complexity [33], notably that a problem has polynomial smoothed complexity if and only if it admits a pseudo-polynomial-time algorithm [34], since we can approximate by truncating input parameters. A thorough survey of early results is presented in two earlier publications of Spielman and Teng [35, 36], and a more recent survey is given by Manthey and Röglin [37].

Nash Equilibrium Computation in 2-Player Games. Recall that we have asked whether Nash equilibrium can be efficiently found in 2-player games, in a beyond-worst-case sense. We ask if randomly perturbing the entries of the payoff matrices will help with finding equilibria, with high probability.

Giving evidence for a "yes" answer, it was shown that when the parameters of the game are sampled fully at random and *i.i.d.* then equilibria are easy to find [25]. Furthermore, it is known that when equilibria are robust to input perturbations, finding equilibria can be done efficiently [38]. However, the original approximation-hardness result of [4] argued that, since it is PPAD-hard to find n^{-c}-approximate equilibria in $n \times n$ games with $O(1)$-sized payoff values, then equilibria cannot be found efficiently after n^{-c}-sized perturbations under standard conjectures. This is because an exact equilibrium of a perturbed game is by definition an approximate equilibrium of the original, unperturbed game, where ϵ is the size of the perturbations. Thus, a smoothed-efficient algorithm would allow for efficient, randomized, n^{-c}-approximation algorithms, by artificially perturbing the input game.

These two lines of work leave a gap when the payoff values are $O(1)$, but the perturbations are $\Omega(1)$. We show in Chapter 2 that it is hard to find equilibria in this regime, as these also imply efficient algorithm for n^{-c}-approximate NE, a PPAD-complete problem. This is despite the existence of efficient approximation algorithms in the $\epsilon = \Omega(1)$ regime [9]. Inspired by the smoothed-hardness result of [4] which relied on approximation-hardness, Spielman and Teng [35] asked in a survey whether there was a connection between the two. The result of Chapter 2 answers this in the negative, as we have a problem which admits approximation algorithms, but no smoothed efficient algorithms, under standard conjectures.

1.1.5 Potential Games, the class PLS, and Network Coordination Games

Potential games are an important class of games that has seen much interest in the literature. Classic examples of potential games include *congestion games*, which model road or internet traffic with selfish drivers or packets, and *network coordination games*, which model social coordination among many agents. Potential games are so-called since they are equipped with a global *potential function* that captures every player's possible gains. Formally, the potential function is a mapping from the space of strategy profiles $\Phi : [m]^n \to \mathbb{R}$, such that when a single player changes their choice of strategy, the potential function should change by the same amount as that player's reward,[3] regardless of which player it is. Formally,

$$\Phi(\sigma_i', \boldsymbol{\sigma}_{-i}) - \Phi(\sigma_i, \boldsymbol{\sigma}_{-i}) \;=\; u_i(\sigma_i', \boldsymbol{\sigma}_{-i}) - u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) \qquad \forall\ i,\ \sigma_i,\ \sigma_i',\ \boldsymbol{\sigma}_{-i}\ . \tag{1.6}$$

In this sense, an improvement for a player is an increase in the potential, and at an equilibrium, when no player can improve, we are at a local maximum of the potential function. This immediately implies the existence in potential games of equilibria where no player randomizes, known as *pure Nash equilibria*. These can be found greedily by a potential-ascent algorithm.

The Complexity Class PLS. The complexity class associated to such potential-following algorithms, not just for equilibrium computation, is known as PLS, and finding pure Nash equilibria in potential games is a PLS-complete problem. Furthermore, a canonical complete problem for this class is that of finding a local-maximum to an integer-valued function represented as a boolean circuit [40]. Here, we are looking for an input whose output cannot be increased by flipping exactly one input bit. Another PLS-complete problem is finding a locally maximal cut in a graph [41], defined as a cut whose value cannot be improved by moving a single node across the cut. Although it is widely conjectured that PLS is unlikely to lie in P [10, 42, 43], we note that problems in this class admit local-search algorithms [40], which have been observed to be empirically fast [40, 44, 45], but require exponential time in the worst case [41, 46].

Network Coordination Games. An important subclass of potential games is that of *network coordination games*, which model collaboration between many agents. These games naturally arise in various settings like social networks, biological networks, routing and congestion on roads, etc. [47, 48, 49, 50, 51], and have been extensively studied in various areas like economics, learning, networks, and more [52, 53, 54, 55, 56]. A network coordination game is represented by an undirected *game graph* $G = (V, E)$, where the nodes are the players, and each edge $uv \in V$ represents a 2-player game (A, B) where $A = B$, so both players receive the same reward, known as a *coordination game*. The player-nodes simultaneously play in all incident game-edges, and must choose one strategy to play in all game edges. The

[3] Many generalizations of the concept of potential games which weaken the constraints on the potential function exist in the literature (see *e.g.* [39]), though they will not be relevant to this thesis.

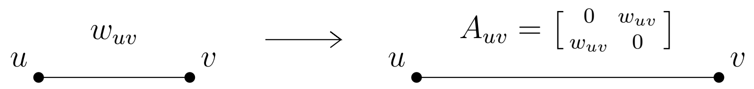
w_{uv}
u
v
$A_{uv} = \left[\begin{smallmatrix} 0 & w_{uv} \\ w_{uv} & 0 \end{smallmatrix}\right]$
u
v

Figure 1.1: The reduction from local-max-cut to 2-strategy network coordination games: Each edge (u, v) is mapped to a game-edge where the strategy chosen by the players represents the cut-side they have chosen. They receive as reward the edge's weight if they choose different sides, *i.e.* the edge spans the cut, or zero, if they choose the same side. Note that the potential function (1.7) is exactly the total cut value.

players receive the sum of the payoffs from each incident edge.

We denote the number of strategies for each player in these games as k, to not confuse with the parameter m for the number of edges of G. Thus, every edge in G is labelled with a $k \times k$ payoff matrix A_{uv}. Once every player chooses a strategy represented by the profile $\boldsymbol{\sigma}$, the payoff value for each edge is fixed, and each player gets the sum of the payoffs on its incident edges, $\sum_{v:uv\in E} A_{uv}(\sigma_u, \sigma_v)$ for a fixed u. Network coordination games are potential games, as witnessed by the following potential function:

$$\Phi(\boldsymbol{\sigma}) = \sum_{uv\in E} A_{uv}(\sigma_u, \sigma_v) \ . \tag{1.7}$$

Observe that when a single player changes their strategy, the only terms in the sum which are affected are from the edges incident to u, and the change is exactly the change to u's payoff. Thus, network coordination games are potential games, and always admit pure equilibria. However, computing a pure equilibrium in network coordination games is a PLS complete problem, even when $k = 2$, by a reduction from the PLS-complete problem of finding a locally maximal cut in a graph [57], introduced above. See Figure 1.1 for an overview of the reduction.

Smoothed Equilibrium Computation in Network Coordination Games. In Chapter 3 we study the smoothed complexity of the natural greedy algorithm for computing pure Nash equilibria in network coordination games: the algorithm arbitrarily chooses a player who can improve their payoff, and arbitrarily chooses an improving strategy. We show that this greedy algorithm converges in quasi-polynomially many steps with high probability, when the entries of the A_{uv} matrices are perturbed. This result extends recent results on the smoothed analysis of the greedy algorithm for local-max-cut [31, 32].

1.1.6 Smoothness-Preserving Reductions.

Note that standard Karp reductions do not suffice to extend a smoothed efficient algorithm from one problem to another. This is because, among other things, such a reduction needs to ensure that independently perturbed parameters of the original problem produce independent perturbations of all parameters in the reduced problem. In Chapter 4, we introduce a notion of a *smoothness-preserving reduction*, which to the knowledge of the author, has not been studied prior to the work on which the chapter is based [58].

Such a reduction allows us to translate smoothed analysis results from one problem to another. Consider a reduction (f, g) from instances of a problem $\mathcal{P}$ to instances of $\mathcal{Q}$. Given an instance P of $\mathcal{P}$, $f(P)$ is an instance of $\mathcal{Q}$ such that, if y is the solution to $f(P)$, then $g(y)$ is the solution to P. If we were to directly apply this reduction to a smoothed instance of $\mathcal{P}$, we would have a randomly distributed P, and map it to the random instance $f(P)$. However, if $\mathcal{Q}$ has a smoothed efficient algorithm, this result of smoothed efficiency only holds when the random input satisfies certain distributional constraints. Therefore, unless f is chosen carefully, we may not satisfy this conditions, depending on how P is sampled.

We argue that if f maps each individual input variable to some output variable, without combining them, then most smoothed analysis results are likely to be translatable via such a reduction. However, reductions of this form are rare. Instead, we show that when f is a full-rank linear transformation, then reductions to the types of problems we consider in this thesis will also allow smoothed-efficient algorithms to be translated back to the original problem. A formal description of this, and two such reductions, are given in Chapter 4. These rely on the proof methods presented in Chapter 3 in the smoothed analysis of Nash equilibrium computation for network coordination games, which were originally introduced in the study of local max cut [31, 32].

1.2 DECISION THEORY AND OPTIMAL PRICING

In the second part of this thesis, we focus on optimization problems that may be faced by a rational agent in economic decision making, broadly categorized as *decision theory*, and *mechanism design*. In the former, an agent is maximizing their own utility in an economic setting, and we view their task through the lens of optimization. In the latter, an agent has control over the parameters of a game, and seeks to influence behaviour. Recall that in the definition of a game, it was assumed that agents have un-bounded computational power. In some settings, however, the space of actions an agent may choose from is combinatorial, and determining the optimal decision to take for a given agent in a given situation becomes a nontrivial question.

In decision theory [59, 60], the problems are mostly economically motivated optimization problems, where an agent must expend resources to explore their options and make a utility-optimal choice. These include many classical optimal-stopping questions, such as the *secretary problem* [61, 62].

In mechanism design, when the agent being considered has the power to design the game, *e.g.* if they are a planner or a monopolist, they may wish to incentivize some behaviour in the players. These problems are often cast in the language of optimization. See [63] for a survey on the classical results and proof techniques of mechanism design. In this thesis, we turn our attention to *pricing problems*, which focus on the mechanism designer's task when they only have control over prices. The goal may be profit maximization, or to set prices and rewards to counteract issues in the fairness of an allocation.

1.2.1 Decision Problems

We consider first two optimization problems with economic motivations and non-standard inputs. First, we ask how to design an optimal stopping rule in a variant of the secretary problem known as the Pandora's box problem, when we add precedence constraints. Second, a common task in machine learning is choosing an optimal classifier. We consider the problem when the objective function is not known, and we can only ask comparison queries to the stakeholders. The following sections formalize these problems.

The Pandora's Box Poblem

The first decision problem studied in this thesis concerns an extension of the classical Pandora's box problem, first introduced by Weitzman [64] to model the task of selecting one of many alternatives when there is a cost to evaluating each option but the distribution of possible rewards is known. As an example, consider the task of hiring for a specialized position that only needs one candidate. We have a distribution over the skill of each candidate from information found in their résumé, but time and resources must be spent to interview a candidate to accurately gauge their skill level. Any one of the evaluated candidates may be hired, but there is only one position. We seek to develop optimal-in-expectation strategies to maximize the quality of the final outcome, minus the cost of evaluation. This sets up a trade-off between exploration and exploitation, as further interviews give decreasing marginal returns, for a similar cost.

Formally, we — or Pandora — are faced with boxes $b_1, \ldots, b_n$. Box b_i has a random payoff X_i distributed according to $\mathcal{D}_i$, which is known, and it costs c_i to observe X_i's value, or formally, to sample $\mathcal{D}_i$. We only get to keep one of the rewards, and so our objective function is

$$\mathbb{E}_{S,X_1,\ldots,X_n}\left[\max_{i\in S} X_i - \sum_{i\in S} c_i\right] , \tag{1.8}$$

where S is the random subset of boxes that the algorithm chooses to open, which can depend adaptively on the observed X_i values. Our goal is to maximize this expected objective over the space of adaptive strategies.

Notice that a box only has value in this objective if its reward is greater than the greatest reward seen in the past. Formally, if S_{t-1} is the (random) set of the first $t-1$ boxes opened, then the decision of which box to open as the t-th should only depend on the set of remaining boxes, and on $\max_{i\in S_{t-1}} X_i$, independent of which box achieved this maximum, and at what price. To this end, Weitzman [64] introduced the *reservation price* of a box, defined as the solution σ_i to the following equation:

$$\mathbb{E}_{X_i}\left[\max\{0, X_i - \sigma_i\}\right] = c_i \ . \tag{1.9}$$

Informally, σ_i is the value of $\max_{j\in S_{t-1}} X_j$ at which we are indifferent between opening b_i as the t-th box or stopping, since the expected marginal gain is equal to the price. Weitzman's proves that the optimal strategy opens the boxes in decreasing order of σ_i and stops when the greatest seen value is greater than the next reservation value. Thus, the order of exploration is independent of the values seen and can be pre-computed. This simple strategy is optimal despite the breadth of complex strategies available to choose from.

Order Constraints. In Chapter 5, we extend the model of Weitzman by constraining the order in which the boxes can be opened. This models, for example, a research-and-development process, where new surveys or product trials are costly, but have a chance of leading to a profitable product, and certain questions must be answered before others can be asked. We can also extend the example of the hiring problem in the original Pandora's box problem with the order constraint. Now, each employee is associated with a collection of boxes, which must be opened in order: the first one is the cost of interviewing, but gives no reward, and the future ones model the expected revenue generated by an employee in a year, and learning the true revenue value comes at the cost of their salary. We may choose to terminate search along that path by laying off the employee, or proceed along the path, paying their salary each time.

These order constraints add complexity to the search problem, requiring us to trade off between exploring in depth a current "branch" of the order constraints, or going on to explore the start of another "branch". The order constraints are formally modelled as a directed acyclic graph, where the nodes index the boxes, and a box can be opened if at least one of its parents has been opened. The more natural model where all parents need to be opened was considered, but we found no results in this setting. In Chapter 5, we show that when the order constraint is rooted tree, or forest of rooted trees, then a Weitzman-like strategy is still optimal. There exist efficiently computable thresholds for the boxes, and the optimal strategy is to open the box with the largest available threshold, or to stop if the remaining thresholds are less than the largest value seen in the past. When the order constraints are more general DAGs, we give an example where the optimal order of exploration cannot be fixed *a priori*, and must be chosen adaptively. We also show that it is NP-hard to solve the problem approximately, even in small-depth, low-degree DAGs. Finally, we extend past work to show that the approximation factor lost when restricting ourselves to non-adaptively choosing which boxes to open is bounded. Using these results, we give approximation algorithms in a special case.

The notion of approximation used in this context is not the traditional multiplicative factor of the objective function. Instead, we say a policy is an α-approximation of the optimal policy if it earns at least $\mathbb{E}[\alpha \cdot \max_{i \in S^*} X_i - 1 \cdot \sum_{i \in S^*} c_i]$ in expectation. Thus, we are discounting the expected return by a multiplicative factor of the earned rewards, ignoring costs. To motivate this notion of approximation, note that a single box with no reward but adversarial cost may be placed as a root of the DAG, with the price chosen to bring the optimal reward arbitrarily close to 0. In such a setting, any algorithm which yields any positive-valued expected profit — let alone a constant-factor approximation to the optimum — is effectively optimal. Furthermore, this notion of approximation has been used in similar contexts.

Optimal Classification

The second problem considered is the task of finding an optimal classifier in a low-information setting. Classification is a common machine learning problem, where elements X from a population $\mathcal{X}$ have correlated labels $Y \in \{1, \ldots, k\}$, and the goal is to develop a classifier $h : \mathcal{X} \to [k]$ which agrees with Y often. Common examples of classification problems include loan decisions, where $X \in \mathcal{X}$ represents the details of a bank loan application, such as loan size, along with measures of customer creditworthiness, and the label Y is the (random) indicator for whether the customer(s) represented by X will re-pay their debt. Another example is diagnostics, where $X \in \mathcal{X}$ represents a patient's demographics, lifestyle, and symptoms, and the label Y is the (random) indicator for whether or not the patient(s) represented by X suffer from some illness. Both examples highlight the economic importance of selecting optimal classifiers.

Formally, we assume that X is sampled from a probability distribution over $\mathcal{X}$, and Y is jointly distributed with X. Thus it need not be a deterministic function of X. In most standard settings, it is always to our advantage to assume that h is a deterministic function of X, despite Y's randomness. Usually this is because we want h to be the indicator of the most likely label Y, therefore maximizing the probability of overlap. To measure the performance of a classifier, we must introduce the *confusion matrix* $C \in [0,1]^{k\times k}$. Here,

$$C(h)_{ij} := \Pr_{X,Y,h}[Y = i, h = j] \ , \tag{1.10}$$

and we recall that Y, and optionally h, are jointly distributed with X. Clearly, we would like to choose h in such a way as to maximize the diagonal terms, and minimize the off-diagonal terms. Unfortunately, this is a multi-objective problem, and a specific objective function must be tailored to the specific application. Instead, we assume there is a performance metric $\phi : [0,1]^{k\times k} \to \mathbb{R}$, which is a function on the confusion matrix, and does not otherwise depend on h. The classification problem then becomes the task of finding a ϕ-maximizing classifier h.

Most results concerning optimal classification in the machine learning literature focus on the task of learning the joint distribution on x and y, so as to choose an optimal classifier. Common objective functions include linear functions of the confusion matrix, or ratios of linear functions [65, 66]. However, we focus on the problem of learning the parameters of the objective function in these restricted families, assuming knowledge of the underlying distributions. Learning a performance metric which correlates with human preferences has been studied before [67, 68]; however, these studies learn a regression function over some predefined features which is fundamentally different from our problem. In addition, while [69, 70] address how one might qualitatively choose between metrics, no work addresses the task of choosing the optimal metric from user feedback.

Comparison Oracles. In our setting, we assume that instead of being given access to the oracle ϕ, we have an oracle which, given two classifiers h and h', can determine whether or not $\phi(C(h)) > \phi(C(h'))$. This oracle represents some form of external input on the quality of classifiers. For example, in the application to classification for bank loans, a bank may be offered to test two different classifiers and get a sense of when and how often they mis-predict loan repayment, and in which demographic groups, whether it makes more false positive or false negative errors, etc. The bank then has an opinion on which of the two it prefers, and can report this, even without necessarily having explicit knowledge of their own objective function.

In Chapter 6, we study the problem of finding optimal classifiers in the $k = 2$ setting — known as binary classification — when the objective function is not known, but is known to lie in a restricted family of classifiers. This restriction is required to ensure that comparison oracles can be used efficiently, and commonly used objectives fall in the families considered. We give bounds on the least number of oracle queries required to obtain desired approximation guarantees in the quality of the classifier. Furthermore, when restricting to these common classes of objective functions, our algorithm will simultaneously learn the objective ϕ while finding an approximately ϕ-optimal classifier. Furthermore, we assume that the oracle can be *noisy*, *i.e.* may make a mistake when $\phi(C(h)) \approx \phi(C(h'))$. This will be formalized in Chapter 6.

1.2.2 Optimal Pricing

We consider here two mechanism design problems concerning optimal price selection. First, we ask how to price time on a server with selfish jobs arriving online from an underlying distribution. Second, we tackle the problem of computing competitive equilibria for chore allocation, a notion of fair and efficient allocation which assigns tasks and payments to balance disutility across agents. The following sections formalize these problems.

Server Scheduling, Posted Prices, and Revenue Maximization

A common modern application of mechanism design is to the problem of allocating time on a server, such as in cloud computing applications (*cf.* [71, 72]). Due to the popularity of these services, and the recurring, frequent, and measurable interactions between the services and their customers, and the technical nature of the product, much work has been done by practitioners in understanding the problem of pricing computational resources.

The problem is of theoretical interest as well: since the server capacity is limited, mechanisms must trade off immediate revenue for future supply, and must incentivize good behaviour from the agents who wish to schedule time on the server, while being profitable, or attaining some other objective. For example, Chawla et al. [73] recently studied "time-of-use" pricing mechanisms, to match demand to supply with deadlines and online arrivals. Their result assumes large-capacity servers, and seeks to maximize welfare in a setting in which the jobs arriving over time are not identically distributed. Further, [74, 75] give methods for optimizing the sum of the values of schedules jobs, known as the social welfare.

In Chapter 7, we consider the problem of pricing time on a single server in real time, for some long or infinite time horizon. In this model, the agents arrive online, each with a job to complete, and some value attained from completing the job. Job lengths and values of agents are drawn *i.i.d.* from an unknown distribution. Since agents are strategic they would like to maximize the difference between their value, and the price paid. In fact, we are never given access to a job's true parameters even after they are scheduled, and we must infer whatever we can from their behaviour. This can lead to complications as agents may lie about their requirements if it is profitable to them.

Our goal is to design a pricing scheme that maximizes the revenue for the scheduler while incentivizing truthful reporting from the agents. From the discussion above, prior work has often sought to maximize the sum of valuations of scheduled jobs. The problem of revenue-maximization has been more difficult. To better model the reality of server pricing, we restrict ourselves to *posted prices*, where the mechanism broadcasts (posts) a price for each job length, and the arriving jobs choose what they want to buy.

Posted price mechanisms (PPM) were introduced by [76] and have gained attention due to their simplicity, robustness to collusion, and their ease of implementation in practice. One of the first theoretical results concerning posted price mechanisms is an asymptotic comparison to classical single-parameter mechanisms [77]. They were later studied by [78] for the objective of revenue maximization, and further strengthened by [79] and [80]. [81] shows that sequential posted prices can $1/2$-approximate social welfare for XOS valuation functions — a class which generalizes submodular functions — if the price for an item is equal to the expected contribution of the item to the social welfare.

The result of Chapter 7 gives posted-price mechanisms for single-server scheduling which maximize expected revenue up to additive error, for buyers arriving online, with parameters of value, length and maximum delay drawn from an underlying distribution. The pricing problem is first modelled as a Markov decision process with an exponentially large action space. We then go on to show how the action space can be reduced under standard assumptions, to give an efficient algorithm. We then go on to show that these procedures are robust to approximate knowledge of the distribution, and give learning procedures.

Competitive Equilibria

Mechanism design has often been concerned with the study of *fair* but *efficient* allocations of goods to agents, where fairness has traditionally been measured with respect to envy-freeness, meaning no agent would prefer another's outcome to their own. The seminal work in envy-freeness deal with *cake cutting*, first introduced by [82], where a "cake" with non-homogeneous value is to be divided among n agents, who value the different portions of the cake differently.

The goal is to achieve some sense of fairness, while also achieve some economic efficiency: we should not be satisfied with assigning nothing to anyone. We should instead strive to extract as much utility for the agents as possible from the cake, while maintaining fairness. Fairness here can mean envy-freeness, where no agent prefers another agent's piece, or fair-share, where every agent values the portion they receive at least as much as a $\frac{1}{n}$ fraction of their value for the whole cake. Efficiency is often measured by Pareto-optimality: no alternate way of dividing the cake leaves everyone better off. See [83] for a survey on the rich literature that has developed around answering this question.

The notion of *competitive equilibrium with equal income* has emerged as capturing all of these features simultaneously, and it can be proved to almost always exist. Thus, it is one of the best mechanisms for this problem: it simultaneously guarantees envy-freeness, fair-share, and Pareto-optimality, and has constructive proofs of existence. Competitive equilibria were first introduced in [84] in the study of commodity markets. We seek to divide m commodities among n agents. Assume there is a total of 1 unit of each commodity, up to re-scaling quantities. Each agent i is allocated a vector $\boldsymbol{x}_i = (x_{i1}, \dots, x_{im})$, also known as a *bundle*, where x_{ij} is the total fraction of good j that is allocated to agent i. Agent i has a utility function $u_i : [0,1]^m \to \mathbb{R}$ on the space of bundles. Finally the goods have a price, where each unit of good j has price p_j, and the price of a bundle $\boldsymbol{x}$ is given by the inner product $\langle \boldsymbol{p}, \boldsymbol{x} \rangle$. It is assumed that each agent has a budget equal to 1, *i.e.* that incomes are all equal, and prices are re-scaled without loss of generality. A competitive equilibrium is a combination of an allocation $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, and a price vector $\boldsymbol{p}$, such that :

- Every agent spends less than their budget, normalized to 1: $\langle \boldsymbol{p}, \boldsymbol{x}_i \rangle \leq 1$ for all i.
- Every agent receives a utility-maximizing bundle, subject to the budget constraint.
- Every good is fully allocated, *i.e.* $\sum_{i=1}^n \boldsymbol{x}_i = \mathbf{1}$.
- The allocation, ignoring prices, is Pareto-optimal.

It was shown by [85, 86] that such an equilibria can be found by optimizing the product of the agents' utility when the utility functions are concave functions of x and 1-homogeneous. Formally,

$$u_i\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \;\geq\; \lambda u_i(\boldsymbol{x}) + (1-\lambda) u_i(\boldsymbol{y}) \qquad \text{and} \qquad u_i(a \cdot \boldsymbol{x}) = a \cdot u_i(\boldsymbol{x}) \; . \tag{1.11}$$

This allows us to conclude that the set of competitive equilibria with equal income (CEEI) form a convex set. Furthermore, since the utilities are log-concave, then such an equilbrium can be efficiently found by maximizing the logarithm of the product of utilities, as this is a concave objective over a convex feasible region. When utilities are linear, folklore results give strongly polynomial time algorithms for computing CEEI.

A recent work of [87] extends this result to the setting of *mixed manna*, where not all of the items are goods that the agents desire, but instead, some are "bads", such as chores that the agents are required to complete, and have negative utility for. In this setting, costs are allowed to be negative, as in payments for completing chores. In the special case where all agents value all of the items negatively, such as when all items are chores, they show that an allocation which is a local optimum for the product of the absolute values of the utilities satisfies the conditions of competitive equilibria, as long as nobody is assigned a zero allocation, as this would not be envy-free, unless it is the trivial (infeasible) allocation where all agents get zero.

In Chapter 8, we provide an efficient FPTAS to find such an *approximately-competitive* equilibrium in the negative utility setting. We give explicit algorithms when utilities are linear, and our results can be extended to the general utility setting with separation oracles and nearest-point-in-a-convex-set oracles.

Despite the breadth of knowledge in the positive-utility setting, where many (strongly) polynomial time algorithms are known [88, 89, 90], few algorithmic results are known in the negative utility setting. [87] show that in the negative-utilities setting, the space of equilibria might be disconnected, whereas in the positive setting, they form a connected, convex set, and would often be unique. When either the number of items or the number of agents is a constant, efficient methods exist to find exact equilibria in the mixed utilities setting [91, 92]. In general, a simplex-like method for finding competitive equilibria when utilities are separable, piecewise-linear, and concave was shown to exist [93]. The problem is known to be PPAD-hard to approximate in general, even when utilities are positive, but requiring non-homogeneous valuations [94]. In the negative-utility setting, for a more general "exchange" setting, a recent result showed PPAD-hardness of approximation for finding competitive equilibria even when (negative) utilities are linear [93].

We tackle the problem of finding a *approximately* competitive equilibrium, in the setting of negative utility items. Our algorithm seeks to find an extreme point for a concave minimization problem (the product of the absolute values of the negative utilities) in a convex region. Normally, extreme points can be found by gradient-following methods, even when they do not guarantee optimality. Unfortunately, if any agent were to be allocated a zero bundle, then we would lose envy-freeness, and the product of the utilities would be 0. Constraining that each agent is allocated a positive amount is an open constraint, and these are not well-handled by standard optimization techniques. In brief experiments conducted by the author, gradient descent methods which use log-barrier methods for the zero-allocation boundary tend to get stuck arbitrarily close to this boundary.

Instead of minimizing the product of absolute utility inside the convex region, we develop procedures to efficiently maximize the product *outside* the region. Since only local optima on the boundary are required, solutions to both problems coincide. We develop an iterative technique which alternatingly finds points on the boundary of the feasible region, and seeks to increase the product of negative utilities along supporting hyperplanes. We show that this procedure must efficiently converge to a point that approximately satisfies the KKT conditions in a multiplicative sense, and extend the results of [87] to show that such a point is approximately a competitive equilibrium, multiplicatively.

1.3 OUTLINE OF THESIS AND NOTATION

The first part of this thesis gives the details for our results on the smoothed analysis of Nash equilibrium computation. In Chapter 2, we prove that finding Nash equilibria in 2-player games will not admit smoothed efficient algorithms under standard complexity assumptions, even for constant-sized perturbation, unless PPAD admits quasi-polynomial algorithms. In Chapter 3, we prove that Nash equilibria in network coordination games can be found in smoothed polynomial and quasi-polynomial time, for complete and general game graphs, respectively. In Chapter 4, we introduce the notion of smoothness-preserving reductions, and give two examples of smoothness-preserving reductions to the local max cut problem, giving an alternate proof of one of the results of Chapter 3.

The second part of this thesis gives our decision theory and mechanism design results. In Chapter 5, we extend the classical *Pandora's box problem* of Weitzman [64], by adding a precedence constraint on the order of search, and give efficient optimal methods when precedence constraints are tree-like. For general constraints, we show hardness of approximation and give approximation algorithms. In Chapter 6, we ask if good binary classifiers can be found efficiently, when the objective function is not known to the algorithm, but instead, we are given a comparison oracle which can determine which of two classifiers is better, up to some noise.

In Chapter 7, we study the problem of pricing time on a single server to maximize revenue. We reduce the problem to solving a Markov decision process, and give natural assumptions under which the decision process can be optimally determined efficiently. Finally, in Chapter 8, we given an efficient algorithm to find a fair and efficient allocation of chores to agents, in a competitive equilibrium. This algorithm uses a recent characterization by [87] of competitive equilibria in the language of convex optimization.

1.3.1 Notation

In this thesis, matrices will be denoted by capital letters, *e.g.* M, A, and B, and their entries denoted by the double subscripts, *e.g.* $M_{i,j}$. Vectors are denoted by bold lowercase variables, with entries unbolded, *e.g.* $\boldsymbol{x} = (x_1, \ldots, x_n)$. It is common notation in game theory to denote by $\boldsymbol{x}_{-i}$ the vector $\boldsymbol{x}$ with its i-th entry excluded, and in an abuse of notation, we write $\boldsymbol{x} = (x_i, \boldsymbol{x}_{-i})$, to highlight the i-th entry. Random variables are denoted by capital letters, in both univariate and multivariate cases. The distinction will be made clear from context, *e.g.* $X \in \mathbb{R}$, or $Y = (Y_1, \ldots, Y_n)$. These will often be called X, Y, or Z.

The inner product between vectors may be denoted by the transpose-product $\boldsymbol{x}^\top \boldsymbol{y}$, or with angle-bracket notation $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. These refer to the same operation, and we use whichever is most clear in context. Inequality between vectors is taken to mean dominance, *i.e.* $\boldsymbol{x} \geq \boldsymbol{y}$ refers to the relation $x_i \geq y_i \ \forall i$. The vector $\mathbf{1}$ refers to the all-1's vector, and $\mathbf{0}$ refers to the all-0's vector. If specifically mentioned in context, they may refer to the all-1's or all-0's matrices, respectively, but this is not common. We commonly use $\boldsymbol{x} \geq \mathbf{0}$ to state that the entries of $\boldsymbol{x}$ are non-negative.

The signed halves of the real line are denoted with inequalities in the subscripts, *e.g.* $\mathbb{R}_{\geq 0}$ or $\mathbb{R}_{<0}$, to denote non-negative numbers and strictly negative numbers, respectively. The notation $[k]$ denotes either $\{0, 1, \ldots, k\}$ or $\{1, 2, \ldots, k\}$, and it will be specified if it is not clear from context.

1.3.2 Acknowledgements

The major technical contributions found in this thesis are based on joint work [58, 95, 96, 97, 98] with all of the co-authors I have had the pleasure of working with during my Ph.D. In alphabetical order, they are: Joshua Brakensiek, Federico Fusco, Gaurush Hiranandani, Samuel B. Hopkins, Sanmi Koyejo, Rucha Kulkarni, Philip Lazos, Stefano Leonardi, Yishay Mansour, Ruta Mehta, and Aviad Rubinstein.

CHAPTER 2: SMOOTHED COMPLEXITY OF 2-PLAYER NASH EQUILIBRIA[1]

This chapter studies the Smoothed Analysis of Nash Equilibrium computation in standard-form games, which have been defined in the introduction. We show that finding equilibria for games that have been perturbed is computationally equivalent to finding equilibria in worst-case games, even when the magnitude of perturbation is large relative to the payoff entries. We formally define the model here.

Definition 2.1 (X-SMOOTHED-NASH)**.** For a random variable X on $\mathbb{R}$ and problem size n, fix worst-case $n \times n$ matrices W_A, W_B with entries in $[-1, 1]$, and let N_A, N_B be $n \times n$ matrices whose entries are *i.i.d.* copies of X. X-SMOOTHED-NASH is the problem of computing, with probability at least $1 - \frac{1}{n}$, a Nash equilibrium of the game $(W_A + N_A, W_B + N_B)$.

This chapter gives the proof for the following result:

Theorem 2.1. There exists a universal constant $\epsilon > 0$, such that for any random variable X supported on $[-\epsilon, \epsilon]$, X-SMOOTHED-NASH is PPAD-hard under a randomized reduction.

For complexity-theoretic implications, it will be necessary that samples from a distribution approximating X be computable. This is formalized in the following assumption.

Assumption 2.2. We assume that there is a random variable X', which is jointly distributed with X, such that with probability $1 - 1/\operatorname{poly}(n)$, we have $|x - x'| < 1/\operatorname{poly}(n)$, and samples of X' can be generated by a randomized polynomial-time algorithm.

Assumption 2.2 holds for any natural smoothing distribution, such as truncated Gaussian noise or uniform noise. For most well-behaved distributions, it suffices to let X' be a rounding of X to nearby multiples of $1/\operatorname{poly}(n)$, with weight shifted to nearby gridpoints so that the probability mass function has polynomial description complexity, while maintaining the high-probability proximity of X and X'. A polynomial-sized description of the cumulative density function suffices to sample X', and if it must be included as advice to the input, we can only conclude that X-SMOOTHED-NASH is PPAD-hard under randomized reductions with polynomial-length advice.

In their 2006 survey on smoothed analysis, Spielman and Teng posed the challenge ([35], Open Question 11) of exploring the connections between smoothed complexity and hardness

[1] This chapter is based on collaboration with Joshua Brakensiek, Aviad Rubinstein, and Samuel B. Hopkins [97].

of approximation. Concretely, they considered the example of two-player Nash equilibrium subject to σ-bounded perturbations: Given a hard game $A, B \in [-1,1]^{n\times n}$, perturbing each entry independently gives rise to a new instance $\hat{A}, \hat{B} \in [-1-\sigma, 1+\sigma]^{n\times n}$; any Nash equilibrium of $\hat{A}, \hat{B}$ is an $O(\sigma)$-approximate-Nash equilibrium of the original game A, B. Hence, solving Nash equilibrium in the smoothed model is at least as hard as approximating Nash [35, Proposition 9.12].

When σ is inverse-polynomial in n, the results of [4] imply that finding approximate Nash equilibria and SMOOTHED-NASH are equally hard, namely, both are PPAD-hard problems. However, when $\sigma = \Omega(1)$, the approximation algorithm of [9] runs in quasi-polynomial time. Under the assumption that PPAD problems cannot be solved in randomized quasi-polynomial time, our result shows a divergence between the approximate problem and the smoothed problem in this regime, giving a negative answer to the conjecture of Spielman and Teng.

2.1 OVERVIEW

This section presents an overview of results and methods contained in this chapter. We begin by recalling the original theorem by [4] on the hardness of equilibrium computation in normal-form games.

Theorem 2.3 ([4])**.** For all $c > 0$, computing an n^{-c}-approximate Nash equilibrium of an $n \times n$ bimatrix game with entries bounded in $[0, 1]$ is PPAD-complete.

The reduction, presented in Section 2.5, will ultimately take a hard instance of Theorem 2.3 and transform it into a instance of X-SMOOTHED-NASH, for suitable distributions X supported on $[-\epsilon, \epsilon]$. The starting point of our reduction is the following simple idea: for any mixed strategies $(\boldsymbol{x}, \boldsymbol{y})$ which are *spread* (in a sense we make precise later) over a large number of actions, the noise from the smoothing averages out, and the payoffs to the two players will be approximately equal to their original payoffs. Therefore, if we start with an off-the-shelf PPAD-hard game (P, Q) and amplify it by simple repetition (formally, we tensor both P and Q with the all ones matrix $J := \mathbf{1}_{\ell\times\ell}$), the signal from P, Q will remain strong even with respect to "well-spread" strategies. This means that given a "well-spread" Nash equilibrium $\boldsymbol{x}, \boldsymbol{y}$ for a tensored, smoothed game $(P \otimes J + N_P, Q \otimes J + N_Q)$, we can recover a $1/\operatorname{poly}(n)$-approximate equilibrium for (P, Q).

There is one major problem with the reduction suggested above: solving X-SMOOTHED-NASH on this tensored game might not return a well-spread equilibrium $(\boldsymbol{x}, \boldsymbol{y})$. Our goal is therefore to to modify this construction to create a game where no Nash equilibrium has strategies concentrated on a small number of actions. Notice that small-support equilibria do not break this approach: they can be found efficiently by brute-force enumeration, so such games cannot be hard.

This begs the question, which games have no strategies concentrated on a small number of actions? At one extreme, if the entries of the payoff matrices are all sampled *i.i.d.* from any continuous distribution, a folklore result [25] states that the game has a pure equilibrium with probability approaching $1 - 1/e$, and therefore large, independently distributed, noise may lead to equilibria with small supports, which will not be hard to find.

In contrast, we observe that random zero-sum games tend to have only well spread equilibria [99, 100]. For example, they are exponentially unlikely to have a pure equilibrium: intuitively, if a pure strategy profile is exceptionally good for one player, it is likely exceptionally bad for the other. In the context of our proof approach, another advantage of random zero-sum games is that with respect to well-spread mixed strategies, they will also average out. That is, even if we add a random zero-sum game Z, we can still hope to recover a $1/\operatorname{poly}(n)$ Nash equilibrium for (P, Q) from a well-spread equilibrium for $(P \otimes J + Z + N_P, Q \otimes J - Z + N_Q)$. Our main technical task is to show that adding a random zero-sum game in this fashion produces a game with only well-spread Nash equilibria, even in the presence of the *i.i.d.* smoothing N_A, N_B.

Outline of the proof method. Our first step is to rule out small-support equilibra. In Section 2.3 we formalize the above intuition, showing that every equilibrium of a random zero-sum game has large supports, even when we add constant-magnitude perturbations. For technical reasons, our proof in this section works for random zero-sum games whose entries are drawn uniformly from discrete $\{-1, 1\}$.

Our second step is to obtain a robust version of no-small-support. Namely, building on the fact every equilibrium has large support, in Section 2.4 we prove that it must be well-spread (formally, the mixed strategies have small $|| \cdot ||_2$ norm). For technical reasons, our proof in this section works for random zero-sum games whose entries are drawn uniformly from continuous $[-1, 1]$. Fortunately, we can make both of proofs work simultaneously by taking the sum of a $\{-1, 1\}$ and a $[-1, 1]$ zero-sum games.

Combining these steps, our final construction of hard instance is given by:

$$\begin{array}{rccccc} A := & P \otimes J & + & Z_{\{-1,1\}} & + & Z_{[-1,1]} \\ B := & \underbrace{Q \otimes J}_{\textsf{PPAD}\text{-hard}} & - & \underbrace{Z_{\{-1,1\}}}_{\text{large support}} & - & \underbrace{Z_{[-1,1]}}_{\text{well-spread}} , \end{array} \tag{2.1}$$

where $Z_{\{-1,1\}}, Z_{[-1,1]}$ are random matrices with *i.i.d.* entries uniformly sampled from $\{-1, 1\}$

and $[-1,1]$ (respectively), and (P,Q) is a PPAD-hard bimatrix game, and J is an (appropriate-dimension) all-ones matrix.

In Section 2.5, we show that when the Nash equilibrium strategies are well-spread, the random zero-sum games and random perturbations average out. Thanks to the amplification, the signal from (P,Q) remains sufficiently strong. Thus, we can map any Nash equilibrium of (A,B) to a $1/\operatorname{poly}(n)$-approximate Nash equilibrium of (P,Q). By [4] this suffices to establish PPAD-hardness (under randomized reductions).

2.2 AN ANTI-CONCENTRATION LEMMA

This section provides the proof of a concentration lemma which will be essential for the following section, but does not provide much insight into the proof method itself.

We begin by formally defining the notation of the problem. Let $\epsilon > 0$ be a sufficiently small constant. Let X be any symmetric distribution on $[-\epsilon,\epsilon]$. Let n, b be positive integers such that b divides n, $b = n^{0.01}$, and n is sufficiently large. We divide $[n]$ into b *blocks* which we label $I_i := \{(i-1)\frac{n}{b}+1, (i-1)\frac{n}{b}+2, \ldots, i\cdot\frac{n}{b}\}$. We let $\ell := n/b = n^{0.99}$ denote the *block length*.

Let $P, Q \in \mathbb{R}^{b\times b}$ be payoff matrices. Let J_ℓ denote the $\ell\times\ell$ all 1's matrix. Let Z_0 be an $n\times n$ matrix whose entries are sampled *i.i.d.* from the Rademacher distribution (i.e., the uniform distribution on $\{-1,1\}$). Let Z_1 be an $n\times n$ matrix whose entries are sampled *i.i.d.* from the uniform distribution on $[-1,1]$. Let A_ϵ, B_ϵ be $n\times n$ matrices whose entries are *i.i.d.* sampled from X (all distributions independent).[2]

$$\begin{aligned} A &:= P\otimes J_\ell + Z_0 + Z_1 + A_\epsilon \\ B &:= Q\otimes J_\ell - Z_0 - Z_1 + B_\epsilon, \end{aligned} \tag{2.2}$$

where $P\otimes J_\ell$ denotes the $n\times n$ matrix, where every entries in block $I_i\times I_j$ is $P_{i,j}$.

Observe that a Nash equilibrium of (A,B) requires that

$$\begin{aligned} & \boldsymbol{x}^\top A\boldsymbol{y} \ge \boldsymbol{e}_i^\top A\boldsymbol{y} && \text{for all } i\in[n] \\ & \boldsymbol{x}^\top B\boldsymbol{y} \ge \boldsymbol{x}^\top B\boldsymbol{e}_j && \text{for all } j\in[n] \\ \implies\ & \boldsymbol{x}^\top(A+B)\boldsymbol{y} \ge \boldsymbol{e}_i^\top A\boldsymbol{y} + \boldsymbol{x}^\top B\boldsymbol{e}_j && \text{for all } i,j\in[n]. \end{aligned} \tag{2.3}$$

In the following sections, we will show that (2.3) cannot hold when the support of $(\boldsymbol{x},\boldsymbol{y})$ is

[2] The to meet the definition of X-SMOOTHED-NASH, which specifies that the hard game must have entries between $[-1,1]$, we can scale the construction (and thus X) by a factor of 3.

sufficiently small. To do that, we introduce a "benchmark" function, to which both the LHS and the maximum value of the RHS of (2.3) are comparable to. To define this benchmark, we begin by introducing a notion of *robust partition* of the strategy vectors. Consider $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ such that $\|\boldsymbol{x}\|_1 = 1$. Let $L = \lceil \log_2 n \rceil / 2^{100}$. Let $D = 2^{2^{500}}$. Let $E_1, \ldots, E_L$ be intervals such that $E_i = (D^{-i}, D^{-(i-1)}]$ for all $i < L$ and $E_L = [0, D^{-(L-1)}]$. Let $\boldsymbol{x} = \boldsymbol{x}^{(1)} + \cdots + \boldsymbol{x}^{(L)}$ such that

$$\boldsymbol{x}^{(i)}_j = \begin{cases} \boldsymbol{x}_j & \boldsymbol{x}_j \in E_i \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

We say that $\boldsymbol{x}^{(i)}$ is *sparse* if it has at most L nonzero coordinates; otherwise we say $\boldsymbol{x}^{(i)}$ is *dense.* Let $\boldsymbol{x}_{\text{sparse}}$ be the sum of the sparse $\boldsymbol{x}^{(i)}$'s and $\boldsymbol{x}_{\text{dense}}$ be the sum of the dense ones. Note that $\boldsymbol{x} = \boldsymbol{x}_{\text{sparse}} + \boldsymbol{x}_{\text{dense}}$. Now define the following quantity

$$\beta(\boldsymbol{x}) := \sqrt{\log n}\|\boldsymbol{x}_{\text{dense}}\|_2 + \|\boldsymbol{x}_{\text{sparse}}\|_1. \tag{2.5}$$

We call $\beta(\boldsymbol{x})$ the *benchmark* for $\boldsymbol{x}$. This quantity will appear in a number of concentration and anti-concentration inequalities. The goal of this section is to prove the following key anticoncentration inequality concerning this robust partition.

Lemma 2.1. Assume that X is the uniform distribution on $\{-1, 1\}$ (i.e., the Rademacher distribution). There exists a universal constant $c > 0$ with the following property: For all $\boldsymbol{x} \in \mathbb{R}^n$ such that $\|\boldsymbol{x}\|_1 = 1$, with probability at least $n^{-0.001}$ over $\boldsymbol{v} \sim X^n$

$$\langle \boldsymbol{v}, \boldsymbol{x} \rangle \geq c\beta(\boldsymbol{x}). \tag{2.6}$$

In our result, we need the following bound of Erdős [101].

Theorem 2.4 ([101], variant of [102])**.** Let $a_1, \ldots, a_n \geq 1$ be real numbers and $\epsilon_1, \ldots, \epsilon_n$ be Rademacher random variables (uniform distribution on $\{-1, 1\}$) then for all integers $k \geq 1$,

$$\Pr[a_1\epsilon_1 + \cdots a_n\epsilon_n \geq k - 1] \geq \Pr[\epsilon_1 + \cdots + \epsilon_n \geq k]. \tag{2.7}$$

Furthermore, the following binomial inequality will be useful:

Lemma 2.2 ([103])**.** For all k and n,

$$\binom{n}{k} \geq \frac{2^{nH(k/n)}}{\sqrt{8n}}, \tag{2.8}$$

where $H(\cdot)$ is the binary entropy function.

Note that when $k = \frac{n}{2}(1+\delta)$, then

$$\begin{aligned} H(k/n) &:= -\frac{1+\delta}{2}\log_2(\tfrac{1}{2}(1+\delta)) - \frac{1-\delta}{2}\log_2(\tfrac{1}{2}(1-\delta)) \\ &\geq 1 - \frac{1}{\ln 2}\left(\frac{1+\delta}{2}\cdot\delta + \frac{1-\delta}{2}\cdot(-\delta)\right) \\ &= 1 - (\log_2 e)\delta^2. \end{aligned} \tag{2.9}$$

Combining with the above inequality gives

$$\frac{1}{2^n}\binom{n}{k} \geq \frac{1}{\sqrt{8n}}e^{-n\delta^2}. \tag{2.10}$$

This allows us to show the following:

Claim 2.5. For all integers $n \geq k \geq 0$ with n sufficiently large

$$\frac{1}{2^n}\sum_{i=\frac{n+k}{2}}^{n}\binom{n}{i} \geq \frac{1}{10000}\exp\left(-\frac{10k^2}{n}\right). \tag{2.11}$$

Proof. Note that here, $\delta = \frac{k}{n}$. If $k \geq n - 2\sqrt{n}$, then

$$-10k^2/n \leq -10n + 40\sqrt{n} - 400 \leq -9n \tag{2.12}$$

for n sufficiently large. Note that the LHS of 2.11 is at least $2^{-n} > e^{-9n}$, and thus is at least the RHS.

On the other hand, if $k \leq n - 2\sqrt{n}$, then by Lemma 2.2, the sum of the first $\sqrt{n}$ terms is at least

$$\begin{aligned} \sqrt{n}\binom{n}{\frac{n+k}{2}+\sqrt{n}} &\geq \sqrt{n}\frac{1}{\sqrt{8n}}\exp\left(-n\cdot\left(\frac{k+2\sqrt{n}}{n}\right)^2\right) \\ &= \tfrac{1}{\sqrt{8}}\exp\left(-\frac{k^2+4k\sqrt{n}+4n}{n}\right) \\ &\geq \tfrac{1}{\sqrt{8}}\exp\left(-\frac{5k^2+5n}{n}\right) \\ &= \frac{1}{e^5\sqrt{8}}\exp\left(-\frac{5k^2}{n}\right), \end{aligned} \tag{2.13}$$

which implies the claim. *Q.E.D.*

With the above results, we can prove the lemma.

Proof of Lemma 2.1. Recall, we have defined $\boldsymbol{x}^{(i)}$ to be the vector $\boldsymbol{x}$ restricted to positions i such that $D^{-i} < x_i \leq D^{-(i-1)}$, for all $i < L$, or $0 \leq x_i \leq D^{-(L-1)}$ if $i = L$. Note that if we drop $\boldsymbol{x}^{(L)}$, β changes by at most $\sqrt{\log n}\|\boldsymbol{x}^{(L)}\|_1 \leq n\sqrt{\log n} \cdot n^{-2^{400}+1}$, a negligeably small term. Thus, we can without loss of generality assume that $\boldsymbol{x}^{(L)} = 0$.

Since $\beta(\boldsymbol{x}) = \|\boldsymbol{x}_{\text{sparse}}\|_1 + \sqrt{\log n}\|\boldsymbol{x}_{\text{dense}}\|_2$, we have for any $\boldsymbol{x}$, at least one of $\|\boldsymbol{x}_{\text{sparse}}\|_1$ or $\sqrt{\log n}\|\boldsymbol{x}_{\text{dense}}\|_2$ is at least $\frac{1}{2}\beta(\boldsymbol{x})$. Assume we know that with probability at least $2n^{-0.001}$, $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{sparse}}\rangle = \Omega(\|\boldsymbol{x}_{\text{sparse}}\|_1)$; and with probability at least $2n^{-0.001}$, $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{dense}}\rangle = \Omega(\sqrt{\log n}\|\boldsymbol{x}_{\text{dense}}\|_2)$. Then, we know with probability at least $n^{-0.001}$, one of $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{sparse}}\rangle$ and $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{dense}}\rangle$ is at least $\Omega(\beta(\boldsymbol{x}))$ and the other is at least 0 and thus their sum is at least $\beta(\boldsymbol{x})$. We split the remainder of the proof into two parts.

Part 1, $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{sparse}}\rangle = \Omega(\|\boldsymbol{x}_{\text{sparse}}\|_1)$. Let $\boldsymbol{x}'$ be the $2L$ largest coordinates of $\boldsymbol{x}_{\text{sparse}}$. Note that $\|\boldsymbol{x}'\|_1$ is at least D^2 times the sum of the next $2L$ largest coordinates of $\boldsymbol{x}_{\text{sparse}}$ and at least D^4 times the sum of the next $2L$ largest coordinates after that, etc. Thus, $\|\boldsymbol{x}'\|_1 \geq \frac{1}{2}\|\boldsymbol{x}_{\text{sparse}}\|$.

Now with probability $1/2^{2L}$, because $\boldsymbol{v}$ has *i.i.d.* Rachemacher entries, $\langle \boldsymbol{v}, \boldsymbol{x}'\rangle = \|\boldsymbol{x}'\|_1$, and with probability at least $1/2$, $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{sparse}} - \boldsymbol{x}'\rangle \geq 0$. Thus, with probability at least $1/2^{2L+1} \geq 2n^{-0.001}$, $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{sparse}}\rangle \geq \frac{1}{2}\|\boldsymbol{x}_{\text{sparse}}\|_1$.

Part 2, $\langle \boldsymbol{v}, \boldsymbol{x}_{\text{dense}}\rangle = \Omega(\sqrt{\log n}\|\boldsymbol{x}_{\text{dense}}\|_2)$. Since $\boldsymbol{x}_{\text{dense}} = \sum_{i\in F}\boldsymbol{x}^{(i)}$, we have that

$$\Pr\left[\langle \boldsymbol{v}, \boldsymbol{x}_{\text{dense}}\rangle \geq \frac{\sqrt{\log n}}{1000D}\|\boldsymbol{x}_{\text{dense}}\|_2\right] \geq \prod_{i\in F}\Pr\left[\langle \boldsymbol{v}, \boldsymbol{x}^{(i)}\rangle \geq \frac{\sqrt{\log n}\cdot\|\boldsymbol{x}^{(i)}\|_2^2}{1000D\|\boldsymbol{x}_{\text{dense}}\|_2}\right] \tag{2.14}$$

Consider $i \in F$, and let $m_i \geq L+1$ be the support size of $\boldsymbol{x}^{(i)}$. Since $\boldsymbol{x}_j^{(i)}D^i \geq 1$ for all j in the support of $\boldsymbol{x}^{(i)}$, we have by Theorem 2.4 and Claim 2.5, that for any integer $k \in [0, m_i]$

$$\Pr\left[\langle \boldsymbol{v}, \boldsymbol{x}^{(i)}\rangle \geq \frac{k}{D^i}\right] \geq \sum_{i=\frac{m_i+k}{2}+1}^{m_i}\binom{m_i}{i} \geq \frac{1}{10000}\exp\left(-10m_i\left(\frac{k+2}{m_i}\right)^2\right). \tag{2.15}$$

Observe that $\|\boldsymbol{x}^{(i)}\|_2 \leq \sqrt{m_i}\|\boldsymbol{x}^{(i)}\|_\infty \leq \sqrt{m_i}D^{-(i-1)}$. Thus,

$$\Pr\left[\langle \boldsymbol{v}, \boldsymbol{x}^{(i)}\rangle \geq \frac{k}{D\sqrt{m_i}}\|\boldsymbol{x}^{(i)}\|_2\right] \geq \frac{1}{10000}\exp\left(-10m_i\left(\frac{k+2}{m_i}\right)^2\right). \tag{2.16}$$

Let

$$k = \left\lceil \frac{1}{1000}\sqrt{m_i\log n}\cdot\frac{\|\boldsymbol{x}^{(i)}\|_2}{\|\boldsymbol{x}_{\text{dense}}\|_2}\right\rceil. \tag{2.17}$$

Then, note that

$$\frac{k+2}{m_i} \le \frac{3}{m_i} + \frac{1}{1000}\sqrt{\frac{\log n}{m_i}} \cdot \frac{\|\boldsymbol{x}^{(i)}\|_2}{\|\boldsymbol{x}_{\text{dense}}\|_2} \tag{2.18}$$

Thus, since $(a+b)^2 \le 2a^2 + 2b^2$,

$$-10m_i \cdot \left(\frac{k+2}{m_i}\right)^2 \ge -\frac{180}{m_i} - \frac{\log n}{5 \cdot 10^4} \cdot \frac{\|\boldsymbol{x}^{(i)}\|_2^2}{\|\boldsymbol{x}_{\text{dense}}\|_2^2}. \tag{2.19}$$

Therefore,

$$\Pr\left[\langle \boldsymbol{v}, \boldsymbol{x}^{(i)}\rangle \ge \frac{\sqrt{\log n}}{1000D} \cdot \frac{\|\boldsymbol{x}^{(i)}\|_2^2}{\|\boldsymbol{x}_{\text{dense}}\|_2}\right] \ge \frac{1}{10^4}\exp\left(-\frac{180}{m_i} - \frac{\log n}{5 \cdot 10^4} \cdot \frac{\|\boldsymbol{x}^{(i)}\|_2^2}{\|\boldsymbol{x}_{\text{dense}}\|_2^2}\right). \tag{2.20}$$

Applying Eq. 2.14, and noting that each $m_i \ge L \ge |F|$.

$$\begin{aligned}
\Pr\left[\langle \boldsymbol{v}, \boldsymbol{x}_{\text{dense}}\rangle \ge \frac{\sqrt{\log n}}{1000D}\|\boldsymbol{x}_{\text{dense}}\|_2\right] &\ge \frac{1}{10^{4L}}\exp\left(-\sum_{i\in F}\frac{180}{m_i} - \frac{\log n}{5\cdot 10^4}\right) \\
&= \frac{1}{10^{4L}e^{180}}n^{-10^{-5}} \\
&\ge n^{-2^{-90}}n^{-10^{-5}} \\
&\ge n^{-0.001},
\end{aligned} \tag{2.21}$$

For n sufficiently large. This concludes the proof. *Q.E.D.*

2.3 BOUNDING THE SUPPORT SIZE OF EQUILIBRIA

In this section, we use the previous lemma and concepts to show that for any equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ on our smoothed input, the strategies will be supported on a large subset of the coordinates. Using this, the next section will bound $\|\boldsymbol{x}\|_2$, $\|\boldsymbol{y}\|_2$, which is the key ingredient in the proof of Theorem 2.1. The main result of this section is the following lemma.

Lemma 2.3. With probability $1 - n^{-3}$, for every Nash equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ of (A, B), we have that $|\text{supp}(\boldsymbol{x})| = |\text{supp}(\boldsymbol{y})| > n^{0.96}$.

We prove this result using methods partially inspired by [100]. The following concentration bound will be of use. For any distribution X, we let $X^{n\times n}$ denote the distribution of $n \times n$ matrices with entries *i.i.d.* samples from X. Recall, the *benchmark* function β introduced in equation (2.5):

$$\beta(\boldsymbol{x}) := \sqrt{\log n}\|\boldsymbol{x}_{\text{dense}}\|_2 + \|\boldsymbol{x}_{\text{sparse}}\|_1. \tag{2.22}$$

Claim 2.6. Let X be any distribution on $[-1,1]$. There exists a universal constant $C > 0$ such that for all $n \geq 0$, with probability $1 - 1/n^4$ over $M \sim X^{n \times n}$, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ such that $\|\boldsymbol{x}\|_1 = \|\boldsymbol{y}\|_1 = 1$, we have that

$$|\boldsymbol{x}^\top M \boldsymbol{y}| \leq C \cdot (\beta(\boldsymbol{x}) + \beta(\boldsymbol{y})). \tag{2.23}$$

To prove the claim, we will first need the following lemma. Recall, a random variable $X \in \mathbb{R}$ is said to be *sub-gaussian with variance proxy* $s^2 > 0$ if for all $t > 0$, $\mathbb{E} \exp(tX) \leq \exp(s^2 t^2 / 2)$.

Lemma 2.4. Let A be an $n \times n$ matrix with independent subgaussian entries with variance proxy at most 1. For all $u > 0$, with probability at least $1 - \exp(-u^2)$, all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ with $\|\boldsymbol{x}\|_2 = \|\boldsymbol{y}\|_2 = 1$ have

$$\boldsymbol{x}^\top A \boldsymbol{y} \leq O(\sqrt{\log n} + u)(\|\boldsymbol{x}\|_1 + \|\boldsymbol{y}\|_1)\,. \tag{2.24}$$

As a corollary, with the same probability, all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ with $\|\boldsymbol{x}\|_1, \|\boldsymbol{y}\|_1 \leq 1$ have

$$\boldsymbol{x}^\top A \boldsymbol{y} \leq O(\sqrt{\log n} + u)(\|\boldsymbol{x}\|_2 + \|\boldsymbol{y}\|_2)\,. \tag{2.25}$$

To prove this lemma, we rely on the following powerful comparison inequality of Talagrand.

Theorem 2.7 (Talagrand's comparison inequality, high-probability version. [104], Exercise 8.6.5)**.** Suppose that $\{X_{\boldsymbol{s}}\}_{\boldsymbol{s} \in S}$ is a collection of $\mathbb{R}$-valued random variables, indexed by some $S \subseteq \mathbb{R}^n$, $0 \notin S$. Suppose that for all $\boldsymbol{s}, \boldsymbol{t} \in S$, $X_{\boldsymbol{s}} - X_{\boldsymbol{t}}$ is subgaussian with variance proxy at most $\|\boldsymbol{s} - \boldsymbol{t}\|_2$. There is a universal constant $C > 0$ such that for all $u > 0$, with probability at least $1 - \exp(-u^2)$,

$$\sup_{\boldsymbol{s} \in S} X_{\boldsymbol{s}} \leq C \left(\mathop{\mathbb{E}}_{\boldsymbol{g} \sim \mathcal{N}(0,I)} \sup_{\boldsymbol{s} \in S} \langle \boldsymbol{g}, \boldsymbol{s} \rangle + u \cdot \sup_{\boldsymbol{s} \in S} \|\boldsymbol{s}\|_2 \right)\,. \tag{2.26}$$

Proof of Lemma 2.4. Consider for each $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ the random variable $\boldsymbol{x}^\top A \boldsymbol{y} / (\|\boldsymbol{x}\|_1 + \|\boldsymbol{y}\|_1)$. Since the entries of A are subgaussian with variance proxy 1, there is a universal $C > 0$ such that $\langle U, A \rangle$ is subgaussian with variance proxy $C\|U\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm, for any $n \times n$ matrix U. Hence, for $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}' \in \mathbb{R}^n$,

$$\frac{\boldsymbol{x}^\top A \boldsymbol{y}}{\|\boldsymbol{x}\|_1 + \|\boldsymbol{y}\|_1} - \frac{(\boldsymbol{x}')^\top A \boldsymbol{y}'}{\|\boldsymbol{x}'\|_1 + \|\boldsymbol{y}'\|_1} \tag{2.27}$$

is subgaussian with variance proxy $C\|\boldsymbol{x}\boldsymbol{y}^\top/(\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1)-(\boldsymbol{x}')(\boldsymbol{y}')^\top/(\|\boldsymbol{x}'\|_1+\|\boldsymbol{y}'\|_1)\|_F^2$. We claim that

$$\left\|\frac{\boldsymbol{x}\boldsymbol{y}^\top}{\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1}-\frac{(\boldsymbol{x}')(\boldsymbol{y}')^\top}{\|\boldsymbol{x}'\|_1+\|\boldsymbol{y}'\|_1}\right\|_F^2 \le \left\|\frac{(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1}-\frac{(\boldsymbol{x}',\boldsymbol{y}')}{\|\boldsymbol{x}'\|_1+\|\boldsymbol{y}'\|_1}\right\|_2^2, \tag{2.28}$$

where $(\boldsymbol{x},\boldsymbol{y})$ denotes the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}$ to a $2n$-length vector. To see this, recalling that $\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=\|\boldsymbol{x}'\|_2=\|\boldsymbol{y}'\|_2=1$, let $m=\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1$ and $m'=\|\boldsymbol{x}'\|_1+\|\boldsymbol{y}'\|_1$ and expand both sides, it is equivalent to prove

$$\frac{m^2+(m')^2-2m(m')\langle\boldsymbol{x},\boldsymbol{x}'\rangle\langle\boldsymbol{y},\boldsymbol{y}'\rangle}{m^2(m')^2} \le \frac{2m^2+2(m')^2-2m(m')\langle\boldsymbol{x},\boldsymbol{x}'\rangle-2m(m')\langle\boldsymbol{y},\boldsymbol{y}'\rangle}{m^2(m')^2}. \tag{2.29}$$

This is equivalent to

$$\frac{1}{m^2}+\frac{1}{(m')^2}-2\frac{\langle\boldsymbol{x},\boldsymbol{x}'\rangle+\langle\boldsymbol{y},\boldsymbol{y}'\rangle-\langle\boldsymbol{x},\boldsymbol{x}'\rangle\langle\boldsymbol{y}',\boldsymbol{y}\rangle}{mm'} \ge 0\,. \tag{2.30}$$

Dividing by $2/mm'$ and using $1/m^2+1/(m')^2\ge 2/mm'$, it is enough to show

$$1-\langle\boldsymbol{x},\boldsymbol{x}'\rangle-\langle\boldsymbol{y},\boldsymbol{y}'\rangle-\langle\boldsymbol{x},\boldsymbol{x}'\rangle\langle\boldsymbol{y}',\boldsymbol{y}\rangle\ge 0\,. \tag{2.31}$$

This factors as $(1-\langle\boldsymbol{x},\boldsymbol{x}'\rangle)(1-\langle\boldsymbol{y},\boldsymbol{y}'\rangle)\ge 0$ since we assumed $\boldsymbol{x},\boldsymbol{x}',\boldsymbol{y},\boldsymbol{y}'$ were unit vectors.

Now we can apply Theorem 2.7 to see that with probability at least $1-\exp(-u^2)$,

$$\sup_{\substack{\boldsymbol{x},\boldsymbol{y}\\ \|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1}}\frac{\boldsymbol{x}^\top A\boldsymbol{y}}{\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1}\le C\left(\mathop{\mathbb{E}}_{\boldsymbol{g}\sim\mathcal{N}(0,I)}\sup_{\substack{\boldsymbol{x},\boldsymbol{y}\\ \|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1}}\frac{\langle(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{g}\rangle}{\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1}+u\right) \tag{2.32}$$

where C is a universal constant, $\boldsymbol{g}$ is a length $2n$ Gaussian vector with independent coordinates, and we have used that $\|(\boldsymbol{x},\boldsymbol{y})\|_2\le\|(\boldsymbol{x},\boldsymbol{y})\|_1=\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1$. To finish the argument, observe that

$$\mathop{\mathbb{E}}_{g\sim\mathcal{N}(0,I)}\sup_{\substack{\boldsymbol{x},\boldsymbol{y}\\ \|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1}}\frac{\langle(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{g}\rangle}{\|\boldsymbol{x}\|_1+\|\boldsymbol{y}\|_1}=\mathop{\mathbb{E}}_{\boldsymbol{g}\sim\mathcal{N}(0,I)}\|\boldsymbol{g}\|_\infty\le O(\sqrt{\log n})\,. \tag{2.33}$$

Finally, to prove the corollary, note that we just showed that with probability at least $1-\exp(-u^2)$, all $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n$ with $\|\boldsymbol{x}\|_1=\|\boldsymbol{y}\|_1=1$ have $\boldsymbol{x}^\top A\boldsymbol{y}/\|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2\le O(\sqrt{\log n}+u)\cdot(1/\|\boldsymbol{x}\|_2+1/\|\boldsymbol{y}\|_2)$. Multiplying by $\|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2$ implies the corollary. *Q.E.D.*

With this lemma in hand, we can prove the claim.

Proof of Claim 2.6. Apply Lemma 2.4 to M with $u = \sqrt{3 \log n}$. Then, there is a universal constant C' such that with probability $1 - 1/n^3$, for all $\boldsymbol{x}, \boldsymbol{y}$ with ℓ_1 norm 1,

$$|\boldsymbol{x}_{\text{dense}}^\top M \boldsymbol{y}_{\text{dense}}| \leq C' \sqrt{\log n} (\|\boldsymbol{x}_{\text{dense}}\|_2 + \|\boldsymbol{y}_{\text{dense}}\|_2). \tag{2.34}$$

Thus, since the entries of M have absolute value at most 1,

$$\begin{aligned} |\boldsymbol{x}^\top M \boldsymbol{y}| &\leq |\boldsymbol{x}^\top M \boldsymbol{y}_{\text{sparse}}| + |\boldsymbol{x}_{\text{sparse}}^\top M \boldsymbol{y}_{\text{dense}}| + |\boldsymbol{x}_{\text{dense}}^\top M \boldsymbol{y}_{\text{dense}}| \\ &\leq \|\boldsymbol{y}_{\text{sparse}}\|_1 + \|\boldsymbol{x}_{\text{sparse}}\|_1 + C' \sqrt{\log n} (\|\boldsymbol{x}_{\text{dense}}\|_2 + \|\boldsymbol{y}_{\text{dense}}\|_2) \\ &\leq \max(C', 1)(\beta(\boldsymbol{x}) + \beta(\boldsymbol{y})). \end{aligned} \tag{2.35}$$

Thus, we can set $C = \max(C', 1)$. *Q.E.D.*

These results will allow us to prove Lemma 2.3. We present first the following facts about equilibria in random games.

Proposition 2.8. With probability 1, for nonempty $S, T \subset [n]$ there is at most one Nash equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ of (A, B) with $S = \text{supp}(x)$ and $T = \text{supp}(y)$. Further, with probability 1 all such equilibria have $|S| = |T|$.

Proof. Fix nonempty $S, T \subset [n]$. Fix $i_0 \in S$. Assume without loss of generality that $|S| \geq |T|$. Denote A^{00} as the sub-matrix of A restricted to rows indexed by S and columns indexed by T. For any equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ with supports S and T, we have that $\boldsymbol{x}^\top A \boldsymbol{y} = \boldsymbol{e}_i^\top A^{00} \boldsymbol{y}$ for all $i \in S$, when treating $\boldsymbol{x}$ and $\boldsymbol{y}$ as $|S|$- and $|T|$-dimensional vectors, respectively. Therefore,

$$(\boldsymbol{e}_i - \boldsymbol{e}_{i_0})^\top A^{00} \boldsymbol{y} = 0 \text{ for all } i \in S \setminus \{i_0\}. \tag{2.36}$$

Since all the entries of A^{00} are drawn independently from a continuous distribution, the null space of the linear system (2.36) has dimension $\max(|T| - |S| + 1, 0) \leq 1$ with probability 1. Since $\boldsymbol{y} \neq 0$ the null space must have dimension exactly 1. Thus, $|T| - |S| + 1 \geq 1$, which implies that $|S| = |T|$ and the solution $\boldsymbol{y}$ is unique, as there can be at most one vector in a 1-dimensional subspace with coordinates summing to one. By a similar argument $\boldsymbol{x}$ is also unique.

Since there are only finitely many choices of S and T, with probability 1 the proposition holds for all Nash equilibria simultaneously. *Q.E.D.*

With probability 1, all equilibria of A and B will have the same support size, and further, for every pair of possible supports $S \subset [n]$ and $T \subset [n]$ there is at most one equilibrium. We let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ denote the probability distributions of strategies in this equilibrium. We can now prove Lemma 2.3

Proof of Lemma 2.3. Assume (which happens with probability $1 - n^{-4}$) that the event described in Claim 2.6 occurs for $M = \frac{1}{2\epsilon}(A_\epsilon + B_\epsilon)$. Fix $S, T \subset [n]$ with $|S|, |T| < \ell/10$. We seek to show that with probability at most $2^{-\ell}$, S and T can be the support of a Nash equilibrium. By Proposition 2.8, we can assume that $|S| = |T|$.

Also by Proposition 2.8, with probability 1, there is at most one equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ on the game (A^{00}, B^{00}) with full support, where we have defined A^{00} to be A whose rows and columns have been restricted to S and T, respectively. Note that $\boldsymbol{x}$ and $\boldsymbol{y}$, if they exist, are independent of the entries of A and B outside of $S \times T$. As mentioned in Section 2.2, in order for the equilibrium to extend, inequality (2.3) must hold:

$$\boldsymbol{x}^\top (A + B)\boldsymbol{y} \geq \boldsymbol{e}_i^\top A\boldsymbol{y} + \boldsymbol{x}^\top B\boldsymbol{e}_j \qquad \text{for all } i, j \in [n]. \tag{2.37}$$

Say that $i \in [n] \setminus S$ is *S-good* if $\boldsymbol{e}_i^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y} > c\beta(\boldsymbol{y})$. By Lemma 2.1, we know that $\boldsymbol{e}_i^\top Z_1 \boldsymbol{y} > c\beta(\boldsymbol{y})$ with probability at least $n^{-0.001}$. Independently, we have that $\boldsymbol{e}_i^\top (Z_0 + A_\epsilon)\boldsymbol{y} \geq 0$ with probability at least $1/2$ (since $Z_0 + A_\epsilon$ is a mean-zero matrix distribution). Therefore, both this event happens with probability at least $n^{-0.001}/2 \geq n^{-0.01}$.

Likewise, say that $j \in [n] \setminus T$ is *T-good* if $\boldsymbol{x}^\top (-Z_0 - Z_1 - B_\epsilon)\boldsymbol{e}_j > c\beta(\boldsymbol{x})$. By the same argument, this also happens with probability at least $n^{-0.01}$. Furthermore, the S-good events and T-good events are independent of each other because each event is based on a disjoint subset of entriesZ from Z_0 and Z_1.

Since $\boldsymbol{x}$ and $\boldsymbol{y}$ are probability distributions, there exists $i_0 \in S$ and $j_0 \in T$ such that $\boldsymbol{e}_{i_0}^\top (P \otimes J_\ell)\boldsymbol{y} \geq \boldsymbol{x}^\top (P \otimes J_\ell)\boldsymbol{y}$ and $\boldsymbol{x}^\top (Q \otimes J_\ell)\boldsymbol{e}_{j_0} \geq \boldsymbol{x}^\top (Q \otimes J_\ell)\boldsymbol{y}$. Let $i', j' \in [b]$ be the indices of the blocks such that $i_0 \in I_{i'}$ and $j_0 \in I_{j'}$. Since we assume that $|S|, |T| \leq \ell/10$, we have that $I_{i'} \setminus S$ and $I_{j'} \setminus T$ both have size at least $9\ell/10$.

Now, for any good $i \in I_{i'} \setminus S$ and good $j \in I_{j'} \setminus T$, we have

$$\begin{aligned}
\boldsymbol{x}^\top (A + B)\boldsymbol{y} &= \boldsymbol{x}^\top (P \otimes J_\ell)\boldsymbol{y} + \boldsymbol{x}^\top (Q \otimes J_\ell)\boldsymbol{y} + \boldsymbol{x}^\top (A_\epsilon + B_\epsilon)\boldsymbol{y} \\
&\leq \boldsymbol{e}_{i_0}^\top (P \otimes J_\ell)\boldsymbol{y} + \boldsymbol{x}^\top (Q \otimes J_\ell)\boldsymbol{e}_{j_0} + 2C\epsilon(\beta(\boldsymbol{x}) + \beta(\boldsymbol{y})) \\
&= \boldsymbol{e}_i^\top (P \otimes J_\ell)\boldsymbol{y} + \boldsymbol{x}^\top (Q \otimes J_\ell)\boldsymbol{e}_j + 2C\epsilon(\beta(\boldsymbol{x}) + \beta(\boldsymbol{y})) \\
&< \boldsymbol{e}_i^\top (P \otimes J_\ell)\boldsymbol{y} + \boldsymbol{x}^\top (Q \otimes J_\ell)\boldsymbol{e}_j + c(\beta(\boldsymbol{x}) + \beta(\boldsymbol{y})) && (\epsilon < 2c/C) \\
&< \boldsymbol{e}_i^\top (P \otimes J_\ell + Z_0 + Z_1 + A_\epsilon)\boldsymbol{y} + \boldsymbol{x}^\top (Q \otimes J_\ell - Z_0 - Z_1 + B_\epsilon)\boldsymbol{e}_j \\
&= \boldsymbol{e}_i^\top A\boldsymbol{y} + \boldsymbol{x}^\top B\boldsymbol{y}, && (2.38)
\end{aligned}$$

which contradicts Ineq. 2.3. Thus, there must either be no good $i \in I_{i'} \setminus S$ or there is no good $j \in I_{j'} \setminus T$. This happens with probability at most

$$2\left(1 - n^{-0.01}\right)^{9\ell/10} \leq 2e^{-(0.9)\ell/n^{0.01}} \leq e^{-n^{0.97}}, \tag{2.39}$$

where we use in the last inequality that n is sufficiently large. The number of pairs S, T with support at most $n^{0.96}$ is at most

$$\binom{n}{\leq n^{0.96}}^2 \leq n^{2n^{0.96}}. \tag{2.40}$$

Note that for n sufficiently large, $n^{2n^{0.96}} e^{-n^{0.97}} \ll n^{-4}$. Thus, all equilibria have support size greater than $n^{0.96}$ with probability at least $1 - 2n^{-4} \geq 1 - n^{-3}$. *Q.E.D.*

2.4 BOUNDING THE NORM OF EQUILIBRIA

Towards building the norm bound needed to show Theorem 2.1, the previous section showed that with high probability, any equilibrium must have polynomially large support. This section completes the proof of the norm bound. We wish to show the following.

Lemma 2.5. With probability $1 - 20n^{-3}$, for every Nash equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ of (A, B), we have that $\|\boldsymbol{x}\|_2, \|\boldsymbol{y}\|_2 \leq n^{-0.2}$.

We must, however, begin this section with a few technical results. We will need the following theorem, which is derived from the fact that the VC-dimension of the set of halfspaces in $\mathbb{R}^d$ has VC-dimension at most $d+1$ – that is, the VC-dimension of $\{\boldsymbol{x} \mapsto 1[\langle \boldsymbol{x}, \boldsymbol{v}\rangle + t \geq 0] : \boldsymbol{v} \in \mathbb{R}^d, t \in R\}$ is at most $d+1$. (See e.g. [105], Example 4.21.)

Theorem 2.9 (Multivariate Glivenko-Cantelli)**.** Let X be a random vector in $\mathbb{R}^d$ and let $X_1, \ldots, X_n$ be independent copies of X. For all $\delta \in [0,1]$, with probability $1 - \delta$,

$$\sup_{\boldsymbol{v} \in \mathbb{R}^d, t \in \mathbb{R}} \left| \frac{1}{n} \sum_{i=1}^{n} 1[\langle X_i, \boldsymbol{v}\rangle \geq t] - \Pr_X(\langle X, \boldsymbol{v}\rangle \geq t) \right| \leq O\left(\sqrt{\frac{d}{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right). \tag{2.41}$$

We also need the following Littlewood-Offord-type theorem.

Theorem 2.10 ([106], Theorem 1.2)**.** Let $X_1, \ldots, X_n$ be real-valued independent random variables with densities almost everywhere bounded by K. Let $a_1, \ldots, a_n \in \mathbb{R}$ with $\sum_{i \leq n} a_i^2 = 1$. Then the density of $\sum_{i \leq n} a_i X_i$ is bounded by $\sqrt{2}K$ almost everywhere.

The following lemma, which we obtain as a corollary of these two theorems, allows us to argue that the entries of a product of a random matrix with a fixed vector are relatively spread out.

Lemma 2.6. Let n, d be positive integers. Let X be an $\mathbb{R}$-valued random variable with density bounded by K. Let $\boldsymbol{g}_1 \ldots, \boldsymbol{g}_n$ be independent random vectors in $\mathbb{R}^d$ whose coordinates are independent copies of X. With probability $1-\delta$, for all unit vectors $\boldsymbol{v} \in \mathbb{R}^d$ and all intervals $[a,b] \subset \mathbb{R}$,

$$\frac{1}{n}\sum_{i=1}^{n} 1[\langle \boldsymbol{g}_i, \boldsymbol{v}\rangle \in [a,b]] \leq \sqrt{2}K|a-b| + O\left(\sqrt{\frac{d}{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right). \tag{2.42}$$

Proof. By Theorem 2.9, with probability at least $1-\delta$, the CDFs of $\langle \boldsymbol{g}, \boldsymbol{v}\rangle$ and the empirical distribution of $\langle \boldsymbol{g}_i, \boldsymbol{v}\rangle$ have distance at most $O\left(\sqrt{\frac{d}{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right)$, for all $\boldsymbol{v} \in \mathbb{R}^d$. So it suffices to show that for every unit $\boldsymbol{v} \in \mathbb{R}^d$, $\Pr_{\boldsymbol{g}}(\langle \boldsymbol{g}, \boldsymbol{v}\rangle \in [a,b]) \leq \sqrt{2}K|a-b|$. This follows immediately from Theorem 2.10. *Q.E.D.*

Finally, this lemma allows us to prove the following claim.

Claim 2.11. Let X be a distribution on $[-1,1]$ whose probability density is at most 100 everywhere. Let $M \sim X^{n\times n}$. With probability $1-n^{-4}$, for every $S, T \subset [n]$ with $|S| \geq n^{0.95}$ and $|T| \leq n^{0.85}$, there exists disjoint $S_1, S_2 \subset S$ of size at least $n^{0.94}$ each such that for all unit vectors $\boldsymbol{y} \in \mathbb{R}^n$ with support in T there exists $r \in \mathbb{R}$ such that

$$\begin{aligned} &\boldsymbol{e}_{i_1}^{\top} M\boldsymbol{y} \geq r + n^{-0.07} && \text{for all } i_1 \in S_1 \\ &\boldsymbol{e}_{i_2}^{\top} M\boldsymbol{y} \leq r && \text{for all } i_2 \in S_2. \end{aligned}$$

Proof. For every $T \subset [n]$ of size at most $n^{0.85}$, apply Lemma 2.6 to the rows of M restricted to the columns of T (so $d = |T| \leq n^{0.85}$) with $\delta = e^{-n^{0.86}}$. Thus, with probability $1 - e^{-n^{0.86}}$, for every unit vector $y \in \mathbb{R}^d$ supported on T and every interval $[a,b]$ of length $n^{-0.06}/10$, the number of $i \in [n]$ such that $\boldsymbol{e}_i^{\top} M\boldsymbol{y} \in [a,b]$ is at most

$$n\left[100\sqrt{2}|a-b| + O\left(\sqrt{\frac{d}{n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right)\right] = O(n^{0.94}) \tag{2.43}$$

choices of $i \in [n]$ for which $\boldsymbol{e}_i^{\top} A\boldsymbol{y}$ falls in that interval. Since $|S| \geq n^{0.95}$, this implies there exist $r \in \mathbb{R}$, and disjoint $S_1, S_2 \subset S$ of size at least $n^{0.94}$ such that

$$\begin{aligned} &\boldsymbol{e}_{i_1}^\top M\boldsymbol{y} \geq r + \frac{n^{-0.06}}{10} \geq r + n^{-0.07} && \text{for all } i_1 \in S_1 \\ &\boldsymbol{e}_{i_2}^\top M\boldsymbol{y} \leq r && \text{for all } i_2 \in S_2. \end{aligned} \tag{2.44}$$

Taking the union bound over all choices of T we get this all happens with probability at most

$$1 - \binom{n}{\leq n^{0.85}} e^{-n^{0.86}} \geq 1 - e^{-n^{0.85}} \geq 1 - n^{-4}. \text{ } Q.E.D. \tag{2.45}$$

We can now prove Lemma 2.5.

Proof of Lemma 2.5. With probability $1 - n^{-3}$, by Lemma 2.3, for every equilibrium $(\boldsymbol{x}, \boldsymbol{y})$ of (A, B) with support S and T, respectively, we have that $|S| = |T| \geq n^{0.96}$. Since there are $n^{0.01}$ blocks. By the pigeonhole principle there exists $i_0, j_0 \in [b]$ such that $|S \cap I_{i_0}|, |T \cap I_{j_0}| \geq n^{0.95}$.

With probability $1 - 2n^{-4}$, Claim 2.11 holds with for both $M = \frac{1}{2+\epsilon}(Z_0 + Z_1 + A_\epsilon)$ and $M = \frac{1}{2+\epsilon}(-Z_0 - Z_1 + B_\epsilon)$. Further, with probability at least $1 - 2n^{-3}$, Lemma 2.4 holds for $M = \frac{1}{2+\epsilon}(Z_0 + Z_1 + A_\epsilon)$ and $M = \frac{1}{2+\epsilon}(-Z_0 - Z_1 + B_\epsilon)$ with $u = \sqrt{3}\log n$.

We seek to show that any large-support equilibrium also has small ℓ_2 norm. Assume for sake of contradiction (and without loss of generality) that $\|\boldsymbol{y}\|_2 \geq n^{-0.2}$. Let $S' = S \cap I_{i_0}$ and T' be the set of coordinates of $\boldsymbol{y}$ which are greater than $n^{-0.85}$. Clearly $|T'| \leq n^{0.85}$. Let $\boldsymbol{y}_{T'}$ be the coordinates of $\boldsymbol{y}$ supported on T' and $\bar{\boldsymbol{y}}_{T'}$ be the remaining coordinates. Observe that

$$\|\bar{\boldsymbol{y}}_{T'}\|_2^2 \leq n \cdot (n^{-0.85})^2 = n^{-0.7} \leq \frac{\|\boldsymbol{y}\|_2^2}{2} \tag{2.46}$$

$$\|\boldsymbol{y}_{T'}\|_2^2 = \|\boldsymbol{y}\|_2^2 - \|\bar{\boldsymbol{y}}_{T'}\|_2^2 \geq \frac{\|\boldsymbol{y}\|_2^2}{2}. \tag{2.47}$$

Applying Claim 2.11 for $M = \frac{1}{2+\epsilon}(Z_0+Z_1+A_\epsilon)$ and the sets S', T' and the vector $\boldsymbol{y'} := \frac{\boldsymbol{y}_{T'}}{\|\boldsymbol{y}_{T'}\|_2}$, there exists $S_1', S_2' \in S'$ and $r \in \mathbb{R}$ such that (scaling by $2 + \epsilon \geq 1$)

$$\begin{aligned} &\boldsymbol{e}_{i_1}^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y'} \geq r + n^{-0.07} && \text{for all } i_1 \in S_1' \\ &\boldsymbol{e}_{i_2}^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y'} \leq r && \text{for all } i_2 \in S_2'. \end{aligned} \tag{2.48}$$

Thus,

$$\begin{aligned} \boldsymbol{u}_{S_1'}^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y'} &\geq r + n^{-0.07} \\ \boldsymbol{u}_{S_2'}^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y'} &\leq r \\ \implies (\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'})^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y'} &\geq n^{-0.07}. \end{aligned} \tag{2.49}$$

Applying (2.47),

$$(\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'})^\top (Z_0 + Z_1 + A_\epsilon)\boldsymbol{y}_{T'} \geq n^{-0.07}\|\boldsymbol{y}\|_2/2 \geq n^{-0.28}. \tag{2.50}$$

Since Lemma 2.4 holds for $M = \frac{1}{2+\epsilon}(Z_0 + Z_1 + A_\epsilon)$, we have that

$$\begin{aligned} (\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'})^\top (Z_0 + Z_1 + A_\epsilon)\bar{\boldsymbol{y}}_{T'} &\geq -(2+\epsilon)C'\sqrt{\log n}(\|\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'}\|_2 + \|\bar{\boldsymbol{y}}_{T'}\|_2) \\ &\geq -n^{0.01}\max(\sqrt{2}n^{-0.94/2}, n^{-0.7/2}) \\ &\geq -n^{-0.34}. \end{aligned} \tag{2.51}$$

Therefore, since $\boldsymbol{y} = \boldsymbol{y}_{T'} + \bar{\boldsymbol{y}}_{T'}$

$$(\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'})^\top (Z_0 + A_\epsilon)\boldsymbol{y} \geq n^{-0.28} - n^{-0.34} \geq n^{-0.29}. \tag{2.52}$$

Since S_1' and S_2' are subsets of the same block, we have that $\boldsymbol{u}_{S_1'}(P \otimes J_\ell) = \boldsymbol{u}_{S_2'}(P \otimes J_\ell)$. Therefore,

$$(\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'})^\top A\boldsymbol{y} \geq n^{-0.29}. \tag{2.53}$$

But, since S_1' and S_2' are subsets of the support of $\boldsymbol{x}$, we know that

$$(\boldsymbol{u}_{S_1'} - \boldsymbol{u}_{S_2'})^\top A\boldsymbol{y} = 0, \tag{2.54}$$

thus we have a contradiction. Therefore, $\|\boldsymbol{y}\|_2 \leq n^{-0.2}$. By a similar argument (also with probability $1 - 5n^{-3}$, $\|\boldsymbol{x}\|_2 \leq n^{-0.2}$, as desired. By the union bound, the total probability of success is at least $1 - 20n^{-3} \geq 1 - n^{-2}$. *Q.E.D.*

2.5 THE REDUCTION, AND PROOF OF MAIN THEOREM

First, we show in Section 2.5.1 the reduction in the case that the noise distribution X is symmetric, i.e., the probability of sampling a and $-a$ is identical for all $a > 0$. We then show in Section 2.5.2 a slight modification which works for any distribution X.

Recall, we have defined

$$\begin{aligned} A &:= P \otimes J + Z_{\{-1,1\}} + Z_{[-1,1]} + A_\epsilon \\ B &:= Q \otimes J - Z_{\{-1,1\}} - Z_{[-1,1]} + B_\epsilon, \end{aligned} \tag{2.55}$$

where $Z_{\{-1,1\}}, Z_{[-1,1]}$ are random matrices with *i.i.d.* entries uniformly sampled from $\{-1,1\}$ and $[-1,1]$ (respectively), and (P,Q) is a $n^{0.01} \times n^{0.01}$ PPAD-hard bimatrix game, and J is an $n^{0.99} \times n^{0.99}$ all-ones matrix. A_ϵ and B_ϵ denote the *i.i.d.* noise added in the setting of X-SMOOTHED-NASH.

2.5.1 The symmetric case

We seek to show that equilibria of the reduced game (A,B) can be used to efficient produce approximate equilibria to the game (P,Q), which we have assumed is hard to approximate. Let $(\boldsymbol{x},\boldsymbol{y})$ be an equilibrium of (A,B). In Section 2.4, we showed that, with high probability, $\|\boldsymbol{x}\|_2, \|\boldsymbol{y}\|_2 \le n^{-0.2}$, even when ϵ is a constant. Note that $b = n^{0.01}$ is the dimension of the input game (P,Q). Define $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ to be distributions over $[b]$ such that for all $i \in [n]$

$$\hat{x}_i = \sum_{i' \in I_i} x_{i'}, \qquad \hat{y}_i = \sum_{i' \in I_i} y_{i'}. \tag{2.56}$$

Theorem 2.12. With probability $1 - n^{-2}$, we have that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is a b^{-19}-approximate equilibrium of (P,Q).

Proof. We claim that $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ is an $b^{-19} = n^{-0.19}$-approximate equilibrium of (P,Q) with high probability. Assume not, without loss of generality, Alice would benefit from deviating from $\hat{\boldsymbol{x}}$. That is, there exists $i \in [b]$ such that

$$\hat{\boldsymbol{x}}^\top P \hat{\boldsymbol{y}} \le \boldsymbol{e}_i^\top P \hat{\boldsymbol{y}} - b^{-19}. \tag{2.57}$$

Define $\boldsymbol{u}_S$ to be the uniform probability vector on support S, then, the above is equivalent to

$$\boldsymbol{x}^\top (P \otimes J_\ell) \boldsymbol{y} \le \boldsymbol{u}_{I_i}^\top (P \otimes J_\ell) \boldsymbol{y} - b^{-19}. \tag{2.58}$$

By Lemma 2.4, we may assume that the concentration inequality holds for $\frac{1}{2+\epsilon}(Z_0 + Z_1 + A_\epsilon)$, then we know that

$$|\boldsymbol{x}^\top(Z_0 + A_\epsilon)\boldsymbol{y}| \le O(\sqrt{\log n}\; n^{-0.2}) \tag{2.59}$$

$$|\boldsymbol{u}_{I_i}^\top(Z_0 + A_\epsilon)\boldsymbol{y}| \le O(\sqrt{\log n}\; n^{-0.2}) \tag{2.60}$$

Combining Eqs. 2.58, 2.59, and 2.60 we get

$$\boldsymbol{x}^\top A\boldsymbol{y} \le \boldsymbol{u}_{I_i}^\top A\boldsymbol{y} - b^{-19} + O(\sqrt{\log n}\; n^{-0.2}) < \boldsymbol{u}_{I_i}^\top A\boldsymbol{y}. \tag{2.61}$$

since $b = n^{0.01}$. This contradicts that $(\boldsymbol{x}, \boldsymbol{y})$ is a Nash equilibrium of (A, B).

By a similar argument, Bob does not wish to deviate with high probability. Therefore, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is a b^{-19}-approximate Nash equilibrium of (P, Q). *Q.E.D.*

Since finding a b^{-19}-approximate Nash equilibrium is PPAD-hard [4] when P and Q have constant sized entries, finding the smoothed equilibrium of (A, B) is PPAD-hard. Since the proofs of Sections 2.3 and 2.4 hold when X is supported on $[-\epsilon, \epsilon]$ for $\epsilon > 0$ constant, this is an instance of X-SMOOTHED NASH, and therefore concludes the proof of Theorem 2.1 when X is a symmetric distribution.

2.5.2 General X

Let X be any distribution supported on $[-\epsilon/2, \epsilon/2]$. Let $Y := X - X'$ be the distribution on $[-\epsilon, \epsilon]$ which takes two *i.i.d.* samples from X and subtracts them. Note that Y is a symmetric distribution, so by the previous section we have that Y-SMOOTHED NASH is hard. In particular, it is hard to find an equilibrium from the distribution

$$\begin{aligned} A &:= P \otimes J_\ell + Z_0 + Z_1 + A_Y \\ B &:= Q \otimes J_\ell - Z_0 - Z_1 + B_Y, \end{aligned} \tag{2.62}$$

where A_Y and B_Y are matrix whose entries are *i.i.d.* samples from Y. We can rewrite $A_Y = A_X - A'_X$ and $B_Y = B_X - B'_X$, where A_X, A'_X, B_X, B'_X are all *i.i.d.* matrix samples from X. Thus, the distribution can be rewritten as

$$\begin{aligned} A &:= (P \otimes J_\ell + Z_0 + Z_1 - A'_X) + A_X \\ B &:= (Q \otimes J_\ell - Z_0 - Z_1 - B'_X) + B_X, \end{aligned} \tag{2.63}$$

This is an instance of X-SMOOTHED NASH, and we conclude Theorem 2.1 for arbitrary X, losing a factor 2 on ϵ.

CHAPTER 3: SMOOTHED EFFICIENT ALGORITHMS FOR NETWORK COORDINATION GAMES[1]

We consider here the smoothed analysis of the problem of finding a (pure) Nash equilibrium in *network coordination games*, which were introduced in Section 1.1.5. Recall, a network coordination game is an n-player game, represented by a graph $G = (V, E)$, where the vertices are the n players, and the edges represent 2-player coordination games, and are labelled by the $k \times k$ payoff matrices. Here k and n are used rather than n and m to not confuse notation with standard graph quantities. The players choose only one strategy simultaneously for all incident game-edges, and earn the sum of their rewards. Formally,

$$u_i(\boldsymbol{\sigma}) = \sum_{e=ij \in E} (A_e)_{\sigma_i,\sigma_j} \tag{3.1}$$

Recall, from Section 1.1.5, that network coordination games admit a *potential function*, which is the sum of the edge payoffs. Namely,

$$\Phi(\boldsymbol{\sigma}) = \sum_{e=ij \in E} (A_e)_{\sigma_i,\sigma_j} = \tfrac{1}{2} \sum_{in \in V} u_i(\boldsymbol{\sigma}) \tag{3.2}$$

Thus, the problem of finding equilibria in network coordination games lies in PLS, and was in fact shown to be PLS-complete [57]. In this chapter, we consider the *smoothed* setting, where the entries of the payoff matrices are independently distributed random variables with bounded density, *i.e.* with density functions $f : [-1, 1] \to [0, \phi]$ for some density bound $\phi \geq \frac{1}{2}$.

Related Work. In the following section, we formally define the smoothed Local Max Cut problem in Definition 4.2: given a weighted graph G, whose edges weights are independently distributed random variables with density bounded by ϕ, we wish to find a cut of G which cannot be improved by moving one node across the cut. This smoothed problem was first studied by [31], who showed that with high probability, any local-search algorithm which only search in directions of strict improvement, must terminate in at most $\phi n^{O(\log n)}$. This bound was later improved to to $\phi n^{O(\sqrt{\log n})}$ [107]. In the special case of complete graphs, it was shown that the local search procedure must terminate in $poly(\phi, n)$ steps [32, 108].

We begin this chapter by highlighting a proof framework that is common to many of these smoothed analysis results, proceed to outline the results shown and give an overview of the proof method, and then finally give the proof details.

[1] This chapter, along with the next, is based on collaboration with Rucha Kulkarni and Ruta Mehta [58].

3.1 A COMMON FRAMEWORK

The analyses of all Local Max Cut papers introduced above [31, 32, 107, 108], as well as our result, all follow a common framework, and this framework will also be central to the next chapter. We introduce it here as it is of independent interest, and will be relevant in introducing part of our results. The main idea is to apply the anti-concentration bound below to show slow improvements are unlikely in the smoothed setting, leading to a fast convergence with high probability.

Lemma 3.1 ([109])**.** Let $M \in \mathbb{Z}^{n\times n}$ be a matrix with rank $0 \le r \le n$, and let X be a random vector whose entries are independently distributed random variables with density at most ϕ, supported on $[-1, 1]$. Formally, the i-th entry of X has density function $f_i : [-1, 1] \to [0, \phi]$. Then for any $\epsilon > 0$, and $\boldsymbol{a} \in \mathbb{R}$,

$$\Pr[MX - \boldsymbol{a} \in (0, \epsilon)^n] \;\le\; (\phi\epsilon)^r \tag{3.3}$$

We will present a proof in the following section. Note that, like the potential function for Network coordination games (1.7), the total value of a cut in a graph is also a linear combination of the input values, *i.e.* the edge weights. For consistent notation, we will denote a cut in the graph $G = (V, E)$ as $\boldsymbol{\sigma} : V \to \{0, 1\}$, and the cut value as

$$\Phi_G(\boldsymbol{\sigma}) := \textstyle\sum_{ij\in E} w_{ij} \cdot \mathbb{I}[\sigma_i \ne \sigma_j] \tag{3.4}$$

Therefore, if we denote as $S = (\boldsymbol{\sigma}^0,\, \boldsymbol{\sigma}^1,\, \ldots,\, \boldsymbol{\sigma}^T)$ the sequence of cuts observed in any local-search procedure, then the changes in the cut value throughout the local search can be given as an integer-coefficient linear transformation of the edge weights. Formally, if $\boldsymbol{w} \in \mathbb{R}^{|E|}$ is the vector of edge weights, then there exists $M_S \in \{-1,\, 0,\, +1\}^{T\times|E|}$ such that

$$\Phi(\boldsymbol{\sigma}^t - \boldsymbol{\sigma}^{t-1}) = (M_S\boldsymbol{w})_t \qquad \text{for all } t = 1,\, 2,\, \ldots,\, T \tag{3.5}$$

For any *fixed* sequence local search sequence, if $(M_S\boldsymbol{w})_t \le 0$ for any t, then it cannot be the result of a strictly improving search, and if $(M_S\boldsymbol{w})_t \ge \epsilon$ for any t, then the total improvement due to S must be at least ϵ. Therefore, the event that every strictly improving search sequence of length T makes an improvement at least ϵ, is a subset of $\{M_S\boldsymbol{w} \notin (0, \epsilon)^T\}$ for all sequences S.[2] By applying Lemma 3.1, we can immediately bound the probability of this event by $(\phi\epsilon)^{\mathrm{rank}(M_S)}$.

[2] This analysis was tightened in [108] to $M_S(\boldsymbol{w}) \notin \{\boldsymbol{w} \in (0, \epsilon)^T \,|\, \sum_i w_i \le \epsilon\}$, since it suffices to bound the sum of terms, rather than the individual terms. This substitution adds an extra factor of $1/r!$ to the probability bound, which tightens the running time analyses.

The Common Framework. The core idea to the common framework is an observation that is at the beginning of all of the results included herein. Namely, we begin by first observing that there is a potential function whose improvements can be seen as a linear map of the (smoothed, independently distributed) input values, and then go on to apply this lemma. If we can choose ϵ small enough, and work hard enough to bound the rank of M_S, we will then be able to take a union bound over all sequences S, and show that with high probability, for every long-enough sequence, either it is not a legal local-search sequence, or it makes improvement at least ϵ, and therefore we cannot make a large number of local improvement moves.

If there are N moves to choose from at each local improvement step, then there will be N^T fixed sequences of length T. If we can bound $\text{rank}(M_S) \geq r_0$ for every legal sequence S of length T, then the probability of success will be lower-bounded by $1 - N^T(\phi, \epsilon)^{r_0}$. If the potential function has a bounded range of $\Phi \in [-M, M]$, then we can conclude that, with probability at least $1 - N^T(\phi, \epsilon)^{r_0}$, the local search procedure will terminate in at most $2TM/\epsilon$ steps, as any sequence of T steps is making an improvement of at least ϵ.

Note that any sequence of moves S which re-visits the same configuration twice is trivially not a legal local-improvement sequence, and so we do not need to take these into account in the computation of r_0. This is essential for eliminating trivially low-rank M_S matrices.

For example, for the case of the Local Max Cut problem, we have $N = n$, and $M = \binom{n}{2}$, since we have assumed that the edge weights lie in $[-1, 1]$. Therefore, with probability $1 - n^T(\phi\epsilon)^{r_0}$, any strictly-improving local search procedure must terminate in at most Tn^2/ϵ steps. Therefore, if we can show that for $T = n$, $r_0 > n/c$ — as is shown in the result of [32] — then setting $\epsilon = n^{-c}/\phi$ allows us to conclude that, with high probability, any local search procedure will terminate in at most ϕn^{c+3} steps.

The main technical result therefore reduces to bounding this r_0. Unfortunately, such a naïve application of the union bound is often too loose, and further work is needed to refine the probability bound.

3.1.1 An Anti-Concentration Bound

We prove here Lemma 3.1, which will be key to the analyses of this chapter and the following. It was originally presented in [109], but it is central to the framework above, and so we present it and its proof for completeness.

We first present a proof for the following statement, which will allow us to conclude the above lemma, and will be of independent interest. The proof here follows closely with a proof given in [31].

Proposition 3.1. Let $X \in \mathbb{R}^d$ be a random vector such that the joint probability on any $a \leq d$ coordinates of X is upper-bounded by ϕ^a at all points, and let $M \in \mathbb{R}^{\ell \times d}$ be full-rank, with entries which are multiples of η, for $\ell \leq d$. Then the random variable $Y := MX$ also has bounded joint density $f_Y(y) \leq (\phi/\eta)^\ell$ for all $y \in \mathbb{R}^d$.

Proof, following [31]. Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ be the standard basis vectors of $\mathbb{R}^d$, and let $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell$ denote the (linearly independent) rows of M. Without loss of generality, $\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell, \boldsymbol{e}_{\ell+1} \ \ldots, \boldsymbol{e}_d\}$ is a basis for $\mathbb{R}^d$. Let $\widetilde{M}$ be the matrix whose rows are given by this basis. Let $\boldsymbol{x} \in \mathbb{R}^\ell$, and define $C_\epsilon(\boldsymbol{x}) := [x_1, x_1 + \epsilon] \times \cdots \times [x_\ell, x_\ell + \epsilon]$, a rectangular region in $\mathbb{R}^\ell$, and $\widetilde{C}_\epsilon(\boldsymbol{x}) := C_\epsilon(\boldsymbol{x}) \times \mathbb{R}^{d-\ell}$. We have $\Pr[MX \in C_\epsilon(\boldsymbol{x})] = \Pr[\widetilde{M}X \in \widetilde{C}_\epsilon(\boldsymbol{x})] = \Pr[X \in \widetilde{M}^{-1}\widetilde{C}_\epsilon(\boldsymbol{x})]$.

Observe that $\widetilde{M}^{-1}$ is the identity on the coordinates $d-\ell, \ldots, d$, since $\widetilde{M}$ is as well. Thus, we have that $\widetilde{M}^{-1}\widetilde{C}_\epsilon(\boldsymbol{x}) = C' \times \mathbb{R}^{d-\ell}$ for some region C' of volume at most ϵ^ℓ/η^ℓ. Therefore,

$$\begin{aligned}\Pr[X \in \widetilde{M}^{-1}\widetilde{C}_\epsilon(\boldsymbol{x})] &= \int_{C'} \mathrm{d}f_{X_1,\ldots,X_\ell} \cdot \int_{\mathbb{R}^{d-\ell}} \mathrm{d}f_{X_{\ell+1},\ldots,X_d} \\ &\leq \phi^\ell \cdot (\epsilon/\eta)^\ell \cdot 1 \qquad (3.6)\end{aligned}$$

Where the first integral bound comes from our assumption on the joint densities of collections of entries of X, and the second is simply integrating a probability density over the whole domain. Now taking the limit

$$\lim_{\epsilon \to 0} \frac{\Pr[MX \in C_\epsilon(\boldsymbol{x})]}{\mathrm{vol}(C_\epsilon(\boldsymbol{x}))} \leq \frac{\phi^\ell \epsilon^\ell \eta^{-\ell}}{\epsilon^\ell} = (\phi/\eta)^\ell \qquad (3.7)$$

which gives our desired bound on the density. *Q.E.D.*

The above proof in fact allows us to conclude the lemma, as follows:

Proof of Lemma 3.1. Since the entries of X are independently distributed, and have density bounded by ϕ, then any sub-collection of a many entries has joint density bounded by ϕ^a. Let M be as in the statement of the lemma, and assume without loss of generality that its first r rows are linearly independent, letting $\hat{M}$ be the restriction of M to its first r rows. Node that

$$\Pr[MX - \boldsymbol{a} \in (0, \epsilon)^n] \ \leq \ \Pr\left[(\hat{M}X)_i \in (a_i, a_i + \epsilon) \quad \forall\, 1 \leq i \leq r\right] \qquad (3.8)$$

By Proposition 3.1, setting $\eta = 1$, $d = n$, and $\ell = r$, with the above observation on the joint densities, and setting $\hat{M}$ to be the matrix in the proposition's statement, we have that the random variable $\hat{M}X$ has density upper-bounded by ϕ^r. Since the Cartesian product of the $(a_i, a_i + \epsilon)$ intervals has volume ϵ^r, the total probability mass of $\hat{M}X$ in this region is therefore upper-bounded by $(\phi\epsilon)^r$, as desired. *Q.E.D.*

3.2 RESULTS AND RELATED WORK

Recall, the *better-response algorithm* (or BRA) consists of repeatedly choosing an arbitrary player who can unilaterally improve their payoff, and making the improvement, until no such improvement is possible. The BRA is guaranteed to find a pure Nash equilibrium, but may take exponentially many improvement steps [110]. Following the Common Framework outlined in Section 3.1, and extending the lemmas of [31, 32], we study the BRA for finding Nash equilibria in smoothed network coordination games. We show that when the payoff values are independently sampled from distributions with density bounded by ϕ, the runtime will be polynomial in ϕ and the input size with probability $1 - 1/\operatorname{poly}(n)$. One may interpret ϕ as the inverse of the minimum allowed perturbation. Formally, we show the following.

Theorem 3.2. Let $G = (V, E)$ be a game graph for an instance of NetCoordNash, with $k \times k$ payoff matrices, whose entries are independently distributed, continuous, random variables, with densities $f_{u,v,i,j} : [-1, 1] \to [0, \phi]$. Let $n := |V|$. If G is a complete graph, then with probability $1 - (nk)^{-3}$, all valid executions of the BRA will converge to a PNE in at most $(nk\phi)^{O(k)}$ steps, and the expected maximum number of steps of any valid execution is polynomial in ϕ^k and n^k.

If G is arbitrary, all valid executions of the BRA, from all starting points, will converge to a PNE in at most $\phi \cdot (nk)^{O(k \log(nk))}$ steps with probability $1 - (nk)^{-2}$ over the payoff entries. Furthermore, the expected maximum number of steps of any valid execution is polynomial in ϕ and $n^{k \log(nk)}$.

An outline of the proof is given in Section 3.3, and the missing details in the rest of the Chapter. Observe that in the theorem statement, the randomness is only on the values in the payoff matrices, and not on the choice of BR moves. These results hold if the BR moves are chosen *adversarially* in response payoffs. We distinguish the two halves of the theorem statement in the following remark:

Remark 3.1. The distinction between complete graphs with all-zero edges, and incomplete graphs, is important: in the smoothed setting, all present edges must have random payoffs, which cannot be point masses at 0. Therefore, when the graph is complete, with probability 1, any two players share a nontrivial game-edge. Conversely, when the graph is incomplete, the players do not share a game edge, and will never be influenced by each others' decisions.

The polynomial running time requires the graph to be complete so that *all* parameters can be perturbed. This seems to be unavoidable as all known results on *polynomial* smoothed complexity so far, *e.g.,* linear-programming [26], local-max-cut[32], etc., require this.

The above performance guarantees are only (quasi-)polynomial in the input size for k fixed. We leave it as an open problem to improve this. This can be achieved either by showing that local-max-bisection has polynomial smoothed complexity (see below), or by directly tightening the bounds in the proof presented in this chapter.

3.2.1 Related Work

The works most related to ours are [31] and [32], who first analyzed the smoothed complexity of local-max-cut, and [108] who refined the analysis of the latter. As discussed earlier local-max-cut is a special case of NetCoordNash, therefore techniques of the former do not directly apply. Independently, [108] also obtained smoothed polynomial algorithms for local-max-3-cut on complete graphs, and quasi-polynomial algorithms in general for local-max-k-cut with constant k. Local-max-k-cut naturally reduces to NetCoordNash with $k \times k$ payoff matrices, however we note that our result does not subsume theirs as the reduction is not *smoothness preserving.*

Beyond-worst case complexity of NE computation. For Nash equilibrium (NE) computation, the smoothed complexity of two-player games is known to not lie in P unless RP = PPAD [111], which follows from the hardness of (1/poly)-additive approximation. On the contrary, for most PLS-complete problems, the natural local-search algorithm often finds an additive approximate solution efficiently. There is always a “potential function” that the algorithm improves in each step. Intuitively, until an approximate solution is reached, the algorithm will improve the associated potential function significantly in every local-search step.

Towards average case analysis, Bárány, Vempala, and Vetta [25] showed that a game picked uniformly at random has a NE with support size 2 for both the players with high probability. The average case complexity of a random potential game was shown to be polynomial in the number of players and strategies by Durand and Gaujal [112].

3.3 OVERVIEW AND NOTATION

In this section, we give the high-level structure for the proof of Theorem 3.2. The remaining sections of the chapter provide the missing details. Recall that a profile $\boldsymbol{\sigma}$ is a PNE if and only if it is a local maximum of the potential Φ defined in equation (3.2), and note that Φ may only take values in the interval $[-n^2, n^2]$, since the payoffs are in $[-1, 1]$. Therefore it suffices to show that with high probability, Φ will increase significantly in *every* linear-length sequence of BR moves. This is formalized in the following theorem.

Theorem 3.3. Let $G = (V, E)$ be a game graph, with random payoff vector A, and $\boldsymbol{\sigma}^0 \in [k]^n$ be an arbitrary strategy profile. With probability $1 - (nk)^{-2}$ over the values of A, all BR sequences of length at least $2nk$, initiated at any choice of $\boldsymbol{\sigma}^0$, must have at least one step in which the potential increases by $\epsilon = \phi^{-1}(2n^2k^3)^{-20k\log(nk)}$. If G is a complete graph, then with probability $1 - (nk)^{-3}$, all BR sequences of length at least $2nk$, will have at least one step increasing by $\epsilon' = (20\phi^2 n^3 k^3)^{-4k-4}$.

This theorem, along with the above observations, implies that the BRA must terminate in $\phi(nk)^{O(k\log(nk))}$ steps with probability $1-(nk)^{-2}$, and if the graph is complete, in $(\phi nk)^{O(k)}$ steps with probability $1-(nk)^{-3}$. The results in expectation follow from the high-probability results, as proved in Section 3.6.3. Therefore, Theorem 3.2 follows from Theorem 3.3.

Following the common framework outlined in Section 3.1, we will begin by expressing the increase in potential as a linear combination of the payoff values, and reduce the proof of Theorem 3.3 to the application of Lemma 3.1 and a union bound.

Each step of the BRA consists of some player, u, deviating from their previously chosen strategy to a new $\sigma \in [k]$, which we will denote as the (player,strategy) pair (u, σ). Thus, an execution of the BRA is fully specified by a sequence of pairs $S = (u_1, \sigma_1), (u_2, \sigma_2), \ldots$, along with an initial strategy vector $\boldsymbol{\sigma}^0 \in [k]^n$. The strategy profile at time t is given by $\boldsymbol{\sigma}^t := (\sigma_t, \boldsymbol{\sigma}^{t-1}_{-u_t})$. We introduce next the *potential-change matrix* for a BR sequence, which allows us to control the value of $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$ as a function of the payoff values.

Definition 3.1. For any fixed BR sequence S of length ℓ, we define the set of vectors $L(S, \boldsymbol{\sigma}^0) = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_\ell\}$, where $\boldsymbol{\lambda}_t \in \{-1, 0, 1\}^{(|E| \times k^2)}$, for all t. The entries of $\boldsymbol{\lambda}_t$ are indexed by indices of payoff matrix entries, denoted $((v,i)(w,j))$. The values of its entries are chosen as follows:

$$\boldsymbol{\lambda}_t((v,i)(w,j)) = \begin{cases} 1 & \text{if:} \quad u_t \in \{v,w\} \quad \text{and} \quad \sigma_v^t = i \quad \text{and} \quad \sigma_w^t = j. \\ -1 & \text{if:} \quad u_t \in \{v,w\} \quad \text{and} \quad \sigma_v^{t-1} = i \quad \text{and} \quad \sigma_w^{t-1} = j. \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

That is, every entry signifies if the corresponding payoff value gets added to the total payoff (+1), removed (−1), or has its status unchanged (0). We term this set of vectors, or equivalently the matrix whose columns consist of the $\boldsymbol{\lambda}_t$'s, as the *potential-change matrix* of a sequence.

The arguments $S, \boldsymbol{\sigma}^0$ will be omitted if they are clear from context. Observe $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1}) = \langle \boldsymbol{\lambda}_t, A \rangle$, where A is the vector of payoff values, so the vector $L(S, \boldsymbol{\sigma}^0) \cdot A$ represents the sequence of changes in Φ along an execution of the BRA. Theorem 3.3 is then equivalent to bounding the probability of $LA \notin [0, \epsilon]^\ell$ for all sequences of length $\ell \geq 2nk$. We will then apply Lemma 3.1, as outlined in the common framework. Thus, our goal is to lower-bound the rank of $L(S, \boldsymbol{\sigma}^0)$, for a worst-case choice of S and $\boldsymbol{\sigma}^0$, and taking a union bound over the choice of S and $\boldsymbol{\sigma}^0$. The right rank bound would imply Theorem 3.3. The task at hand then is to get the largest possible rank bounds and tight union bounds to get good running time overall. We introduce here some parameters.

Definition 3.2 (Active, Inactive, Repeating, and Non-Repeating players.)**.** Let S be a BR sequence, then player u is said to be *active* if they appear in the sequence, and otherwise, the player is termed *inactive*. An active player u is said to be *repeating* if there exists some strategy i such that (u, i) appears at least twice in S, or if (u, σ_u^0) appears in S at all. An active player which is not repeating is said to be *non-repeating*. We introduce the following notation:

Figure 3.1: Relevant variables for the analysis in this chapter.

$p(S)$	number of active players in S,	$d(S)$	number of distinct (u, i) moves in S,
$p_1(S)$	# non-repeating players in S,	$d_1(S)$	distinct moves by non-rep. players,
$p_2(S)$	number of repeating players in S,	$q_0(S)$	number of distinct (u, σ_u^0) moves

Observe that $p = p_1 + p_2$, $k \cdot p \geq d \geq p$, $k \cdot p_1 \geq d_1 \geq p_1$, and $q_0 \leq p_2$. We will often use the quantity $d(S) - q_0(S)$, which is the number of "new" strategies played by the players.

3.3.1 Union Bounds and Rank Bounds

As discussed above, we wish to take a union bound over all the possible sequences S of size, say, ℓ. Naïvely, we have $k^n (nk)^\ell$ choices of sequence of length ℓ and initial strategy profile. However, if $p(S) \ll n$, the rank bound cannot exceed $d(S) \leq k \cdot p(S)$ in our model, which does not match the union bound. To fix this, we will modify the matrix L in two different ways.

The two modified matrices will be relevant in the remaining analysis for the cases $p_1(S) \geq p_2(S)$ and $p_2(S) \geq p_1(S)$, respectively. This case analysis is similar to the proofs in [32, 108]. Unlike these two papers, we need both constructions to reach both bounds, whereas each of these papers used one of the two constructions for both bounds.

Control by rounding. The first matrix construction builds on the work of [32]. While the construction works for arbitrary graphs, the rank bounds hold only for complete graphs. Observe that if $V_0 \subset V$ is the set of inactive players, and $u \in V$ is active, then for i fixed, all $((u,i)(v,\sigma_v^0))$ rows of L for $v \in V_0$ are identical, modulo flipping a row's signs. This is because v's strategy never changes. Therefore, in the inner product $\langle \boldsymbol{\lambda}_t, A \rangle$, these $((u,i),(v,\sigma_v^0))$ terms are added or subtracted together, and we may simply take a union bound on an approximation of this value, instead of controlling for strategy choices. This idea is formalized in Section 3.4.

For $p(S)$ fixed, there are at most $(nk)^\ell$ choices of the BR sequence, $k^{p(S)}$ choices of initial strategy profiles for the active players, and $d(S) - q_0(S)$ different "buckets" of the payoffs with $4n/\epsilon$ choices for the approximate value. Thus, we have a union bound of size $k^{p(S)}(nk)^\ell(4n/\epsilon)^{d(S)-q_0(S)}$. In Section 3.4, we show that, *whenever the graph is complete*, $L(S,\boldsymbol{\sigma}^0)$ has rank at least $d(S)-q_0(S)+d_1(S)/2$, after we consider a "bucketing" operation.

To bound the probability of success for all BR sequences, we restrict our attention to *critical subsequences*, as used in [32, 108]. These are maximal (up to inclusion) continuous subsequences S' satisfying $\ell(S') \geq 2(d(S') - q_0(S'))$, formally defined in Section 3.4. As we will show, these must exist, and satisfy $\ell(S') = 2(d(S') - q_0(S'))$, which by definition is at most $2kp(S)$. For a fixed choice of p, p_1, p_2, d, d_1, q_0, we bound the probability of *any* sequence having all improvements between 0 and ϵ, by $(nk\phi)^\ell \epsilon^{d_1(S)/2} \leq (nk\phi)^{O(kp(S))}\epsilon^{p_1(S)/2}$, using the above rank bounds and Lemma 3.1. Summing over all choices of parameters only introduces a polynomial blow-up.

Control by cyclic sums. The second is more intricate, and is loosely based on a construction in [31]. The bounds proved here hold for arbitrary graphs. The intent is to construct a new matrix Q whose columns lie in the span of L, which cancels the contributions of inactive players, but allows us to perform a similar analysis as above.

Suppose the move (u,i) appears twice in S, or (u,σ_u^0) appears in S. Let τ_0 be the index of the first occurrence of (u,i) in the BR sequence ($\tau_0 = 0$ in the latter case), and let $\tau_1, \tau_2, \ldots$ be all subsequent appearances of $(u,\cdot)$ in the sequence. Suppose τ_m is the second occurrence of (u,i) in the BR sequence. Then we let $\boldsymbol{q}_{u,i} := \sum_{j=1}^m \boldsymbol{\lambda}_{\tau_j}$, noting that the τ_0 is omitted, and we show in Section 3.5 that the entries of $\boldsymbol{q}$ corresponding to inactive players are all 0. Let $Q(S,\boldsymbol{\sigma}^0)$ be the matrix whose columns consist of the $\boldsymbol{q}$'s. To take a union bound on all Q matrices, it suffices to fix the initial strategy of only the active players. Furthermore, we have that $L \cdot A \in [0,\epsilon]^\ell \implies Q \cdot A \in [0, \ell\epsilon]^{d-d_1}$, so we may apply Lemma 3.1 on the matrix Q.

Fixing $p(S)$, there are at most $(nk)^\ell$ choices of the BR sequence, and $k^{p(S)}$ choices of initial strategy profiles for the active players. Thus, we have a union bound of size $k^{p(S)}(nk)^{\ell}$. In Section 3.5, we show that, on any graph, $Q(S, \boldsymbol{\sigma}^0)$ has rank at least $p_2(S)/2$. Thus, for a fixed choice of p, p_1, p_2, d, d_1, q_0, the probability of *any* sequence having all improvements being between 0 and ϵ is then bounded by $(nk\phi)^{O(kp(S))}(\ell\phi\epsilon)^{p_2(S)/2}$, by Lemma 3.1. Summing over all choices of the fixed parameters only introduces a polynomial blow-up.

Conclusion. We conclude Theorem 3.3 from the above bounds, which we summarize here:

Figure 3.2: Summary of the results proven in this chapter, towards the proof of Theorem 3.3.

Graph	Rank Bound	Union Bound	Probability of Success $\forall S$
compl.	$d(S) - q_0(S) + d_1(S)/2$	$k^{p(S)}(4n/\epsilon)^{d(S)-q_0(S)}(nk)^{\ell}$	$1 - (nk\phi)^{O(k \cdot p(S))}\epsilon^{d_1(S)/2}$
gen.	$p_2(S)/2$	$k^{p(S)}(nk)^{\ell}$	$1 - (nk)^{O(k \cdot p(S))}(\ell\phi\epsilon)^{p_2(S)/2}$

The result on general graphs uses only the bounds from the cyclical sum construction, and a lemma of [31] which ensures that any sequence must contain a sub-sequence S' such that $p_2(S') \geq \Omega(p(S')/\log(nk))$. Applying the above bounds with $\epsilon = 1/(nk\phi)^{O(k\log(nk))}$ gives the desired result for general graphs. For complete graphs, we restrict ourselves to critical blocks as described above, using rounding when $p_1(S) \geq p_2(S)$, and cyclic sums when $p_2(S) \geq p_1(S)$. Setting $\epsilon = 1/(nk\phi)^{O(k)}$ for both gives the second half of Theorem 3.3. Along with the details presented in the rest of the chapter, this concludes our proof of Theorem 3.3, and as a result, Theorem 3.2.

3.3.2 Matrix Notation and Goals

Fix a game graph $G = (V, E)$, and random vector of payoff values $A \in [-1, 1]^{|E|k^2}$, where, for all $uv \in E$, $A((u, i)(v, j))$ is the payoff that players u and v receive in the uv game when u plays $i \in [k]$ and v plays $j \in [k]$. For a sequence of BR moves $S = (u_1, \sigma_1), \ldots, (u_\ell, \sigma_\ell)$ starting at initial strategy profile $\boldsymbol{\sigma}^0 \in [k]^n$, the strategy profiles over the sequence are defined as $\boldsymbol{\sigma}^t := (\sigma_t, \boldsymbol{\sigma}^{t-1}_{-u_t})$. Recall, $L(S, \boldsymbol{\sigma}^0) := \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell\}$, as defined above.

Recall, we wish to prove that for any sufficiently long sequence, at least one move must have increased the potential significantly. Thus, we define the following:

Definition 3.3 (Minimum Improvement)**.** For a fixed sequence S and initial state $\boldsymbol{\sigma}^0$, recall that the t-th entry of $L(S, \boldsymbol{\sigma}^0) \cdot A$ is the value $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$. Since A is random, these values are random, and not necessarily positive. If some entry is negative, then S is not a

BR sequence. We define Δ_N as the increase in potential of the worst BR sequence. Formally,

$$\Delta_N = \min_{S,\boldsymbol{\sigma}^0} \|L(S,\boldsymbol{\sigma}^0)\cdot A\|_\infty \quad \text{subject to} \quad |S| = N, \;\; L(S,\boldsymbol{\sigma}^0)\cdot A \geq \mathbf{0} \tag{3.10}$$

Observe that for $N = O(nk)$, if $\Delta_N = 1/poly(n^k, \phi)$, then the running time concluded in Theorem 3.3 is polynomial, and if $\Delta_N = 1/poly(n^{k\log nk}, \phi)$, then the running time is quasi-polynomial. Thus, Theorem 3.3 is equivalent to showing that

$$\Pr_A\left[\Delta_{2nk} < \phi^{-1}(2n^2k^3)^{-20k\log(nk)}\right] \leq \tfrac{1}{(nk)^2} \qquad \text{in general} \tag{3.11a}$$

$$\Pr_A\left[\Delta_{2nk} < (20\phi^2n^3k^3)^{-4k-4}\right] \leq \tfrac{1}{(nk)^3} \qquad \text{for complete game graphs} \tag{3.11b}$$

It remains to give the proof of this result, which we will do by case analysis. We wish to distinguish between the cases when $p_1(S) \leq p_2(S)$, and the converse, following the outline above. We define $\overline{\Delta}_N$ and $\underline{\Delta}_N$ to be defined similarly, but requiring that $p_1(S) \geq p_2(S)$ and $p_1(S) \leq p_2(S)$, respectively.

3.4 RANK BOUNDS AND UNION BOUNDS VIA ROUNDING, IN COMPLETE GRAPHS

This section formalizes the outline presented in Section 3.3.1 in the "rounding" construction, and proves the stated bounds. The cyclic-sum construction is analyzed in the following section. We begin with the case where the game graph is *complete*.

We'll first define the matrix $\widetilde{L}$ for the rounded values, then introduce the concept of *critical block*, which will allow us to show the desired rank bounds. For this section, we may assume that the graph is *complete*. We will use completeness to ensure that between any two nodes, there is a game edge we may use for rank bounds.

3.4.1 Matrix Construction

Fix a BR sequence S, and let V_0 be the set of inactive players, and V_1 the set of active players. Since we are looking to control the rate at which $\Phi(\boldsymbol{\sigma}^t)$ grows with t, we may without loss of generality assume $\Phi(\boldsymbol{\sigma}^0) = 0$ by adding a constant shift. Formally, let $\Psi(t) := \Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^0)$, which satisfies $\Psi(t) - \Psi(t-1) = \Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$. Further, define

$\widetilde{A}((u,\sigma_u)(v,\sigma_v)) = A((u,\sigma_u)(v,\sigma_v)) - A((u,\boldsymbol{\sigma}_u^0)(v,\boldsymbol{\sigma}_v^0))$. Then

$$\begin{aligned}\Psi(t) &= \sum_{u,v\in V} A((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) - A((u,\boldsymbol{\sigma}_u^0)(v,\boldsymbol{\sigma}_v^0)) \qquad (3.12)\\ &= \sum_{u,v\in V_1} \widetilde{A}((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) + \sum_{w,w'\in V_0} \widetilde{A}((w,\boldsymbol{\sigma}_w^t)(w',\boldsymbol{\sigma}_{w'}^t)) + \sum_{u\in V_1}\sum_{w\in V_0} \widetilde{A}((u,\boldsymbol{\sigma}_u^t)(w,\boldsymbol{\sigma}_w^t))\end{aligned}$$

Rounding the effect of inactive players. Now, for $w \in V_0$, $\boldsymbol{\sigma}_w^t = \boldsymbol{\sigma}_w^0$, so middle terms on the second line are 0. Furthermore, the rightmost terms are in fact constants, depending only on σ_u. Let then $C(u,\sigma) := \sum_{w\in V_0} \widetilde{A}((u,\sigma)(w,\boldsymbol{\sigma}_w^0))$. Then the above sum can be expressed as

$$\Psi(t) \ := \sum_{u,v\in V_1} \widetilde{A}((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) \ + \ 0 \ + \sum_{u\in V_1} C(u,\sigma_u^t) \qquad (3.13)$$

Observe also that $C(u,\sigma_u^0) = 0$, since the $\widetilde{A}$ terms cancel. Finally, instead of controlling for $\Psi(t) - \Psi(t-1)$ exactly, it suffices to control for an approximation thereof, to approximate the change in Φ.

We round the C values to the nearest multiple of ϵ, as was first introduced in [32]. Let $C'(u,\sigma)$ be the nearest multiple of ϵ to $C(u,\sigma)$. Since $C(u,\sigma) \in [-n,n]$ for all $u \in V_1$ and $\sigma \in [k]$, then there are $2n/\epsilon$ possible choices for $C'(u,\sigma)$. Let $\Psi'(t) := \sum_{u,v\in V_1} \widetilde{A}((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) + \sum_{u\in V_1} C'(u,\sigma_u^t)$, the approximation to Ψ obtained by using C' terms instead of C. Since $\Psi(t) - \Psi(t-1)$ depends only on two C terms, namely $C(u,\sigma_u^t)$ and $C(u,\sigma_u^{t-1})$, we have

$$|C(u,i) - C'(u,i)| < \epsilon/2 \implies \Big|\big(\Psi(t) - \Psi(t-1)\big) - \big(\Psi'(t) - \Psi'(t-1)\big)\Big| \leq \epsilon \qquad (3.14)$$

And therefore $\Pr[\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1}) \in (0,\epsilon)] \leq \Pr[\Psi'(t) - \Psi'(t-1) \in (-\epsilon, 2\epsilon)]$. This new definition of potential Ψ will allow us to reduce the needed union bounds.

Union bound size. Let $\widetilde{L}$ be obtained from L where for $u \in V_1$, and $i \in [k]$, we replace the set of rows $\{(u,i)(w,j) : w \in V_0, j \in [k]\}$ with the single row for $C(u,i)$ — and therefore $C'(u,i)$. To define $\widetilde{L}$, it suffices to control $\boldsymbol{\sigma}^0$ restricted to the active players, the sequence S, and the values of the C' terms. Recall that $C'(u,\sigma_u^0) = 0$. Furthermore, the only $C(u,i)$ terms which are considered are those for moves (u,i) which appear in the sequence. Therefore, we only have $d(S)$ terms to control, $q_0(S)$ of which are 0, so we get a union bound of size $k^{p(S)}(nk)^{\ell}(4n/\epsilon)^{d(S)-q_0(S)}$.

3.4.2 Critical Subsequences

As discussed in Section 3.3, to get our rank bounds, we will want a sub-sequence S' whose length is proportional to $d(S') - q_0(S')$. To this end, we define *critical subsequences* below. These are closely based on the definition of a critical block in [32].

Definition 3.4 (Critical Subsequence)**.** Similar to the values defined in Definition 3.2, for every contiguous subsequence B of S, let $\ell(B)$, $d(B)$, and $q_0(B)$ be length, number of distinct pairs, and number of return moves, in B, respectively. Such a subsequence is termed *critical* if $\ell(B) \geq 2\big(d(B) - q_0(B)\big)$, but for every $B' \subseteq B$, $\ell(B') < 2\big(d(B') - q_0(B')\big)$.

Note that a return move — *i.e.* q_0-type move — for a subsequence B which starts at time t_B is a move $(u, \boldsymbol{\sigma}_u^{t_B})$, as opposed to a $(u, \boldsymbol{\sigma}_u^0)$ move. We show that critical subsequences must exist.

Lemma 3.2. A critical subsequence always exists in any sequence S of length $2nk$. Furthermore, if B is a critical subsequence, then $\ell(B) = 2(d(B) - q_0(B))$.

Proof. As there are at most nk distinct player-strategy pairs possible, the entire sequence S satisfies the relation $\ell(S) \geq 2d(S) \geq 2(d(S) - q_0(S))$. Conversely, for every subsequence B of length 1 (*i.e.* a single move), $d(B) = 1, q_0(B) = 0 \Rightarrow 1 = \ell(B) < 2(d(B) - q_0(B)) = 2$. Thus, it suffices to take an inclusion-minimal subsequence which satisfies $\ell(B) \geq 2(d(B) - q_0(B))$ and obtain a critical subsequence.

It remains to show that for B critical $\ell(B) = 2d(B) - 2q_0(B)$. Suppose not, then it is strictly larger. Let B' be obtained from B by dropping the last column. Then,

$$\ell(B') \ = \ \ell(B) - 1 \ \geq \ 2d(B) - 2q_0(B) + 1 - 1 \tag{3.15}$$

Now, we claim $d(B) - q_0(B) \geq d(B') - q_0(B')$. Clearly $d(B) - 1 \leq d(B') \leq d(B)$, and $q_0(B) - 1 \leq q_0(B') \leq q_0(B)$. However, if $q_0(B') = q_0(B) - 1$, then we must also have $d(B') = d(B) - 1$. Thus, in all cases, $d(B) - q_0(B) \geq d(B') - q_0(B')$. This implies $\ell(B') \geq 2(d(B') - q_0(B'))$, contradicting the criticality of B. *Q.E.D.*

The tight bound $\ell(B) = 2(d(B) - q_0(B))$ is key for proving the main rank lemma from this section, below. We show that critical subsequences have high rank, which by Lemma 3.2 extends to any length-$2nk$ sequence.

We extend the definition of Δ_N (Def. 3.3) to include the use of critical sequences.

Definition 3.5. We define $\Delta'(p)$ to be the minimum total increase due to any critical subsequence with exactly p active players, where the initial strategy profile is chosen arbitrarily. Formally,

$$\Delta'(p) = \min_{\boldsymbol{\sigma}^0} \|L(S, \boldsymbol{\sigma}^0) \cdot A\|_\infty \text{ subject to } S \text{ critical}, \quad p(S) = p, \quad L(S, \boldsymbol{\sigma}^0) \cdot A \geq \mathbf{0} \quad (3.16)$$

Similarly, $\overline{\Delta}'(p)$ and $\underline{\Delta}'(p)$ represent the same value, with the extra restriction that $p_1(S) \geq p_2(S)$ and $p_1(S) \leq p_2(S)$, respectively. Observe, $\Delta'(p) = \min\{\overline{\Delta}'(p), \underline{\Delta}'(p)\}$.

Since any sequence must have a critical subsequence, $\Delta_N \geq \min_{p=1}^{N} \Delta'(p)$, we will take a union bound to show that the probability that $\Delta_N < 1/(nk\phi)^{O(k)}$ is small. This bound is performed separately for the two cases specified, $\min\{\overline{\Delta}'(p)$ and $\underline{\Delta}'(p)\}$.

3.4.3 Rank Bounds from Separated Blocks

We provide here the main rank bound of this section, which we begin with a definition

Definition 3.6 (Separated Blocks)**.** Fix a BR sequence S, and let $P_1(S)$ be the set of non-repeating active players. For $u \in P_1$, let T_u be the set of indices where the moving player is u. Let $T = \bigcup_{u \in P_1} T_u$, and denote without loss of generality $T = \{t_1 < t_2 < \cdots < t_m\}$. We will show below how the t_i's "separate" the sequence S, since we will be able to control their ranks separately. To this end, let S_i for $i = 0, 1, \ldots, m$ be the subsequences of S from time t_i to t_{i+1} excluding boundaries, respectively, where $t_0 = 0$ and $t_{m+1} = |S|$. We say these S_i's are the *separated blocks* of S, and denote their collection as $\mathbb{S} = \{S_0, S_1, \ldots, S_m\}$. Furthermore, note that $|T| = d_1(S)$.

The following lemmas allows us to take advantage of this notion of separated block, to break up the rank bounds into simpler subproblems.

Lemma 3.3. Assume the game graph is complete, and let S be a BR sequence with at least one inactive player, and let $L = L(S, \boldsymbol{\sigma}^0) = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots\}$. Then L contains at least $d_1(S) + \sum_{S' \in \mathbb{S}} d(S') - q_0(S')$ linearly independent vectors.

Proof. Let w be some inactive player, which we have assumed exists. Let $T = \{t_1 < t_2 < \cdots < t_m\}$ be the endpoints of the separated blocks, as in Definition 3.6 above. For $i = 0, 1, \ldots, m$, let D_i be the set of distinct (player,strategy) moves which occur in S_i, which are not *return moves* of S_i, *i.e.* $(u, \boldsymbol{\sigma}_u^{t_i})$ moves.

For all i, the move at t_i must be some non-repeating player of S, which we denote v_i, and call the strategy it moves to as σ^i, letting v_0 be the inactive player w, and $\sigma^0 := \sigma_w^0$. For all $(u,\sigma) \in D_i$, let $\tau^i_{(u,\sigma)}$ be the time of the *first* occurrence of (u,σ) in the subsequence S_i, and let $H_i = \{\tau^i_{(u,\sigma)} : (u,\sigma) \in D_i\}$. Let $H = \bigcup_{i=0}^{|\mathbb{S}|-1} H_i \cup \{t_1, \ldots, t_{|\mathbb{S}|-1}\}$. For each $t \in H$, if $t = \tau^i_{(u,\sigma)} \in H_i$ for some i, u, σ, then *associate* to L_t the row $((u,\sigma)(v_i,\sigma^i))$. If, instead, $t = t_i$ for some i, then *associate* to L_t the row $((v_i,\sigma^i)(w,\sigma^0))$. Note that this row exists because the game graph is complete.

Consider the submatrix of L consisting of all columns $\{\boldsymbol{\lambda}_t : t \in H\}$, sorted in "chronological" order, and all of their associated rows, in the same order as their respectively associated columns. We claim that this matrix is upper-triangular, and its diagonal entries are nonzero. For each column $\boldsymbol{\lambda}_t$, the diagonal entry in the submatrix is the entry for the associated row, which we have chosen to be nonzero. Furthermore, if $t = t_i \in H$, then v_i all previous columns have 0 entries in the $((v_i,\sigma^i)(\cdot,\cdot))$ rows, since v_i is non-repeating. Thus, $\boldsymbol{\lambda}_{t_i}$ is the first column where the associated row has a nonzero entry. If, instead, $t = \tau^i_{(u,\sigma)} \in H_i$, then the associated row $((u,\sigma)(v_i,\sigma^i))$ must have been 0 up until column $\boldsymbol{\lambda}_{t_i}$ as described above. Furthermore, since $\tau^i_{(u,\sigma)}$ is the first occurrence of $(u, \sigma \neq \boldsymbol{\sigma}^{t_i}_u)$ after time t_i, we must have had the row $((u,\sigma)(v_i,\sigma^i))$ be 0 before the $\tau^i_{(u,\sigma)}$-th column.

These observations imply that our $|H| \times |H|$ submatrix, with the given row-ordering, must be upper-triangular with nonzero diagonal terms. Therefore, it must be full-rank. Since $|H_i| = d(S_i) - q_0(S_i)$, then $|H| = d_1(S) + \sum_{S' \in \mathbb{S}} d(S') - q_0(S')$, and we conclude the desired bound. *Q.E.D.*

We also extend this proof to the case of all players active.

Corollary 3.1. Let S be a BR sequence where all players are active, and let $L = L(S, \boldsymbol{\sigma}^0)$. Then L contains at least $\left(1 - \frac{1}{n}\right)(d(S) - q_0(S))$ linearly independent vectors.

Proof. Consider the above proof method with $|T| = 0$, and $S_0 = S$. Note that now, $H = D_0$. It is still correct if some arbitrary player is chosen to be the w player, and all $((u,\sigma)(v_0,\sigma^0))$ terms are replaced with $((u,\sigma)(w, \boldsymbol{\sigma}^{\tau^0_{(u,\sigma)}}))$ terms. We must further restrict H not to contain any moves of player w. Suppose we choose, as our w player, the player which appears the least number of times in H, then we suffer a $\left(\frac{1}{n}\right)$-fraction loss in the size of H, concluding the proof. *Q.E.D.*

This above lemma and corollary, along with the notion of critical block, will give us our desired bound.

Lemma 3.4. Assume a complete game graph, and let S be a BR sequence of length $2nk$ which has at least one inactive player. Let B be some critical subsequence of S starting at t_0, and let $L = L(B, \boldsymbol{\sigma}^{t_0})$. Then L contains at least $\frac{1}{2}d_1(B) + d(B) - q_0(B)$ linearly independent vectors.

Proof. Since S has an inactive player, then so must B. Therefore, Lemma 3.3 applies. Recall, Lemma 3.3 shows that L contains at least $d_1(B) + \sum_{S' \in \mathbb{S}(B)} d(S') - q_0(S')$ linearly independent vectors. If $p_1(B) = d_1(B) = 0$, then we are done. Otherwise, since B is critical, then for all $S' \in \mathbb{S}(B)$, $\ell(S') < 2(d(S') - q_0(S'))$. Hence,

$$rank(L) \ \geq \ d_1(B) + \sum_{S' \in \mathbb{S}(B)} d(S') - q_0(S') \ > \ \tfrac{1}{2}d_1(B) + \tfrac{1}{2}d_1(B) + \sum_{S' \in \mathbb{S}(B)} \tfrac{1}{2}\ell(S') \tag{3.17}$$

However, $\ell(B) = d_1(B) + \sum_{S' \in \mathbb{S}(B)} \ell(S')$, and so this implies $rank(L) \geq \frac{1}{2}d_1(B) + \frac{1}{2}\ell(B)$. By criticality and Lemma 3.2, $\ell(B) \geq 2(d(B) - q_0(B))$, giving us our desired bound. *Q.E.D.*

This concludes the first rank- and union-bound of Section 3.3.1. Using this, and Lemma 3.1, we show our first result:

Theorem 3.4. $\Pr\left[\,\overline{\Delta}'(p) \in (0, \epsilon)\right] \leq \left((20\phi^2 n^3 k^3)^k \, \epsilon^{1/4}\right)^p$.

Proof. From the above lemmas, it remains to apply Lemma 3.1. For a fixed critical subsequence S with p active players, if $p_1 \geq p_2$, then by Lemma 3.4, the improvement of each step of the *approximate* potential Ψ along the sequence will lie in $(-\epsilon, 2\epsilon)$ with probability $(3\phi\epsilon)^{d(S)-q_0(S)+p(S)/4}$. Taking a union bound over *all* approximated sequences, this event holds with probability

$$k^{p(S)}(nk)^{\ell(S)}(2n/\epsilon)^{d(S)-q_0(S)}(3\phi\epsilon)^{d(S)-q_0(S)+p(S)/4} \tag{3.18}$$

Noting that $d(S) - q_0(S) \leq k \cdot p(S)$, and by criticality of S, $\ell(S) \leq 2d(S) - 2q_0(S) \leq 2kp(S)$, so

$$\begin{aligned} \Pr\left[\,\overline{\Delta}'(p) \in (0, \epsilon)\right] &\leq k^{p(S)}(nk)^{\ell(S)}(2n/\epsilon)^{d(S)-q_0(S)}(3\phi\epsilon)^{d(S)-q_0(S)+p(S)/4} \\ &\leq 20^{k \cdot p(S)}(nk\phi)^{2k \cdot p(S)}(nk)^{k \cdot p(S)}\epsilon^{p(S)/4} \\ &= \left((20n^3k^3\phi^2)^k \epsilon^{1/4}\right)^{p(S)} \end{aligned} \tag{3.19}$$

as desired. *Q.E.D.*

3.5 RANK BOUNDS AND UNION BOUNDS VIA CYCLIC SUMS, FOR GENERAL GRAPHS

In this subsection, we will prove the second half of the results from Section 3.3.1. Unlike the rank bounds of the previous section, all statements in this section hold for arbitrary game graphs.

Recall that, for a fixed BR sequence S, we have defined the matrix $Q(S, \boldsymbol{\sigma}^0)$, whose columns consist of sums of columns of $L(S, \boldsymbol{\sigma}^0) = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_\ell\}$. We recall its definition here: Let (u, i) be a move which appears twice in S — possibly a (u, σ_u^0) return move. Let τ_0 be the index of the first occurrence of (u, i) in the BR sequence, setting $\tau_0 = 0$ for return moves. Let $\tau_1, \tau_2, \ldots$ be the indices of all subsequent moves by player u in the sequence, and suppose τ_m is the second occurrence of (u, i) in the BR sequence, or first, if it is a return move. Define $\boldsymbol{q}_{u,i} := \sum_{j=1}^{m} \boldsymbol{\lambda}_{\tau_j}$, noting that the τ_0 is omitted, and let $Q(S, \boldsymbol{\sigma}^0)$ be the matrix whose columns consist of the $\boldsymbol{q}$'s.

We wish to show that $Q(S, \boldsymbol{\sigma}^0)$ does not depend on the strategies of the inactive players. Intuitively, this holds because we are taking a "cyclic sum" of the moves of a player u, and therefore we are cancelling out the entering and exiting payoff values. Formally, note that if player u is moving at time t, $\boldsymbol{\lambda}_t$ has only nonzero entries in $(u, \cdot)(\cdot, \cdot)$ rows. Furthermore, if w is inactive, and σ is any strategy played by u, then the $(u, \sigma)(w, \sigma_w^0)$ row of $\boldsymbol{q}_{u,i}$ is given by

$$\boldsymbol{q}_{u,i}\big((u,\sigma)(w,\sigma_w^0)\big) \;=\; \sum_{j=1}^{m} \boldsymbol{\lambda}_{\tau_j}\big((u,\sigma)(w,\sigma_w^0)\big) \;=\; \sum_{j=1}^{m} \mathbb{I}[\sigma_u^{\tau_j} = \sigma] - \mathbb{I}[\sigma_u^{\tau_{j-1}} = \sigma] \;=\; 0 \quad (3.20)$$

Thus, we have that to fully specify $Q(S, \boldsymbol{\sigma}^0)$, it suffices to know S and the initial strategy profiles of the active players. It remains to show $Q(S, \boldsymbol{\sigma}^0)$ has large rank, as follows.

Lemma 3.5. Fix a BR sequence S and starting configuration $\boldsymbol{\sigma}^0$. Then $Q = Q(S, \boldsymbol{\sigma}^0)$ contains at least $p_2(S)/2$ linearly independent vectors.

Proof. We begin by constructing an auxiliary *directed* graph $G' = (V, E')$, where V is the set of players, and E' will be defined as follows: let (u, σ) be some repeating move. We cannot have $\boldsymbol{q}_{u,\sigma}$ be the all-zero vector, as otherwise $\boldsymbol{\sigma}^{\tau_0} = \boldsymbol{\sigma}^{\tau_m}$, which cannot hold for a BR sequence. For *every* player $w \in V$ such that $\boldsymbol{q}\big((u,\sigma)(w,\sigma')\big) \neq 0$ for some $\sigma' \in [k]$, add the directed edge (u, w) to E'.

Let $P_2 \subseteq V$ be the set of repeating players, and note that they all have non-zero out-degree. Consider the following procedure: pick a vertex $r_1 \in P_2$, and let T_1 be the BFS arborescence rooted at r_1 which spans all nodes reachable from r_1 in G'. Then delete $V(T_1)$ from G' and repeat, picking an arbitrary root vertex $r_2 \in P_2 \setminus V(T_1)$, and get the arborescence T_2 on everything reachable from r_2. We may continue this until every vertex of P_2 is covered by some arborescence. For each $i = 1, 2, \ldots$, let T_i^0 and T_i^1 be the set of nodes of T_i which are of even or odd distance from r_i along T_i, respectively. Let P_i' be the larger of $V(T_i^0) \cap P_2$ and $V(T_i^1) \cap P_2$, and $P_2' := \bigcup_{i=1}^{\infty} P_i'$.

We must have that $|P_2'| \geq |P_2|/2 = p_2(S)/2$. We wish to show that the collection $\mathcal{V} := \{\boldsymbol{q}_{u,\cdot} : u \in P_2'\}$ is independent. Every $u \in P_2'$ must have some out-neighbour w. If u was not a leaf of the arborescence it was selected in, then it must have some out-neighbour along the arborescence, and we may choose this neighbour. This out-neighbour can not also be in P_2'. In this case, $\boldsymbol{q}_{u,\cdot}$ will be the only vector from $\mathcal{V}$ to contain a non-zero $((u,\cdot)(w,\cdot))$ entry, since w was not taken in P_2'. If, instead, u was a leaf of its arborescence, then its out-neighbours must be in previously constructed arborescences. Let w be any such neighbour, then $\boldsymbol{q}_{w,\cdot}$ can not contain a non-zero $((u,\cdot)(w,\cdot))$ entry, as otherwise u would have been in the other arborescence. Therefore, $\boldsymbol{q}_{u,\cdot}$ is the only vector in $\mathcal{V}$ to contain a nonzero $((u,\cdot)(w,\cdot))$ entry. Thus, $\mathcal{V}$ must contain a $|\mathcal{V}| \times |\mathcal{V}|$ diagonal submatrix, and therefore has rank at least $|\mathcal{V}| \geq p_2(S)/2$, as desired. *Q.E.D.*

Using the above lemma along with the appropriate union bound discounting the inactive players, we show the following Theorem:

Theorem 3.5. $\Pr\left[\underline{\Delta}'(p) \in (0,\epsilon)\right] \leq \left(2(nk)^{2k}k^{5/4}(n\phi\epsilon)^{1/4}\right)^p$.

Proof. Fix S, $\boldsymbol{\sigma}^0$, and let $L = L(S,\boldsymbol{\sigma}^0)$. Let A be the payoff vector of the network coordination game. Let $\mathcal{V}$ be a collection of $p_2(S)/2$ independent vectors from Lemma 3.5. Let $\boldsymbol{q} \in \mathcal{V}$ and recall $\boldsymbol{q} = \sum_{j=1}^m \boldsymbol{\lambda}_{t_j}$ for some collection of indices $t_1 < \cdots < t_m$. We have $\Pr[\bigwedge_{i=1}^m \langle \boldsymbol{\lambda}_{t_i}, A\rangle \in (0,\epsilon)] \leq \Pr[\langle \boldsymbol{q}, A\rangle \in (0, m\epsilon)]$. Since $m \leq \ell$, taking the collection of all $\boldsymbol{q}$ vectors and applying Lemma 3.1, we have

$$\Pr\left[\bigwedge\nolimits_{t=1}^{\ell(S)} \langle \boldsymbol{\lambda}_t, A\rangle \in (0,\epsilon)\right] \leq \Pr\left[\bigwedge\nolimits_{\boldsymbol{q}\in\mathcal{V}} \langle \boldsymbol{q}, A\rangle \in (0,\ell\epsilon)\right] \leq (\ell\phi\epsilon)^{p_2(S)/2} \tag{3.21}$$

There are at most $k^{p(S)}(nk)^{\ell(S)}$ possible collections $\mathcal{V}$. The quantity $\underline{\Delta}'(p)$ assumes we are in a critical subsequence, which implies $\ell(S) = 2(d(S) - q_0(S))$ by Lemma 3.2 Since $\ell(S) = 2(d(S) - q_0(S)) \leq k \cdot p(S)$, and $p_2 \geq p_2 \implies p_2(S) \geq \frac{1}{2}p(S)$, we have

$$\begin{aligned}\Pr\left[\underline{\Delta}'(p) \in (0,\epsilon)\right] &\leq k^{p(S)}(nk)^{\ell(S)}(\ell\phi\epsilon)^{p_2(S)/2} \\ &\leq n^{2k\cdot p(S)}k^{(2k+1)p(S)}(2kp)^{p(S)/4}(\phi\epsilon)^{p(S)/4} \\ &\leq \left(2(nk)^{2k}k^{5/4}(n\phi\epsilon)^{1/4}\right)^{p(S)} \end{aligned} \tag{3.22}$$

as desired. *Q.E.D.*

3.6 COMBINING THE BOUNDS

This section, using the results above, concludes the proof of Theorem 3.3, which in turn implies the main result, Theorem 3.2.

3.6.1 Polynomial Smoothed Complexity for Complete Game Graphs

We have shown above that $\overline{\Delta}'(p)$ and $\underline{\Delta}'(p)$ have vanishing probability of lying in $(0, \epsilon)$. In this section, we use these results to show that the BRA will terminate in time polynomial in n^k, k and ϕ, with high probability, when the game graph is complete. The following lemma combines our two previous results:

Lemma 3.6. With probability $1 - 1/O(\phi^2 n^3 k^4)$, every BR sequence of length $2nk$ must have an improvement of at least $\epsilon = O\big((\phi^2 n^3 k^3)^{-4k-4}\big)$.

Proof. We will perform a case analysis based on the values of $p_1(S)$ and $p_2(S)$, with cases for $p(S) = n$, $p(S) < n$ and $p_1(S) \geq p_2(S)$, and $p_2(S) \geq p_1(S)$.

If $p(S) = n$, we apply the rank bound of Corollary 3.1 and take a union bound over all initial strategy profiles, and all possible sequences to get

$$\Pr[\Delta'(n) \in (0, \epsilon)] \leq k^n (nk)^{2nk} (\phi\epsilon)^{n-1} \leq \left(k^{3k} n^{2k} \phi\epsilon\right)^n \Big/ \phi\epsilon \tag{3.23}$$

This union bound over-counts the number of sequences with $p(S) = n$, but this isn't a problem. Setting $\epsilon = \phi^{-1}\left(n^2 k^3\right)^{-2k}$ gives $\Pr[\Delta'(n) \in (0, \epsilon)] \leq \left(\frac{1}{n^2 k^3}\right)^{n-2}$.

In the converse case, we combine Theorems 3.4 and 3.5, then take a union bound over all possible values of p to bound the probability for any sequence of the given length. As defined previously, $\Delta'(p) = \min\{\overline{\Delta}'(p), \underline{\Delta}'(p)\}$ and so,

$$\begin{aligned}
\Pr[\Delta'(p) \in (0, \epsilon)] &\leq \left((20\phi^2 n^3 k^3)^k \epsilon^{1/4}\right)^p + \left(2(\phi\epsilon)^{1/4} n^{2k+1/4} k^{2k+5/4}\right)^p \\
&\leq 2\left((20\phi^2 n^3 k^3)^k \epsilon^{1/4}\right)^p
\end{aligned} \tag{3.24}$$

Since any sequence of length $2nk$ must contain a critical subsequence, it suffices to set $\epsilon = \left(20\phi^2 n^3 k^3\right)^{-4k-4}$, and taking the union bound over all choices of p, we get

$$\Pr[\Delta_{2nk} \in (0, \epsilon)] \leq \sum_{p=1}^{n} \left(20\phi^2 n^3 k^4\right)^{-p} \leq \frac{1}{(20\phi^2 n^3 k^4) - 1} \tag{3.25}$$

Combining the two cases of $p = n$ and $p < n$ gives us our desired result. *Q.E.D.*

This concludes the proof of the complete-game-graphs part of Theorem 3.3, noting that $\phi \geq \frac{1}{2}$. This in turn implies that with probability $1 - 1/poly(n, k, \phi)$, any correct implementation of the BRA will converge to a PNE of the network coordination game in at most $(nk\phi)^{O(k)}$ steps.

3.6.2 Quasipolynomial Smoothed Complexity for General Game Graphs

We now show the quasi-polynomial running time when the game graph G is incomplete, and thus prove the remaining part of Theorem 3.2. The analysis mostly uses the lemmas from Section 3.5, paired with the following definition and lemma from [31]:

Definition 3.7. Recall the random variable Δ from Definition 3.3. Call a sequence of length ℓ log-repeating if it contains at least $\ell/(5\log(nk))$ repeating moves (pairs). We denote as $\Delta''(\ell)$ the minimum total potential-improvement after any log-repeating BR sequence of length exactly ℓ.

Lemma 3.7 ([31], Lemma 3.4)**.** Let Δ_N and $\Delta''(\ell)$ be as above. Then $\Delta_{5nk} := \min_{1 \leq \ell \leq 5nk} \Delta''(\ell)$.

The proof of the above lemma shows that any sequence on $5nk$ pairs must contain some contiguous sub-sequence which is log-repeating. Thus, for the remainder of the analysis, it suffices to bound $\Delta''(\ell)$. Since a sequence captured by $\Delta''(\ell)$ must have at least $\ell/(5\log(nk))$ repeated terms, it must have $p_2 \geq \ell/(5k\log(nk))$. Therefore, as we have shown in the proof of Theorem 3.5, we have $\Pr[\Delta''(\ell) \in (0, \epsilon)] \leq k^\ell (nk)^\ell (\ell\phi\epsilon)^{\ell/10k\log(nk)}$. It suffices, then to simply take the union bound over all possible values of ℓ.

Theorem 3.6. Given a smoothed instance of *k-NetCoordNash* with an arbitrary initial strategy profile, then any execution of a BR algorithm where improvements are chosen arbitrarily will converge to a PNE in at most $\phi \cdot (nk)^{O(k\log(nk))}$ steps, with probability $1 - (nk)^{-2}$.

Proof. As discussed above,

$$\begin{aligned} Pr[\Delta''(\ell) \in (0, \epsilon)] &\leq k^\ell (nk)^\ell (\ell\phi\epsilon)^{\ell/10k\log(nk)} \\ &\leq \left(k^2 n (5nk\phi\epsilon)^{1/(10k\log(nk))}\right)^\ell && (\ell \leq 5nk) \\ &\leq \left(2k^3 n^2 (\phi\epsilon)^{1/(10k\log(nk))}\right)^\ell . && (5^{1/10} \leq 2) \end{aligned} \tag{3.26}$$

Setting $\epsilon = \phi^{-1}(2n^2k^3)^{-2\cdot 10k\log(nk)}$, this gives

$$Pr[\Delta''(\ell) \in (0, \epsilon)] \leq \left(\frac{1}{2n^2k^3}\right)^\ell \tag{3.27}$$

Let Δ_{5nk} be the improvement in potential in any length $5nk$ BR sequence. Then using Lemma 3.7, and taking the union bound over all choices of ℓ, we have,

$$\begin{aligned}\Pr[\Delta_{5nk} \in (0,\epsilon)] &\leq \sum_{\ell=1}^{5nk} \Pr[\Delta''(\ell) \in (0,\epsilon)] \\ &\leq \sum_{\ell=1}^{5nk} (2n^2k^3)^{-\ell} \leq \frac{(2n^2k^3)^{-1}}{1-(2n^2k^3)^{-1}} = \frac{1}{2n^2k^3-1} \leq \frac{1}{(nk)^2} \end{aligned} \tag{3.28}$$

Hence, with probability $1 - 1/\mathit{poly}(n,k)$ (over the draw of payoff vector A), *all* BR sequences of length $5nk$ will have total improvement at least ϵ. In that case, any execution of BR algorithm makes an improvement of at least ϵ every $5nk$ moves. Since the total improvement is at most $2n^2$, we conclude that the total number of steps is at most $5nk \cdot 2n^2/\epsilon = 10n^3k(2n^2k^3)^{20k\log(nk)} \cdot \phi = \phi \cdot (nk)^{O(k\log(nk))}$, and this occurs with probability $1 - 1/\mathit{poly}(n,k)$. *Q.E.D.*

This concludes the proof of the arbitrary-graphs part of Theorem 3.3, noting that $\phi \geq \frac{1}{2}$. This in turn implies that with probability $1 - 1/\mathit{poly}(n,k,\phi)$, any correct implementation of the BRA will converge to a PNE of the network coordination game in at most $(nk\phi)^{O(k\log(nk))}$ steps.

This completes our analysis of the smoothed performance of BRA for finding pure Nash equilibria in network coordination games. In the next section, we show that this result indeed holds in expectation, and then go on to show a notion of smoothness-preserving reduction which allows us to prove alternative, conditional, algorithms for this problem.

3.6.3 (Quasi)Polynomial Running time in Expectation

The analysis in the previous section establishes smoothed complexity of network coordination games with respect to the *with high probability* notion. Another aspect of smoothed analysis is to analyze the expected time of completion of the algorithm. Observe that the expected running time of an algorithm can not be immediately concluded from the high-probability running time, and this performance will depend on the explicit bounds computed. In this section, we provide a theorem to obtain expected time results from the with high probability bounds. The results are presented in a general form to allow application to any problem in PLS that has a bounded total improvement in potential value. The following theorem is a generalization of the statement of a result found in [31]. We include the analysis for completeness.

Theorem 3.7. Given a PLS problem with input size N, potential function range $[-N^{r_1}, N^{r_2}]$, and a local-search algorithm $\mathcal{A}$ to solve it, let d be the number of distinct choices the algorithm has in each step and let Λ be the total size of the search space of the algorithm. For an instance I drawn at random with maximum density ϕ with probability at most $\sum_{q=1}^{N^\beta}(N^{f(N)}(\phi^{g'(N)}\epsilon)^{1/g(N)})^q$ that any length-N^β sequence of improving moves of $\mathcal{A}$ results in total improvement in the potential value at most ϵ; the algorithm runs in expected time $O(\phi^{g'(N)} \cdot N^{\beta+r} \cdot g(N) \cdot N^{f(N)g(N)} \cdot \ln \Lambda)$. Here, $f(N)$ $g'(N)$ and $g(N)$ are functions of N.

Proof. The maximum improvement possible before $\mathcal{A}$ terminates is the maximum change in the potential function value, given by $N^{r_2} + N^{r_1}$. For any integer $t \geq 1$, if the algorithm requires more than t steps to terminate, then there must exist some subsequence of length N^β that results in an improvement in the potential value of less than $N^\beta(N^{r_2} + N^{r_1})/t \leq 2N^{\beta+\max\{r_2,r_1\}}/t$. We denote $r := \max\{r_1, r_2\}$.

We define a random variable T as the number of steps $\mathcal{A}$ requires to terminate. Using the notation $\Delta(N^\beta)$ to denote the minimum total improvement in a length-N^β sequence of the algorithm $\mathcal{A}$, this gives the probability of $\mathcal{A}$ running for more than t steps as:

$$\Pr[T \geq t] \leq Pr[\Delta(N^\beta) \in (0, N^{r+\beta}/t)] \leq \sum_{q=1}^{N^\beta} \left(N^{f(N)} \left(\phi^{g'(N)} \cdot \frac{N^{\beta+r}}{t} \right)^{1/g(N)} \right)^q . \tag{3.29}$$

We define $t = \gamma i$, for $\gamma = N^{f(N)g(N)}(\phi^{g'(N)}N^{r+\beta}) = \phi^{g'(N)}N^{f(N)g(N)+\beta+r}$, and compute the probability of $T \geq \gamma i$ for any integer i:

$$\begin{aligned} \Pr[T \geq \gamma i] &\leq \sum_{q=1}^{N^\beta} \left(N^{f(N)} \left(\phi^{g'(N)} \cdot \frac{N^{r+\beta}}{\gamma i} \right)^{1/g(N)} \right)^q \\ &\leq \textstyle\sum_{q=1}^{\infty} \left(\frac{1}{i}\right)^{q/g(N)} \leq g(N) \sum_{q'=1}^{\infty} \left(\frac{1}{i}\right)^{q'} \leq \frac{g(N)}{i-1}. \end{aligned} \tag{3.30}$$

We now sum over all values of t, by using that $\Pr[T \geq t] \leq \Pr[T \geq t \cdot \lceil t/\gamma \rceil]$, and compute the expected time steps as:

$$\mathbb{E}[T] = \sum_{t=1}^{\Lambda} \Pr[T \geq t] \leq \sum_{i=1}^{\Lambda/\gamma} \sum_{t=1}^{\gamma} \Pr[T \geq (i+1)\gamma] \leq \sum_{i=2}^{\Lambda/\gamma} \frac{g(N)\gamma}{i-1} = O(g(N) \cdot \gamma \cdot \ln \Lambda) \tag{3.31}$$

Thus, replacing the value for γ, the expected runtime is at most

$$O(\phi^{g'(N)}N^{\beta+r}g(N)N^{f(N)g(N)} \ln \Lambda) \ . \tag{3.32}$$

Q.E.D.

Corollary 3.2. The smoothed expected time for BR to terminate for all network coordination games is polynomial in $(n^{(k\log(nk))}, \phi)$.

Proof. From (3.26) in Theorem 3.6, we know that the probability that the minimum improvement in a fixed BR sequence of length $5nk$ is no more than ϵ is at most

$$\textstyle\sum_{\ell=1}^{5nk}\left(2n^2k^3(\phi\epsilon)^{1/(10k\log(nk))}\right)^{\ell}\ . \tag{3.33}$$

Applying Theorem 3.7, for $N = nk$ and $\Lambda \leq k^n$, we get $f(N) = O(1)$, $N^{r+\beta} \leq N^3$, $g'(N) = 1$ and $g(N) = O(k\log(nk))$, and the expected running time is

$$O(\phi(nk)^{O(1)}O(k\log(nk))(nk)^{O(1)\cdot O(k\log(nk))}\ln(k^n)) \equiv O(\phi n^{O(k\log(nk))}). \tag{3.34}$$

Q.E.D.

Corollary 3.3. For complete graphs, the smoothed expected time for BR to terminate for network coordination games is polynomial in (n^k, ϕ).

Proof. From (3.23) in Lemma 3.6, for the case of complete graphs when a BR sequence has all active players, we have $\Pr[\Delta(p) \in (0,\epsilon)] \leq (\phi\epsilon)^{-1}\left(k^{3k}n^{2k}\phi\epsilon\right)^n$ which is in turn bounded by $\sum_{i=1}^{n}\left(k^{3k}n^{2k}\phi^{1/2}\epsilon^{1/2}\right)^i \Big/ \phi\epsilon$. Similarly, from (3.24) in Lemma 3.6, the probability that the minimum improvement in a BR sequence of length $2nk$ is at most ϵ, is given by:

$$\Pr[\Delta(p) \in (0,\epsilon)] \leq \textstyle\sum_{p=1}^{n} 2\left((20\phi^2n^3k^4)^k\epsilon^{1/4}\right)^p \tag{3.35}$$

Combining these sums, we get the probability that a BR sequence of length $2nk$ has improvement at most ϵ is:

$$\begin{aligned}\Pr[\Delta(p) \in (0,\epsilon)] &\leq \max\left\{\sum_{p=1}^{n} 2\left((20\phi^2n^3k^5)^k\epsilon^{1/4}\right)^p, \sum_{i=1}^{n}\left(k^{3k}n^{2k}\phi^{1/2}\epsilon^{1/2}\right)^i\right\} \\ &\leq \sum_{j=1}^{n}(((nk)^{c_1k}(\phi^{c_2k}\epsilon)^{1/c_3})^j, \end{aligned} \tag{3.36}$$

for $c_1 \leq 5, c_2 \leq 8$ and $c_3 \leq 4$.

Applying Theorem 3.7, for $N = nk$, $N^{r+\beta} \leq N^3$, and $\Lambda \leq k^n$, we get $f(N) = O(k)$, $g'(N) = 8k$ and $g(N) = O(1)$. The expected running time is $O(\phi^{8k}\cdot(nk)^3\cdot O(1)\cdot(nk)^{O(k)}n\ln(k))$, which is polynomial in (n^k, ϕ^k). *Q.E.D.*

CHAPTER 4: SMOOTHNESS-PRESERVING REDUCTIONS[1]

This chapter introduces and formally defines a notion of *smoothness preserving reductions* for local search problems, and gives two such reductions in the case of Nash equilibrium computation. The smoothness-preserving reductions refine Karp reductions. An algorithm is said to be smoothed-efficient if, on adversarially chosen combinatorial information, and random real-valued inputs, the algorithm runs in time polynomial in the input size and the degree of perturbation, with high probability.

The idea behind the reductions is to allow a perturbed instance of $\mathcal{P}$ to be mapped to a perturbed instance of $\mathcal{Q}$, preserving sufficient randomness to allow for a smoothed efficient algorithm for $\mathcal{Q}$ to be applied. The output of the algorithm is then mapped back to a solution for the original $\mathcal{P}$ instance.

The following definition formalizes the reduction between search problems. We will assume that these problems have instances $I = (D, X)$ consisting of discrete, or structural, information D, and continuous, real-valued information X, which is the parameter that is perturbed in the definition of smoothness. The reduction will map the structural information just as any Karp reduction would, but is careful to restrict how the real-valued parameters are transformed.

Definition 4.1 (Strong and Weak Smoothness-Preserving Reductions)**.** A *weak (randomized) smoothness-preserving reduction* from a search problem $\mathcal{P}$ to problem $\mathcal{Q}$ is defined by poly-time computable functions f_1 and f_2, a full-row-rank matrix M with polynomially bounded entries, a constant η such that $1/\eta$ is polynomial in the input size, and a real probability space $\Omega \subseteq \mathbb{R}^d$; such that the following holds: For any $I = (D, X) \in \mathcal{P}$ and $R \in \Omega$, $J = (f_1(D), \eta M(X \circ R))$ is an instance of $\mathcal{Q}$, such that if $\boldsymbol{\sigma}$ is a solution to J, then $f_2(\boldsymbol{\sigma})$ is a solution to I. Here, $\circ$ denotes concatenation.

We require that $|f_1(D)|$, the dimension of R, and the size of M, be polynomial in $|I|$, that the probability density of the entries of R be polynomial in $|I|$ and the maximum density on X, and that the entries of R be independently distributed. If M is a diagonal matrix, then this is a *strong* smoothness-preserving reduction.

When the reduction is strong, *i.e.* M is diagonal, the random input to the reduced instance has independently distributed entries, which are required by most smoothed analysis results, and so strong reductions easily extend smoothed efficient algorithms. We conjecture that

[1] This chapter, along with the previous, is based on collaboration with Rucha Kulkarni and Ruta Mehta [58].

for most smoothed analysis, an upper-bound on the joint density of the input values suffices for efficient performance of the algorithm.

Conversely, weak smoothness-preserving reductions will be useful when reducing to problems which have smoothed analysis results following the common framework introduced in Section 3.1. The key is that if the mapping is given by a *full-rank linear map with η-integral coefficients*, then we may directly inherit the common framework: namely, if M_S is the change-in potential matrix from the framework, then $M_S \cdot f(X)$ is the change in potential in the reduced problem, and the composition $M_S \circ f$ has the same rank as M_S, so long as f reduces dimension.

Remark 4.1. At first blush, the extra randomness R is superfluous; however, these variables are introduced to ensure that M has full-rank. Following the common framework, the result of Proposition 3.1 ensures that if the entries of X and R have bounded density, and $|\det(\eta M)| \geq \eta^d$, then the joint distribution on $M(X \circ R)$ has polynomially bounded density.

We give smoothness-preserving reductions from Network-Coordination Nash to the Local Max Cut problems defined in the following, and take advantage of the common framework presented above in Section 3.1.

We introduce here two PLS-complete graph cut problems which are relevant to this writeup, namely Local Max Cut and Local Max Bisection. Both take as input a weighted graph, and seek to find a cut which has maximal weight under a neighbourhood structure among cuts.

Definition 4.2 (Local Max Cut and Bisection, [40])**.** Given a weighted graph $G = (V, E)$, *Local Max Cut* refers to the problem of finding a cut $V = A \cup B$ such that the total weight of the cut $\delta(A, B)$ is maximal up to flipping one node across the cut, namely:

$$\delta(A,B) \geq \begin{cases} \delta(A+u, B-u) & \forall\, u \in B \\ \delta(A-u, B+u) & \forall\, u \in A \end{cases} \tag{4.1}$$

The second problem requires more introduction. A *balanced* cut is a cut $V = A \cup B$ such that $|A| = |B|$, and a *swap* of $V = A \cup B$ is a balanced cut which is obtained by swapping a pair of nodes across the cut. From a balanced cut $V = A \cup B$, we call $V = A' \cup B'$ the (unique) greedy swap from $V = A \cup B$ if it is the swap with the greatest improvement (or least decrease), breaking ties lexicographically. Now, given a weighted graph $G = (V, E)$, *Local Max Bisection* refers to the problem of finding a balanced cut $V = A \cup B$ whose value is greater than that of any balanced cut which can be obtained from $A \cup B$ by repeatedly taking (unique) greedy swaps, subject to not re-swapping nodes from a previous swap. Note that these greedy swaps need not increase the cut value, as long as they are the least decrease.

It remains to outline our reductions to these problems. Formally, if X is the $|E|k^2$-dimensional vector of payoff values for an n-player network coordination game on the graph $G = (V, E)$, we map these values to an instance of Local Max Cut on $G' = (V', E')$, by mapping X to the $|E'|$-dimensional vector of edge-weights W. The key property, as highlighted above, is that As highlighted above, we mainly need this to be a full-rank linear mapping.

We obtain the two following reductions:

Theorem 4.1. NetCoordNash with 2×2 payoff matrices admits a weak smoothness-preserving reduction to the local-max-cut problem. Furthermore, NetCoordNash with $k \times k$ matrices for general k admits a weak smoothness-preserving reduction to the local-max-bisection problem. For both results, an instance of NetCoordNash with a general or complete game graph reduces to an instance of local-max-cut/bisection on a general or complete graph, respectively.

Note, the first reduction, together with smoothed efficient algorithms for local-max-cut, gives alternate smoothed efficient algorithms for the $k = 2$ instance of Network-Coordination Nash, which was presented in the previous chapter. In particular, the recent local-max-cut result of [108] gives an $O(n^8)$ algorithm for finding Nash Equilibria when the game graph is complete, and furthermore, We can improve the exponent on the running time for general graphs to $O(\sqrt{\log n})$ due to [107].

For general k, the smoothed complexity of local-max-bisection is open, and so any conclusion on the complexity of NetCoordNash is conditional. The result is of interest, however, as it reduces games on n players and k strategies to graphs on $kn + 2$ vertices. Therefore, a smoothed-efficient algorithm for this case would eliminate the exponential dependence on k in the statements of Theorem 3.2.

4.1 SMOOTHNESS-PRESERVING REDUCTIONS

This section formally proves that strong and weak smoothness-preserving reductions do indeed allow to translate smoothed analysis results. The results follows almost by definition, modulo technicalities.

Lemma 4.1. Let $\mathcal{Q}$ be a search problem with (quasi-)polynomial smoothed complexity. Let $\mathcal{P}$ be a problem which admits a strong smoothness-preserving reduction to $\mathcal{Q}$, given by f_1, f_2, M, as in Definition 4.1. Then $\mathcal{P}$ has (quasi)polynomial smoothed complexity.

Proof. The algorithm for solving instances of $\mathcal{P}$ is as follows:

1. Perform the randomized reduction,
2. Run the smoothed-(quasi-)polynomial-time algorithm for $\mathcal{Q}$ on the reduced instance,
3. Compute the solution to the instance of $\mathcal{P}$ given the solution to the reduced problem.

By the definition of smoothness-preserving reductions and (quasi-)polynomial smoothed complexity, step 2. will always correctly solve the reduced instance in finite time, and therefore step 3. will output a correct solution to the instance of $\mathcal{P}$.

It remains then to show that the algorithm runs in polynomial time with high probability, which we do via Markov's inequality, to control for the effect of the newly introduced randomness. Let (D, X) be an arbitrary instance of $\mathcal{P}$ where D is fixed, and X is a random vector whose entries are independently distributed with density bound ϕ. Let R be a random vector also independently distributed with density bound $\phi' = poly(\phi, |D|, |X|)$. Without loss of generality, $\phi' \geq \phi$. Since the reduction is *strong*, and the matrices are assumed to have integer entries, and therefore the entry-wise densities of the rescaling $M \cdot (X \circ R)$ also has densities bounded by ϕ'.

Let $\mathcal{A}$ be the smoothed efficient algorithm for $\mathcal{Q}$. Thus, there exist constants $c, c' > 0$ such that on random input (C, Y) with density bound ϕ', $\mathcal{A}$ runs in time $(\phi'|C||Y|)^c$ with probability $1 - 1/|C|^{c'}$. By definition, $|f_1(D)| \leq poly(|D|, |X|)$. We wish to show that with $1 - 1/poly(|D|, |X|)$ over the randomness in X and R, the reduced instance given by $C := f_1(D)$ and $Y := M \cdot (X \circ R)$ will be solved by $\mathcal{A}$ in time $poly(\phi, |D|, |X|)$, or quasi-polynomial time. By the assumptions on the performance of $\mathcal{A}$ for instances of $\mathcal{Q}$, this holds by definition, since the entries of $M(X \circ R)$ are independently distributed, and ϕ', $|C|$, and $|Y|$ are polynomial in ϕ, $|D|$, and $|X|$. *Q.E.D.*

Corollary 4.1. Let $\mathcal{Q}$ be a search problem with (quasi-)polynomial smoothed complexity when the input is arbitrarily distributed with a bound on the joint density as in the statements of Proposition 3.1 and Lemma 3.1. Let $\mathcal{P}$ be a problem which admits a *weak* smoothness-preserving reduction to $\mathcal{Q}$, then $\mathcal{P}$ has (quasi-)polynomial smoothed complexity.

The proof of this corollary is identical to the above, combined with Proposition 3.1.

Corollary 4.2. Let $\mathcal{P}$ be a problem which admits a *weak* smoothness-preserving reduction to local-max-cut, then $\mathcal{P}$ has quasi-polynomial smoothed complexity. If it admits a *weak* reduction to local-max-cut on a complete graph, then it has polynomial smoothed complexity.

Proof. Following the common framework of Section 3.1, the proofs of the local-max-cut smoothed results from [31, 32] consist of applying Lemma 3.1 directly to the edge weights of

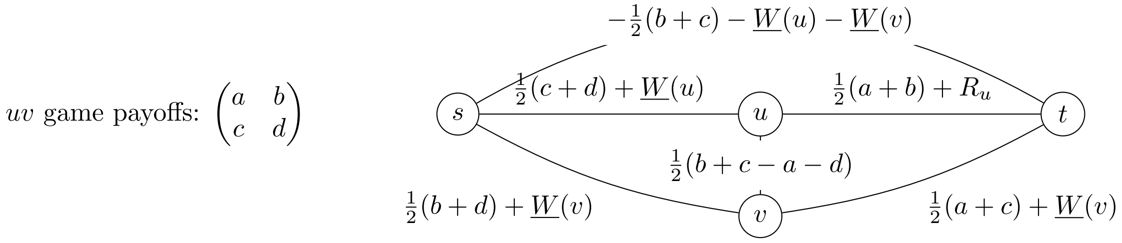
uv game payoffs: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$
$-\frac{1}{2}(b+c) - \underline{W}(u) - \underline{W}(v)$
$\frac{1}{2}(c+d) + \underline{W}(u)$
$\frac{1}{2}(a+b) + R_u$
s
u
t
$\frac{1}{2}(b+c-a-d)$
$\frac{1}{2}(b+d) + \underline{W}(v)$
v
$\frac{1}{2}(a+c) + \underline{W}(v)$

Figure 4.1: An game-gadget in the weak smoothness-preserving reduction from Network-Coordination Nash on 2×2 games, to Local Max Cut.

the graph, and finding bounds on the rank of the linear transformation. By Proposition 3.1, a weak reduction satisfies the conditions for the application of Lemma 3.1, and therefore the local-max-cut satisfies the conditions of the previous corollary, as desired. *Q.E.D.*

We note, as discussed in previous sections, that it would also have sufficed for X to have joint density bounded by $\phi^{|X|}$. Observe that if it were possible to weakly reduce k-NetCoordNash to local-max-cut, then this would imply a (quasi-)polynomial smoothed complexity for k-NetCoordNash, where the degree of the polynomial does not depend on k. Unfortunately, we only achieve a weak reduction to local-max-bisection, which we believe has similar smoothed complexity to local-max-cut, though this is not as of yet known. We leave this as an open problem.

4.2 REDUCTION FROM 2-NetCoordNash TO LOCAL-MAX-CUT

In this section, we give our first reduction from 2-NetCoordNash to local-max-cut, and show it satisfies the conditions of a smoothness-preserving reduction. We begin with an informal description of the reduction.

Let $G = (V, E)$ be the game graph, with payoff vector A. See Chapter 3 for the formal definitions of these problems. We will construct a weighted cut graph $H = (V', E')$ where $V' = V \cup \{s, t\}$, and E' is obtained from E by adding su and ut edges for all $u \in V$. We wish to select edge weights such that (1) every locally maximal cut is an s-t cut, and (2) the value of the cut (S, T) with $s \in S$ and $t \in T$ is equal to the total payoff of the game when $\sigma_u = 1$ if $u \in S$, and 2 if $u \in T$. Thus, changing a player's strategy is equivalent to flipping the vertex across the cut, and so solving for a local max cut is equivalent to solving for a local max of the game's potential function.

Figure 4.1 gives the edge weights for a small 2-player example which satisfies the above properties, with the payoff matrix given in the figure. The general construction, which is specified below, places a copy of this gadget for each of the game-edges in the network game, and sums the edge weights. Observe that this construction indeed has edge weights which are linear combinations of the payoff values. Furthermore, even if $\underline{W}(u) = \underline{W}(v) = 0$, cut values are equal to payoff values, and the maximal cuts are s-t cuts. The $\underline{W}(u)$ and $\underline{W}(v)$ values are added to increase the rank of the reduction matrix, which as discussed above, is the key property of these reductions. We show by induction on the number of players, later in this section, that the matrix has full rank, which implies that it is a valid reduction.

Observe that the cut graph is complete if and only if the game graph is, as all su and ut edges are present in the cut graph, and there is a uv edge in the cut graph whenever there is a uv game.

Formally defined reduction. We give here formally the reduction that was described above. Let $G = (V, E)$ be a game graph, with payoff vector $A \in [0.5, 1]^{4|E|}$. As the inputs are assumed to lie in $[-1, 1]$, this is without loss of generality since we can make this transformation while preserving all distributional assumptions, and at most quadrupling the probability density in each coordinate. We will let $A((u, i), (v, j))$ denote the payoff when the u player chooses strategy i, and v chooses j. As outlined above, we construct a cut graph $H = (V \cup \{s, t\}, E')$, where E' consists of the edges E over V, with an additional su and ut edge for all $u \in V$. We will define the edge weights below, where $w(u, v)$ denotes the weight of edge $uv \in E'$. Let $\Gamma(u)$ denote the neighbours of u in graph G, and $\underline{W}$ be a $|V|$-dimensional vector of extra randomness, assumed to be uniformly distributed in $[-1, -0.5]^{|V|}$. We set:

$$\begin{aligned}
w(u,v) &= \tfrac{1}{2}\Big(A((u,1)(v,2)) + A((u,2)(v,1)) \\
&\qquad\qquad - A((u,1)(v,1)) - A((u,2)(v,2))\Big) && \forall u, v \in V \quad \text{(4.2a)}\\
w(s,u) &= \sum_{v:uv\in E} \Big[\tfrac{1}{2}\big(A((u,2)(v,1)) + A((u,2)(v,2))\big) + \underline{W}(u)\Big] && \forall u \in V \quad \text{(4.2b)}\\
w(u,t) &= \sum_{v:uv\in E} \Big[\tfrac{1}{2}\big(A((u,1)(v,1)) + A((u,1)(v,2))\big) + \underline{W}(u)\Big] && \forall u \in V \quad \text{(4.2c)}\\
w(s,t) &= (-1)\cdot \sum_{uv\in E} \Big[\tfrac{1}{2}\big(A((u,1)(v,2)) + A((u,2)(v,1))\big) + \underline{W}(u) + \underline{W}(v)\Big] && \text{(4.2d)}
\end{aligned}$$

Observe that the above are linear combinations of the input values, and the coefficients are $O(|E|)$-sized integer multiples of $\eta = \frac{1}{2}$.

Lemma 4.2. The above construction satisfies the following conditions:

1. Cut values of s-t cuts are equal to the potential function of the associated strategy profiles,
2. All locally maximal cuts are s-t cuts,
3. The construction is full-rank

Proof. Let (S, T) be a cut such that $s \in S$ and $t \in T$. We will do a quick case analysis for each payoff term. Note first that the $\underline{W}$ terms get cancelled by the st edge, since they must appear exactly once for s or t. We say u is "playing i according to the cut" if $u \in S$ when $i = 1$ or if $u \in T$ when $i = 2$. Suppose u is playing i and v is playing j, then the $A((u, i)(v, j))$ term is added with total weight 1 in the su, sv, ut, and vt edges, and if $i \neq j$, it is also added and removed in the uv and st edges, respectively, so it appears with total weight 1. If u is playing i but v is not playing j, then $A((u, i)(v, j))$ is added with weight $\frac{1}{2}$ in the su and ut edges, and it is subtracted with weight $\frac{1}{2}$ in uv if $i = j$, or st if $i \neq j$. Finally if u is not playing i and v is not playing j, then the term does not appear if $i = j$. Thus, the only terms that appear are the correct ones, and they appear with weight 1.

Condition 1. By inspection, we can see that in the construction of Figure 4.1, the value of the s-t correctly evaluates to the payoff to both players u and v on the uv game edge. Furthermore, the edge weights given in the construction above are exactly equal to summing the gadgets in the figure over every game edge, and therefore, the value of the cut will be the sum of game edge payoffs, namely, the value of the potential.

Condition 2. To show condition 2, first recall that all entries of $\underline{W}$ lie in $[-1, -0.5]$, as it is uniformly distributed in $[-1, -0.5]^{|V|}$. As observed above, in any s-t cut, the $\underline{W}$ terms are cancelled out, and by our assumption on the payoff values, we have cut values between $\frac{1}{2}|E|$ and $|E|$. Consider any cut (S, T) where $s, t \in S$. Then for every $u \in T$, we are contributing $+2\underline{W}(u) \cdot |\Gamma(u)|$ to the cut from su and ut edges, and at most four A terms for each uv edge with weight $\frac{1}{2}$ each, so the cut value must be non-positive, and switching node t to the other side will improve the cut value. Therefore, all locally maximal cuts are s-t cuts.

Condition 3. To show condition 3, we explicitly write out the matrix and show it has full row-rank by induction on the number of players. Let Id_n denote the $n \times n$ identity matrix, $\mathbf{1}_n$ denote the $1 \times n$ row of 1's, $\boldsymbol{\gamma}$ denote the vector of $|\Gamma(u)|$ values, and $\mathbf{\Gamma}$ be the diagonal matrix with diagonal $\boldsymbol{\gamma}$. Let $B_i \in \{0, 1\}^{|V| \times 4|E|}$ denote the payoff-node incidence matrix of

G, where the 1's are in the$(u,(u,i)(v,\cdot))$ entries. We have

$$\left(\begin{array}{c} \vdots \\ w(u,v) \\ \vdots \\ \hline \vdots \\ w(u,t) \\ \vdots \\ \hline w(s,t) \\ \hline \vdots \\ w(s,u) \\ \vdots \end{array}\right) = \frac{1}{2}\left(\begin{array}{c|c} -\mathsf{Id}\otimes(-1,+1,+1,-1) & \mathbf{0} \\ \hline B_1 & 2\boldsymbol{\Gamma} \\ \hline \mathbf{1}\otimes(0,-1,-1,0) & -2\boldsymbol{\gamma} \\ \hline B_2 & 2\boldsymbol{\Gamma} \end{array}\right)\left(\begin{array}{c} \vdots \\ A((u,1)(v,1)) \\ A((u,1)(v,2)) \\ A((u,2)(v,1)) \\ A((u,2)(v,2)) \\ \vdots \\ \hline \vdots \\ \underline{W}(u) \\ \vdots \end{array}\right) \tag{4.3}$$

Note that B_1 and B_2 have disjoint support, and B_1+B_2 is an edge-node incidence matrix, tensored with $(1,1,1,1)$. We will show by induction on $|V|$ that this matrix is full-rank. For $n=2$, the matrix is explicitly

$$\begin{pmatrix} -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 2 \\ 0 & -1 & -1 & 0 & -2 & -2 \\ 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{pmatrix} \tag{4.4}$$

which can be verified to have rank 6. Suppose then that the rank property holds for $|V|=n-1$, and we introduce a new node v. Then the new edges introduced to the matrix, restricted to the columns for edges with v and $\underline{W}(v)$, are of the form

$$\left(\begin{array}{ccccccccc|c} -1 & +1 & +1 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & +1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline\hline +1 & +1 & 0 & 0 & +1 & +1 & 0 & 0 & \cdots & -2\gamma_v \\ 0 & 0 & +1 & +1 & 0 & 0 & +1 & +1 & \cdots & -2\gamma_v \end{array}\right) \tag{4.5}$$

which has full row-rank. Since the entries in the omitted columns are all 0, then we have a block-upper-triangular matrix where the first diagonal block is the matrix for the graph

$V \setminus v$, and the second diagonal block is this above matrix. Therefore, by induction, the matrix has full row rank, as desired. *Q.E.D.*

4.3 REDUCTION FROM K-NetCoordNash TO LOCAL-MAX-BISECTION

We present in this section the final reduction from k-NetCoordNash to Local-Max-Bisection, showing the second part of Theorem 4.1. We begin with a description of the reduction.

Let $G = (V, E)$ be the game graph again, and we will construct a weighted cut graph $H = (V', E')$. V' is given by the set of all (player,strategy) pairs $V \times [k]$, with $n(k-2)+2$ extra vertices s_0, s_1, $\ldots$, $s_{n(k-2)}, t$. The goal is to have all the s_i's stay on the same side of the cut, and act as a "large" s node, while maintaining balanced cuts. E' is obtained as follows: for every node (u, i), we add an $\{s_z, (u, i)\}$ for all z, and a $\{(u, i), t\}$ edge; for every $u \in V$ and $i \neq j$, we add a $\{(u, i), (u, j)\}$ edge, and for every $uv \in E$ and $i, j \in [k]$, we add a $\{(u, i), (v, j)\}$ edge. Finally, for every $0 \leq z < z' \leq n(k-2)$, there is an $\{s_z, s_{z'}\}$ edge.

Definition 4.3 (Valid Cuts, Bisections)**.** Call a cut (S, T) *valid* if $s_z \in S$ for all z, $t \in T$, and S contains exactly one (u, i) node for all $u \in V$. Note that such a cut has exactly $n(k-1)+1$ nodes on each side, and is therefore a balanced cut, *i.e.* a bisection.

To each valid bisection is associated the natural strategy profile where σ_u is given by the unique (u, i) node in S. We wish to choose edge weights such that (1) all locally maximal bisections are valid, and (2) the cut value is equal to $\Phi(\boldsymbol{\sigma})$, which is the potential function which was introduced in Chapter 3. (1) will be achieved by making the $\{(u, i), (u, j)\}$ edges bad, and the $\{s_z, (u, i)\}$ edges good, using the extra randomness available. This respectively ensures that it is always in our interest to have a small number of $(u, \cdot)$ nodes in S, but not none.

As above, we will introduce extra randomness to the edge weights to ensure that M is full-rank. In this case, we will show M is full rank by arguing that it is upper-triangular after basic row operations. The cut graph is again complete if and only if the game graph is, and therefore, we will have shown the second part of Theorem 4.1.

Extra Randomness. We wish to choose edge weights such that, as in the previous reduction, (a) the cut value of a valid bisection is equal to $\Phi(\boldsymbol{\sigma}(S, T))$, (b) all locally maximal bisections are valid cuts, and (c) the construction is full-rank. As in the previous section, we assume without loss of generality that the entries of A are supported in $[2.5, 3]$. Let

- $\underline{W}^z(u, i) \sim \mathcal{U}[-\frac{3}{4}, -\frac{1}{2}]$, *i.i.d.* for all $0 \leq z \leq n(k-2)$, $u \in V$, and $i \in [k]$. (4.6a)

- $\underline{R}(u,ij) \sim \mathcal{U}[-1,-\frac{1}{2}]$ *i.i.d.* for all $u \in V$ and $i \neq j \in [k]$. (4.6b)
- $\underline{Y}(u,i) \sim \mathcal{U}[2, 2.5]$ *i.i.d.* for all $u \in V$ and $i \in [k]$. (4.6c)
- $\underline{A}_0^z \sim \mathcal{U}[0, \frac{1}{2nk})$ *i.i.d.* for all $0 \leq z \leq n(k-2)$. (4.6d)

Note that the $\underline{A}_0^z$'s have density $2nk$, and all other variables have constant density. Define $\underline{A}_0 := \sum_{z=0}^{n(k-2)} \underline{A}_0^z$. Let $S_0 := \{s_0, \ldots, s_{n(k-2)}\}$, and for any valid cut (S,T), let $\psi(S) := \Phi(\boldsymbol{\sigma}(S,T))$. We will also extend ψ to be defined on invalid cuts. If there is no $(u,\cdot)$ node in S, say that $\sigma_u(S) = 0$, and in the definition of $\psi(S)$, let $A((u,i)(v,0)) := \underline{Y}(u,i)$ for all $u,v \in V$ and $i \in [k]$, and let $A((u,0)(v,0)) := \underline{A}_0$ for all $u,v \in V$. Let $\delta(S)$ denote the cut value of $(S, V \setminus S)$. We will construct edge weights with the following properties:

(i) For every valid (S,T), $\delta(S) = \psi(S)$. (From above) (4.7a)

(ii) For every $u \in V$, and $i \neq j \in [k]$, $\delta(S_0 \cup \{(u,i),(u,j)\}) = 2\underline{R}(u,ij)$. (4.7b)

(iii) For every $u \in V$, $i \in [k]$, and $0 \leq z \leq n(k-2)$, $w(s_z,(u,i)) = \underline{W}^z(u,i)$. (4.7c)

Furthermore, we simply assume that for all $0 \leq z < z' \leq n(k-2)$, the weight of the $s_z s_{z'}$ edge is given by the random variable $\overline{W}(z,z')$, which are distributed *i.i.d.* uniformly along $[-1,-0.5]$. The correctness of the reduction will be proved using two lemmas, established using the following claim.

Claim 4.2. Condition (i) is satisfied if (a) $\delta(S_0 \cup \{(u,i)\}) = \psi(S_0 \cup \{(u,i)\})$ for all players u and $1 \leq i \leq k$, and (b) $w((u,i),(v,j)) = \underline{Y}(u,i) - \underline{Y}(v,j) - A((u,i)(v,j)) - \underline{A}_0$.

Proof. Let $S := S_0 \cup \{(u_1,i_1), \ldots, (u_\ell,i_\ell)\}$. We begin by showing the following:

$$\begin{aligned}\psi(S) = \Big[\sum_{j=1}^{\ell} \psi(S_0 \cup \{(u_j,i_j)\})\Big] - (\ell-1)\psi(S_0) \\ - \sum_{\substack{(u,i),(v,j)\in S \\ uv \in E}} 2\left[A((u,i)(v,0)) + A((u,0)(v,j)) - \underline{A}_0 - A((u,i)(v,j))\right]\end{aligned} \tag{4.8}$$

$$\delta(S) = \Big[\sum_{j=1}^{\ell} \delta(S_0 \cup \{(u_j,i_j)\})\Big] - (\ell-1)\delta(S_0) - \sum_{\substack{(u,i),(v,j)\in S \\ uv \in E}} 2w((u,i),(v,j)) \tag{4.9}$$

For (4.8), note first if there is no uv edge in the game graph, then $A((u,\cdot)(v,\cdot))$ does not appear on either side of the equality, and we may restrict our attention to pairs which form game edges. Now, for every v and w which do not appear in S, the left-hand-side has $2A((v,0)(w,0))$, and the right-hand-side has $2(\ell-(\ell-1))\underline{A}_0$ from the first line. If u appears with strategy i, and v does not appear in S, then the left-hand-side has $2A((u,i)(v,0))$, and the right-hand-side has $2A((u,i)(v,0))$ from the $\psi(S_0 \cup \{(u,i)\})$ term. If u appears with strategy i, and v appears with strategy j, then the left-hand-side has $2A((u,i)(v,j))$, and the right-hand-side has $2A((u,i)(v,0))$ and $2A((u,0)(v,j))$ from the $\psi(S_0 \cup \{(u,i)\})$ and $\psi(S_0 \cup \{(v,j)\})$ terms which are canceled out by the second line, $2(\ell - 2 - (\ell - 1))\underline{A}_0$ terms from the first line which is canceled out by the second line, and the term $2A((u,i)(v,j))$ from the second line. A similar argument shows the validity of (4.9).

Since condition (i) requires that $\pi(S_0 \cup \{u,i\}) = \delta(S_0 \cup \{(u,i)\})$, this is necessary, and along with $\delta(S_0) = \psi(S_0)$ from (4.12), the above analysis shows it is sufficient. We are required to set $w((u,i)(v,j)) = A((u,i)(v,0)) + A((u,0)(v,j)) - \underline{A}_0 - A((u,i)(v,j))$, which is equal to $\underline{Y}(u,i) + \underline{Y}(v,j) - \underline{A}_0 - A((u,i)(v,j))$. Observe that this is supported on the interval $[1, 2.5]$. *Q.E.D.*

Lemma 4.3. There exist edge weights w which satisfy conditions (i), (ii), and (iii). Moreover, these are a full-rank, square, integer-valued, linear combinations of the entries of A, $\underline{Y}(u,i)$, $\underline{A}_0^z$, $\underline{R}(u,ij)$, $\underline{W}^z(u,i)$, and $\overline{W}(z,z')$.

Proof. We may ignore the rows indexed by $s_z s_{z'}$, as they depend only on the $\overline{W}$ values, and these values do not appear anywhere else, so they are independent, and do not affect the dependence of other rows. Next, we derive edge weights such that the conditions of Claim 4.2 hold, using the following system:

$$w(s_z,(u,i)) = \underline{W}^z(u,i) \qquad \text{(by def'n)} \tag{4.10}$$

$$w((u,i)(v,j)) = \underline{Y}(u,i) + \underline{Y}(v,j) - A((u,i)(v,j)) - \underline{A}_0 \tag{4.11}$$

$$\psi(S_0) = \delta(S_0) = \sum_{z=0}^{n(k-2)} w(s_z,t) + \sum_{u \in V} \sum_{i=1}^{k} \sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) \tag{4.12}$$

$$\implies \sum_{z=0}^{n(k-2)} w(s_z,t) = \psi(S_0) - \sum_{u \in V} \sum_{i=1}^{k} \sum_{z=0}^{n(k-2)} \underline{W}^z(u,i)$$

$$\forall z \text{ we choose} : w(s_z,t) = \psi(S_0) - \sum_{u \in V} \sum_{i=1}^{k} \underline{W}^z(u,i) \tag{4.13}$$

$$\begin{aligned} 2\underline{R}(u,ij) &= \delta(S_0 \cup \{(u,i),(u,j)\}) \\ &= \delta(S_0 \cup \{(u,i)\}) + \delta(S_0 \cup \{(u,j)\}) - \delta(S_0) - 2w((u,i)(u,j)) \\ \implies w((u,i)(u,j)) &= \tfrac{1}{2}\big(\, \psi(S_0 \cup \{(u,i)\}) + \psi(S_0 \cup \{(u,j)\}) - \psi(S_0) - 2\underline{R}(u,i,j)\,\big) \end{aligned} \tag{4.14}$$

$$\begin{aligned}
&\psi(S_0 \cup \{(u,i)\}) = \\
&\delta(S_0 \cup \{(u,i)\}) = \sum_{z=0}^{n(k-2)} w(s_z,t) + w((u,i),t) + \sum_{(v,j)\neq(u,i)} \Big[\textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(v,j) + w((u,i)(v,j)) \Big] \\
&\implies w((u,i),t) = \psi(S_0 \cup \{(u,i)\}) - \sum_{z=0}^{n(k-2)} w(s_z,t) - \sum_{(u,i)\neq(v,j)} \Big[\sum_{z=0}^{n(k-2)} \underline{W}^z(v,j) + w((u,i)(v,j)) \Big] \\
&\qquad = \psi(S_0 \cup \{(u,i)\}) - \psi(S_0) + \textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) - \sum_{(u,i)\neq(v,j)} w((u,i)(v,j))
\end{aligned} \tag{4.15}$$

We observe first that (4.15) contains as terms the previous numbered equations. Thus, it suffices to perform simple row-elimination to get $w((u,i),t) = \psi(S_0 \cup \{u,i\}) - \sum_{v,j} \sum_z \underline{W}^z(v,j)$. Now, let $G = (V,E)$ be the underlying game graph, and let $d(u)$ be the degree of u in G. Then $\psi(S_0) = 2|E|\underline{A}_0$, and $\psi(S_0 \cup \{(u,i)\}) = \psi(S_0) + 2d(u)[\underline{Y}(u,i) - \underline{A}_0]$. Finally, we have, letting $Q := n \cdot (k-2) + 1$,

$$\begin{pmatrix} w((u,i)(v,j)) \\ \vdots \\ \hline w((u,i)(u,j)) \\ \vdots \\ \hline \widehat{w}((u,i),t) \\ \vdots \\ \hline w(s_z,t) \\ \vdots \\ \hline w(s,(u,i)) \\ \vdots \end{pmatrix} = \left(\begin{array}{c|c|c|c|c} -\mathsf{Id}_{|E|k^2} & \mathbf{0} & * & -\mathbf{1}_{|E|k^2 \times Q} & \mathbf{0} \\ \hline \mathbf{0} & -\mathsf{Id}_{n\binom{k}{2}} & * & * & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & 2d(u)\mathsf{Id}_{nk} & * & \mathsf{Id} \otimes \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & 2|E| \cdot \mathsf{Id}_Q & -\mathbf{1} \otimes \mathsf{Id} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathsf{Id} \otimes \mathbf{1} \end{array}\right) \begin{pmatrix} A_{uv}(i,j) \\ \vdots \\ \hline \underline{R}(u,ij) \\ \vdots \\ \hline \underline{Y}(u,i) \\ \vdots \\ \hline \underline{A}_0^z \\ \vdots \\ \hline \underline{W}^z(u,i) \\ \vdots \end{pmatrix} \tag{4.16}$$

Where $\otimes$ denotes the tensor product, namely,

$$\mathsf{Id} \otimes \mathbf{1} = \begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \qquad \mathbf{1} \otimes \mathsf{Id} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & \cdots \end{pmatrix} \tag{4.17}$$

It is easy to check that the $*$ values are integral, since the ψ values must be even combinations of the A values. Therefore, after the row-operations leading to $\widehat{w}((u,i),t)$ values, the matrix is upper-triangular, which implies that the system is full-rank, square, and integral, as desired. *Q.E.D.*

Lemma 4.4. If conditions (i), (ii), and (iii) are satisfied, then all local-max-bisections are valid cuts, and their associated strategy profiles are Nash equilibria.

Proof. Recall that we have assumed that $0.5 \leq A_{uv}(i,j) \leq 1$ for all edges uv and for all $1 \leq i,j \leq k$, that $0 \leq \underline{A}_0, \underline{Y}(u,i) < 0.5$ for all players u and $1 \leq i \leq k$ (Since $\underline{A}_0 = \sum_{z=0}^{n(k-2)} \underline{A}_0^z$, and the latter is contained in $[0, \frac{1}{2nk}))$, and that $-1 \leq \underline{R}(u,ij), \underline{W}^z(u,i), \overline{W}(z,z') < -0.5$ for all players u, $1 \leq i < j \leq k$, $0 \leq z < z' \leq n(k-2)$.

We will show that from any non-valid cut, there will be a single flip operation towards a valid cut which improves the total cut value, then argue that they may be paired up into swap operations. Fix a bisection (S,T), and consider the following cases:

Case I: $t \in T$, and S contains at least half the s_z's, but $s_z \in T$ for some z

Let s_z in T, we argue that $\delta(S \cup \{s_z\}) - \delta(S) > 0$. The two cuts may only differ on edges incident to s_z. The positive term includes $w(s_z,t)$, $\underline{W}^z(u,i)$ for all $(u,i) \notin S$, and the $\overline{W}(z,z')$ for all $s_{z'} \in T$, and the negative term includes $\underline{W}^z(u,i)$ for all $(u,i) \in S$ and the $\overline{W}(z,z')$ for all $s_{z'} \in S$. However, we know from (4.13) that $w(s_z,t) = \psi(S_0) - \sum_{(u,i)} \underline{W}^z(u,i)$. Therefore, we get

$$\delta(S \cup \{s_z\}) - \delta(S) \quad = \quad 2|E| \underbrace{\underline{A}_0}_{\geq 0} - 2\sum_{(u,i)\in S} \underline{W}^z(u,i) + \sum_{z:s_z \in T} \overline{W}(z,z') - \sum_{z:s_z\in S} \overline{W}(z,z') \quad (4.18)$$

Now, note that, since these are bisections, $|\{(u,i) : (u,i) \in S\}| + |\{z' : s_{z'} \in S\}| = n(k-1)+1$, so

$$-2\sum_{(u,i)\in S} \underline{W}^z(u,i) - \sum_{z:s_z\in S} \overline{W}(z,z') \quad \geq \quad \tfrac{1}{2}(n(k-1)+1) \qquad (4.19)$$

and $\sum_{z:s_z\in T} \overline{W}(z,z') \geq -\frac{1}{2}(n(k-2)+1)$, since we have assumed more than half lie in S, so therefore the above sum is non-negative, and moving s_z into S was an improvement.

Case II. $t \in T$ and S contains fewer than half the s_z's.

We wish to show that in this case, $\delta(S \cup \{t\}) - \delta(\{S\}) \geq 0$. Since there are fewer than $\frac{1}{2}(n(k-2)+1)$ s_z nodes in S, there must be at least $\frac{1}{2}(nk-1)$ (u,i) nodes on the S side. Without loss of generality, we may assume n or k is even, and so at least half of the (u,i) nodes are in S. Therefore, it would suffice to show that $w(t,(u,i)) < 0$, and $w(s_z,t) \geq 0$, so t would benefit from moving to the side with fewer s_z and more (u,i) nodes. We have $w(s_z,t) = \psi(S_0) - \sum_{v\in V}\sum_{i=1}^{k} \underline{W}^z(u,i)$. Since $\underline{W}^z(u,i) \leq 0$, and $\psi(S_0) = 2|E|k^2 \cdot \underline{A}_0 \geq 0$, we have that $w(s_z, t) \geq 0$. Conversely,

$$\begin{aligned}
w(t,(u,i)) &= \psi(S_0 \cup \{(u,i)\}) - \psi(S_0) + \textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) - \sum_{(u,i)\neq(v,j)} w((u,i)(v,j)) \\
&\leq \sum_{v:uv\in E} \underline{Y}(u,i) - \tfrac{1}{2}n(k-2) - \tfrac{1}{2} - k \sum_{v:uv\in E} (\tfrac{1}{2} + \tfrac{1}{2nk}) - \sum_{j\neq i} w((u,i)(u,j)) \\
&< d(u) \cdot \tfrac{5}{2} - \tfrac{k}{2}d(u) - \tfrac{1}{2}n(k-2) \\
&\leq 0 \qquad (4.20)
\end{aligned}$$

For $k \geq 3$ and $n \geq 2k$. This concludes the proof of the claim.

Case III. $t \in T$, S contains at least half of the s_z's, and also (u,i) and (u,j) for some u and $i \neq j$. Recall from (4.15) that

$$\begin{aligned}
w((u,i),t) &= \psi(S_0 \cup \{(u,i)\}) - \psi(S_0) + \textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) - \sum_{(u,i)\neq(v,j)} w((u,i)(v,j)) \\
&\geq \textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) - \sum_{(u,i)\neq(v,j)} w((u,i)(v,j)) \qquad (4.21)
\end{aligned}$$

Now,

$$\begin{aligned}
&\delta(S) - \delta(S \setminus \{(u,j)\}) \\
&\quad= -w((u,i),t) + \sum_{s_z\in S} \underline{W}^z(u,j) - \sum_{s_z\in T} \underline{W}^z(u,j) + \\
&\qquad\quad + \sum_{(v,i)\neq(u,j)\in S} w((u,j)(v,i)) - \sum_{(v,i)\in T} w((u,j)(v,i)) \\
&\quad= -2\sum_{s_z\in T} \underline{W}^z(u,j) - 2\sum_{(v,i)\neq(u,j)\in S} w((u,j)(v,i)) \\
&\quad\leq \tfrac{3}{4}|S_0 \cap T| - |\{(v,i) \neq (u,j) \in S : v \neq u\}| - \sum_{i\neq j:(u,i)\in S} w((u,j)(u,i)) \qquad (4.22)
\end{aligned}$$

Note that, since the cut is a bisection, we have $|\{(v,i) \neq (u,j) \in S : v \neq u\}| = n-1+|S_0 \cap T|$, and also $w((u,i)(u,j)) \geq 0$. Therefore the sum overall is ≥ 0 if $n \geq k-1$, as desired.

Therefore, from any non-valid cut, it is always an improvement to (1) ensure that t is opposite the majority of s_z nodes by swapping it with some redundant (u,i) node, which must exist, (2) ensure that all s_z nodes are on the same side by swapping them with redundant (u,i) nodes, which must exist. If all the s_z are on one side, and t on the other, there must be exactly n nodes of the form (u,i). If some player appears twice, then another player does not appear, and it is in our interest to swap the redundant node with any node of the missing player. Entering the node of the missing player into the cut is an improvement because all $\underline{Y}$ and $\underline{A}_0$ values are smaller than all in-game payoffs, with probability 1. Therefore, all locally-maximal bisections are valid.

Note that we have required $\underline{A}_0^z$ to be distributed over the interval $[0, \frac{1}{2nk})$, so the density bound must be at least $2nk$, which is polynomial. *Q.E.D.*

Claim 4.2, and Lemmas 4.3 and 4.4 prove the correctness of the reduction, thus establishing the following theorem.

Theorem 4.3. There is a smoothness preserving reduction from k-NetCoordNash to Local-Max-Bisection. The reduction maps k-NetCoordNash instances defined on complete graphs to Local-Max-Bisection instances on complete graphs.

CHAPTER 5: THE PANDORA'S BOX PROBLEM WITH ORDER CONSTRAINTS[1]

In this chapter, we provide algorithms and hardness results for an order-constrained variant of the *Pandora's box problem*, introduced by Weitzman [64]. The Pandora's box problem models the problem of choosing one of many random-valued alternatives, when querying value (*i.e.* evaluation) has a cost. We introduce constraints on the order in which the values can be queried, and give computational results depending on the type of order constraints.

Formally, a player is presented with various alternatives modeled by a set of boxes $B = \{b_1, \ldots, b_n\}$, where box b_i costs c_i to open, and has random payoff X_i, whose distribution is known, but its value is initially unknown. Opening a box reveals this value, *i.e.* samples the distribution. A strategy π for the player is a rule which determines adaptively whether to terminate the search, and if not, which box to open next. The goal is to choose a strategy π which maximizes, in expectation, the following objective:

$$\max_{i \in S(\pi)} X_i - \sum_{i \in S(\pi)} c_i, \tag{5.1}$$

where $S(\pi)$ is the set of boxes opened by strategy π. Only *one* reward can be kept in the end, but the player must pay for all the opened boxes.

We enrich the classical Pandora's Box Problem by adding restrictions on the order in which boxes can be opened or evaluated, by precedence constraints. Formally, the order constraints are given as a directed, acyclic graph $G = (V, E)$ where each vertex is labelled by one of the boxes, $V = B$, and a box can be opened if at least one of its parents has been opened, or if it is a source. We show in this chapter that when the order constraint graph is a rooted tree or forest of rooted trees, then optimal search procedures may be found. Furthermore, we investigate more general constraints, where we show hardness of approximation, and give an approximation algorithm in a sepcial case.

In the original work by Weitzman [64], it was shown that despite the broad range of search strategies available, the solution to the unconstrained problem boils down to a simple strategy: Each box is assigned a *reservation value* ζ_i satisfying the equation $\mathbb{E}\left[(X_i - \zeta_i)_+\right] = c_i$, where $(X_i - \zeta_i)_+ := \max\{X_i - \zeta_i, 0\}$ denotes the positive part of $X_i - \zeta_i$. ζ_i is precisely the value that the player would need to have collected in order to be indifferent between opening b_i or not if it were the last box. The reservation value is used as a proxy for the value of b_i in the exploration procedure. Weitzman showed that the optimal strategy is to

[1] This chapter is based on collaboration with Federico Fusco, Philip Lazos, and Stefano Leonardi [96].

greedily open the boxes in descending order of reservation value, and to stop when there is no box left or when the maximum reward seen in the past is greater than the reservation value of the next box. Note that the order of exploration is not adaptive, but the stopping time is.

5.1 OVERVIEW

In what follows, *polynomial time* means polynomial in the number n of boxes and the largest achievable reward M. This implicitly suggests that the random variables are bounded, discrete and have support size polynomial in n and M. These assumptions are not restrictive: extending the techniques of [113], we show in Section 5.6 that a polynomial number of samples suffices to achieve an ϵ additive-approximate solution for the Pandora's Box Problem with order constraints. Moreover, these bounds are tight for the Tree-Constrained Pandora's Box Problem. We further note that this suggests the methods in this paper are robust to imprecise knowledge of the distribution of the rewards on the boxes.

The Tree-Constrained Pandora's Box Problem. In Section 5.3, we consider the Tree case, where we present an optimal strategy with a nice structure analogous to Weitzman's.

Theorem 5.1 (Restatement of Theorem 5.7)**.** When order constraints are given by a rooted tree over the boxes, there exists an optimal-in-expectation strategy of the following form: first, label each box with a "threshold" — an order-aware analog of Weitzman's reservation value. Then:

- From the boxes that can be opened next, choose the one with the largest threshold.
- Terminate if the max observed value exceeds this threshold, otherwise open the box and repeat.

Furthermore, these thresholds can be computed in polynomial time.

Notice that threshold strategies like the one described in the previous Theorem are simple and intuitive, and enforce the desirable property that the order of the exploration is fixed up to tie-breaking. Furthermore, it is surprising that an optimal strategy has this form, since we show below that this need not hold for general constraints. In the Tree-Constrained Pandora's Box Problem, the definition of these thresholds addresses the challenges of depth, and breadth: In considering depth, the value of a box is not only given by its reward and cost, but also from the possibilities its opening makes accessible. This effect propagates level after level, as even the deepest of the leaves can influence the decision to open the root. Conversely, in considering breadth, the order in which the boxes are opened matters. This is the original contribution of Weitzman. In our setting, it is difficult to model and optimize the interplay between different explored branches of a tree, as distant directions of exploration must be compared at every time step.

To overcome these difficulties, the first step is to generalize the reservation values used by Weitzman to the setting where the boxes are constrained to be opened in a fixed order $[b_1, b_2, \ldots, b_n]$. Notice that these values must take into account the *future* as well as the present. Our solution consists in defining the threshold value of the generic box i according to a random *stopping time* $\tau^*(y, i)$, which indicates the last box that will be opened *playing optimally* given that the player has already found reward y and is in front of box i. We call this threshold value z_i and it relates to the original definition as follows:

$$\mathbb{E}\left[(X_i - \zeta_i)_+ - c_i\right] = 0 \quad \longleftrightarrow \quad \mathbb{E}\left[\left(\max_{j=i}^{\tau^*(z_i,i)} X_j - z_i\right)_+ - \sum_{j=i}^{\tau^*(z_i,i)} c_j\right] = 0. \qquad (5.2)$$

From these stopping times and threshold values we can infer that certain consecutive chunks of boxes are essentially treated as large, collective, boxes. We refer to these as *macroboxes.* If the algorithm decides to "open" one of those macroboxes, the exploration will either stop inside of it, captured by the random τ^* stopping time, or will make a decision to open the next one after having exhausted the first. Moreover, these threshold values can be computed in polynomial time by a dynamic programming procedure.

For a single line, this reasoning may seem straightforward, but this property still holds when the constraints consist of a union of disjoint parallel lines. A naïve dynamic program would not be effective, as the state space is exponential in the number of lines. However, the threshold strategy which uses the reservation values computed for each line independently is still optimal: the algorithm will always enter the best available macrobox and either terminate search inside of it, or move on to another one, possibly from a different line. Surprisingly, the same approach works for *trees and forests.* Proceeding from leaves to the root it is possible to *linearize* the trees and use the definition of reservation value to induce a threshold strategy which is indeed optimal.

Impossibility and hardness results. Unfortunately, as shown in Section 5.5, this method does not extend to slightly more general constraint structures. We show that it is NP-hard to approximate an optimal solution to the problem with certain matroid constraints or more general order constraints. More precisely we show the following.

Theorem 5.2 (Restatement of the results in Section 5.5)**.** Consider the Pandora's Box Problem with order constraints when either (i) a matroid constraint is added to the tree constraint or (ii) the tree constraint is generalized to a DAG. In either case, it is NP-hard to find a 0.9997-approximately optimal solution, and furthermore, the optimal solution need not have a fixed order of exploration.

We finally remark that proving hardness of approximation for stochastic problems needs to address the effects of the randomness on the objective and the search trajectory, and as such is challenging. Moreover, the structure of the $\max_S X_i - \sum_S c_i$ objective makes optimal *random* solutions difficult to "hide", in a standard combinatorial sense.

Approximation results. The hardness result above, along with recent work on modified versions of the Pandora's Box Problem, such as [114, 115, 116, 117, 118], motivates the study in Section 5.4 of approximation algorithms for the more general order constraints. Of the above citations, the ones closest to our setting are [115, 116], where the author reduces the Pandora's Box Problem in the presence of downwards-closed constraints to adaptive maximization of non-negative submodular functions. The key concept in their work is the *adaptivity gap*, i.e. the ratio between the best adaptive solution and the best non-adaptive one for this new problem. We similarly show the following.

Theorem 5.3 (Restatement of Theorem 5.9)**.** Consider the Pandora's Box Problem with constraints modeled by some prefix closed family — a generalization of order constraints, defined in Section 7.1.1. For every adaptive strategy π, there exists a non-adaptive strategy, i.e. a feasible set S, such that the following holds:

$$\mathbb{E}\left[\max_{i\in S} X_i - \sum_{i\in S} c_i\right] \geq \tfrac{1}{2}\mathbb{E}\left[\max_{i\in S(\pi)} X_i\right] - \mathbb{E}\left[\sum_{i\in S(\pi)} c_i\right] \tag{5.3}$$

A thorough discussion about the choice of benchmark, *i.e.* fraction of expected reward minus the entire costs, is presented in Section 5.4.1.

Theorem 5.3 effectively reduces the problem of approximating adaptive strategies to the problem of selecting optimal non-adaptive sets. It should be noted, however, that in full generality of matroid constraints or precedence constraints, even this non-adaptive problem could be intractable. We therefore follow an alternative approach. We show that there exists a particular *adaptive* strategy whose performance is better than that of every non-adaptive set, simultaneously.

Theorem 5.4 (Restatement of Theorem 5.11)**.** Consider the Tree-Constrained Pandora's Box Problem with some further downwards-closed constraints, given by generalized knapsack constraints, or any "sufficiently oblivious" matroid constraint (as defined in Section 5.4). There exist an adaptive strategy $\hat{\pi}$ that can be computed efficiently and such that for any fixed set S, $\hat{\pi}$ performs better in expectation than non-adaptively opening S.

5.1.1 Related work.

As discussed above, the starting point for this theory is the seminal work by [64],which was at the time a generalization of previous results for special cases, namely [119, 120]. In the following years, the similarity between this problem and the multi-armed bandits setting was highlighted, introducing the notion of Gittins index to the Pandora's Box Problem, (e.g. [121, 122]). Indeed the reservation value of the classical Pandora's Box Problem is a version of Gittins index, as shown by [123]. [124], and in particular [125], deal with a similar problem, the branching bandit process. The branching process resembles the Tree-Constrained Pandora's Box Problem, though in that model the process does not terminate and the revenue is measured as an infinite-horizon discounted sum of payoffs, even if some finite horizon results are showed. The main difference with our work is that we focus on maximizing the largest reward minus the exploration costs, an objective function they cannot capture with their techniques, moreover their solution is defined with an implicit formula that becomes rapidly cumbersome as a function of the height of the tree. Interestingly enough they prove the optimality of a threshold strategy which is, in spirit, quite similar to ours.

[126], borrowing from the language of financial derivatives, introduce the *covered call value* of a box, which is the minimum of the reservation value, and its true (random) value. They show that the expected performance of any search strategy is at most the expected covered call value of the last kept box. This inequality is tight for any method which 'exercises in the money', i.e. immediately terminates search when the value of an opened box is greater than its reservation value. This novel point of view on the Pandora's Box Problem started a new interest in the problem. [127] investigate the existence of moving threshold strategies to address more general objective functions in the Pandora's setting, while [128] and [114] analyze a setting in which a box can be chosen without paying its cost while retaining its expected reward.

[115, 116] exploits the notion of *surrogate* box, an analog of the *covered call value*, to reduce the Pandora's Box Problem in the presence of downwards-closed constraints to adaptive maximization of nonnegative submodular functions, and bound the *adaptivity gap* of this problem, *i.e.* the ratio between the best adaptive solution and the best oblivious one as in [129] and later [130].

This *surrogate* box approach does not apply to our case because it requires the exploration to stop when a box contains a random reward greater than its own reservation price, *i.e.* the already mentioned 'exercises in the money' property of [126]. This cannot be true for exploration problem with order constraints, because the intrinsic value of a box in the exploration process is given not only by its own cost and reward, but also by the opportunities that its opening provides. Generally speaking the *surrogate* box techniques works well in "concave" situations unlike ours, when opening a box *decreases* the marginal expected value of further exploration, as for knapsack or matroid constraints.

Recently [117] and [118] studied a connection between the Pandora's Box Problem and another well known optimal stopping problem, termed the Prophet Inequalities. In the same line of research is included the study of the Pandora's Box Problem with Commitment [131, 132], where once a box is opened, the algorithm can either choose to keep the reward and terminate search, or discard the reward and continue, similarly to the Prophet Inequalities setting. Very recently [133] addressed the Pandora's Box Problem with correlation between the rewards of the different boxes, with a particular focus on strategies based on sampling, more than on the exact knowledge of the underlying distributions.

5.2 MODEL AND PRELIMINARIES.

In this section, we formally present our model, and give preliminaries. Recall, as the player, we adaptively open a constraint-satisfying set of boxes, paying for each one opened, while learning the actual value. The final payoff received is the largest value observed. This is formalized below.

The Pandora's Box Problem. We are given a set of boxes $B = \{b_1, \ldots, b_n\}$, where b_i costs c_i to open, and has random payoff X_i, whose distribution is known. The $\{X_i\}_{i=1}^n$ are independent and need not be identically distributed. We remark that the assumption of independence is very strong, but is indeed needed in order to obtain constant factor approximation to the optimal strategy, as recently shown by [133].

A strategy π is a rule which determines, at any integer time $t \geq 0$, whether to terminate the search and, if not, which box to open next. The strategy may depend on the time t, the values observed in the past, the structure of the problem and some extra randomness. We use equivalently the terms strategy, rule and policy.

Let $S_t(\pi)$ denotes the (random) set of boxes that have been opened at or before time t by strategy π, and let τ_π be the (random) stopping time given by the same strategy. We use the shortcut $S(\pi)$ to denote $S_{\tau_\pi}(\pi)$, the final set of opened boxes following strategy π. Given constraint-set $\mathcal{F} \subseteq 2^B$, π is said to be $\mathcal{F}$-feasible if $S_t(\pi) \in \mathcal{F}$ with probability 1, for all t.

Our goal is to choose a $\mathcal{F}$-feasible policy π^* which maximizes, in expectation, the following objective:

$$\mathbb{E}\left[\max_{i \in S(\pi^*)} X_i - \sum_{i \in S(\pi^*)} c_i\right]. \tag{5.4}$$

Such strategies are called optimal.

Order constraints. In this paper, we focus on *order constraints*, where some boxes are required to be opened after others. More formally, a (strict) partial ordering "$\prec$" on the boxes B is given, where $b_i \prec b_j$ means that the opening of box b_i "*unlocks*" box b_j, allowing it to be opened. Moreover we assume the existence of a unique *root* or *initial* box r such that $\nexists\, b \in B$ such that $b \prec r$, i.e. it is the least element with respect to $\prec$. Note that by definition of (strict) partial ordering on a finite set there exists at least one minimal element and its uniqueness can be assumed without loss of generality since it is always possible to add a dummy initial box pointing to all the minimal elements of the ordering.

For the ease of exposition we represent the underlying partial ordering on the boxes via its (rooted) directed acyclic graph G that we call the precedence graph. From now on we may refer equivalently to boxes or nodes, i.e. the nodes of the precedence graph where boxes lie. To be consistent with the notation above, given an ordering with initial element r and precedence graph G, the constraint-set is:

$$\mathcal{F}_G := \{S \subseteq B \,|\, \forall u \in S \setminus \{r\},\ \exists\, b \in S,\ \text{such that } b \prec u\} \tag{5.5}$$

Of special importance are partial orderings whose precedence graph is a tree. In this case we talk of Tree-Constrained Pandora's Box Problem. This particular type of constraints models situations in which there is only one way to go from the root to any node; hence the only condition for a box to be accessible is that its parent has already been explored.

Formally, given the precedence tree T with root r, the collection of feasible sets $\mathcal{F}_T$ is:

$$\mathcal{F}_T := \{S \subseteq B \,|\, \forall u \in S \setminus \{r\},\ \textsc{parent}(u) \in S\} \tag{5.6}$$

Order constraints may be seen as a special case of the more general *prefix-closed* constraints, which simply assert that for any legal sequence of moves, any truncation of this sequence is also legal. Formally, given a set of boxes B and a set $\mathcal{C}$ of possible orders of exploration, we say that $\mathcal{C}$ is prefix closed if for every $C \in \mathcal{C}$, every prefix of C is also in $\mathcal{C}$. Note that the order constraints defined above are a special case of this. Furthermore, intersecting any combination of order and downwards closed constraints results in some prefix closed family.

Threshold strategies and the Pandora's Rule. Of central importance for our paper is the concept of threshold strategy. A rule π is said to be a threshold strategy if it pre-computes a collection of *threshold values*, and greedily opens the boxes following these values, stopping when the amount earned is greater than the threshold of all remaining legal moves. Formally, the strategy is defined by a *threshold function* $z : B \to \mathbb{R}$ and works as in Algorithm 5.1.

Algorithm 5.1: Threshold strategy

```
Data: Distributions of the random rewards, box costs and a threshold function
        $z : B \to \mathbb{R}$
$S_0 \leftarrow \emptyset$, $y \leftarrow 0$, $t \leftarrow 0$
while $y < \max\{z(b) | b \in B \backslash S_t \textit{ and } \{b\} \cup S_t \in \mathcal{F}\}$ do
    Let $\hat{b} \in \arg\max\{z(b) | b \in B \backslash S_t \text{ and } \{b\} \cup S_t \in \mathcal{F}\}$, tie-breaking arbitrarily
    Open box $\hat{b}$, observe reward $\hat{X}$ and pay cost $\hat{c}$
    $S_{t+1} \leftarrow S_t \cup \{\hat{b}\},\ y \leftarrow \max\{y, \hat{X}\},\ t \leftarrow t + 1$
end
```

Observe that, given some consistent tie-breaking rule, the order of exploration is fixed, as the next box to consider only depends on the reservation values. We remark that any threshold function is defined *a priori*, *i.e.* it does not depend on the observed rewards, but only on the costs, the random distributions, and $\mathcal{F}$. This implies that the complexity of implementing a threshold strategy is strictly related to the complexity of computing the thresholds. These facts, which follow by definition, are formalized in the following.

Claim 5.5. Let π be a threshold strategy, where ties in the thresholds are solved arbitrarily but consistently. Then the following hold true:

1. **Fixed order:** Following strategy π, if $\Pr[b_i \text{ opened before } b_j] > 0$ then $\Pr[b_j \text{ opened before } b_i] = 0$, for all $i \neq j$.

2. **Efficiency:** If the threshold function z is efficiently computable, then so is π.

In addition to being natural, the importance of threshold strategies is showcased by the fact that the optimal strategy for the Pandora's Box Problem is indeed of this type. Formally, given a box b with cost $c > 0$ and nonnegative random reward X, we define the reservation value ζ of b as the smallest solution to

$$\mathbb{E}\left[(X - \zeta)_+\right] = c. \tag{5.7}$$

It can be shown that if X has finite mean, then the reservation value is well defined. The threshold strategy using the reservation values as thresholds is termed *Pandora's Rule* and is optimal for the Pandora's Box Problem.

The power of this strategy is that the reservation value depends only on the single box, allowing each box to be considered separately, leaving the problem dramatically more tractable. Later we show that threshold strategies are the key to solve optimally also the more complicated Tree-Constrained Pandora's Box Problem.

Markov Decision Process approach. Before proceeding with the details of the analysis, we would like to note that the Pandora's Box Problem with order constraints admits a naïve, albeit exponential-time solution: it suffices to solve a dynamic program whose states are all pairs (S, y) where $S \subseteq B$ is a set of boxes, and $y \in \mathbb{R}$ is the max value observed.

Nevertheless, the the literature on Markov Decision Processes (e.g. [134]) allows us to assess some properties of optimal strategies: there exists an optimal strategy π^* which is a Markovian policy mapping states to actions, *i.e.* the optimal next box to open is deterministic function of the state (S, y).

Distributional assumptions. As discussed in the beginning of Section 5.1, it is not restrictive to assume that the random variables $\{X_i\}_{i=1,\dots,n}$ are discrete, bounded above by M and are supported on $s = \text{poly}(n, M)$ values. When we say an algorithm runs in polynomial time, we mean polynomial in n and M.

Notation. In the following we use interchangeably $\max(a, b)$ or $a \vee b$ to denote the largest between two reals a and b. For the smallest we use $\min(a, b)$ or $a \wedge b$. Moreover, as already mentioned, $(a)_+ := \max\{a, 0\}$. We use the following simple equality repeatedly: $a \vee b - b = (a - b)_+$.

5.3 OPTIMAL SEARCH ON TREE CONSTRAINTS, NEW PROOF

The following section is a simplification of the proof of the same section in the original paper. This simpler proof has not been published, and first appears in this thesis. The original proof may be of interest since it gives a more detailed understanding of the correctness, whereas this sleeker, shorter proof may obfuscate some intuition.

The work of [126] presents an elegant proof for the optimality of the threshold-based search order given by Weitzman, in the un-constrained setting. We present it here first for completeness, and to introduce the concepts required to extend their proof to give the optimal algorithm in the tree-constrained setting, which we present next.

5.3.1 Proof of Optimality in the Unconstrained Setting

Recall, the reservation value ζ_i for box i is defined such that $\mathbb{E}[(X_i - \zeta_i)_+] = c_i$. The main idea in this simplified proof is to note that discounting the value X_i of the box i to be capped by ζ_i is equivalent in expectation to paying cost c_i. Formally, $\mathbb{E}[X_i - c_i] = \mathbb{E}[X_i \wedge \zeta_i]$. In context of an execution of Pandora's search, we can further use this idea to capture, in expectation, the marginal return from having opened a box. If the largest value seen in the past is denoted y, then the expected marginal gain from opening box i is given by $\mathbb{E}[(X_i - y)_+ - c_i]$; furthermore, so long as $y \leq \zeta_i$, this expected marginal return is equal to $\mathbb{E}[(X_i \wedge \zeta_i - y)_+]$. If $y > \zeta_i$, then by definition the expected marginal return of opening box i is negative, and it should not be opened.

Following this reasoning, we substitute each box in the analysis by a cost-less, ζ-capped, copy. In the following, let S be the (random) set of boxes that the algorithm has opened, let $\mathbb{A}_i$ denote the indicator that $i \in \arg\max_{k \in S} X_k$, breaking ties so that $\sum_i A_i = 1$, and let $\mathbb{I}_i$ denote the indicator that $i \in S$. We have then that the expected revenue to Pandora is

$$\begin{aligned}\mathbb{E}[\max_{i \in S} X_i - \textstyle\sum_{i \in S} c_i] &= \mathbb{E}[\textstyle\sum_{i=1}^n \mathbb{A}_i X_i - \mathbb{I}_i (X_i - \zeta_i)_+] \\ &\leq \mathbb{E}[\textstyle\sum_{i=1}^n \mathbb{A}_i (X_i - (X_i - \zeta_i)_+)] = \mathbb{E}[\textstyle\sum_{i=1}^n \mathbb{A}_i (X_i \wedge \zeta_i)] \qquad (5.8)\end{aligned}$$

Note that the inequality above is tight if and only if $\mathbb{I}_i(X_i - \zeta_i)_+$ is nonzero only when $\mathbb{A}_i = 1$. Thus, the above inequality is tight if we are playing a strategy which

(i) immediately terminates search if it opens a box i and observes $X_i \geq \zeta_i$, and

(ii) ensures that we are opening the boxes in decreasing ζ_i order, so as to not have the first condition occur for a box other than the $\arg\max_{k \in S} X_k$.

It remains to show that Weitzman's threshold-based strategy is optimal in expectation.

Claim 5.6 ([126])**.** It is optimal in expectation to follow Algorihm 5.1 with the ζ_i's as thresholds, *i.e.* to open the boxes in decreasing ζ_i order, until the largest value seen exceeds the next largest threshold.

Proof. We note that this strategy satisfies the above two conditions, and therefore the expected return will be equal to the expected, ζ-capped, maximal box value of the right-hand-side of (5.8). Furthermore, this strategy ensures that $\mathbb{A}_i$ is also the indicator for $\arg\max_{i=1}^{n}(X_i \wedge \zeta_i)$ with probability 1, *i.e.* it maximizes the argument of the expectation in the bound, and thus we conclude that it is optimal.

To see that the bound is optimized, Let S be the set of opened boxes, and T be the complement. By definition, we have $\zeta_i \geq \zeta_j$ for every $i \in S$, $j \in T$, and since we have stopped searching, we have that $\max_{i\in S} X_i \geq \max_{j\in T} \zeta_j$. Therefore, the maximizer for $(X_i \wedge \zeta_i)$ must be contained in S. Let $i^* = \arg\min_{i\in S} \zeta_i$ be the last box opened. We must have that $\max_{i\in S-i^*} X_i < \zeta_{i^*}$. If $X_{i^*} \geq \zeta_{i^*}$, then $(X_{i^*} \wedge \zeta_{i^*})$ maximizes the capped values, and $\mathbb{A}_{i^*} = 1$. If not, then none of the opened X_i's are capped by their thresholds, and the capped and un-capped maxima are the same. *Q.E.D.*

5.3.2 Proof of Optimality for Tree-Constraints

We generalize here the proof method for the tree-constrained setting. Let T be the rooted-forest graph, we wish to design thresholds for the nodes of T so that Algorithm 5.1 is optimal in expectation. Analogously to the un-constrained setting, the thresholds will represent a critical best-seen-value at which we are indifferent between opening or not. However, we must take descendants into account.

We define the thresholds recursively. Let $u \prec v$ denote the ancestry-order in T, and $C(u)$ denote the set of u's children. If u is a leaf in T, then the threshold $z(u)$ is the usual Weitzman threshold ζ_u from the previous section, and let $Y_u := X_u \wedge z(u)$ denote the capped box value. If instead u is an internal node, we define $z(u)$ and Y_u recursively. Let $z(u)$ be the solution to

$$\mathbb{E}\Big[\big((X_u \vee \max_{v\in C(u)} Y_v) - z(u)\big)_+\Big] = \mathbb{E}\Big[\big(X_u \vee \max_{v\prec u}(X_v \wedge \min_{v\preceq w\prec u} z(w)) - z(u)\big)_+\Big] = c_u \qquad (5.9)$$

and define $Y_u := z(u) \wedge (X_u \vee \max_{v\in C(u)} Y(u))$, the capped value of the subtree as a whole. Note that $z(u)$ is the value in the past for which we are indifferent between optimally navigating the subtree rooted at u or stopping search.

A key observation is to note that, if v is u's descendant, and $z(v) < z(u)$, then the threshold $z(u)$ does not "see" the sub-tree rooted at v, in the sense that this sub-tree is evaluated as zero in (5.9), since we will have $Y_v < z(u)$. To this end, we define *threholded subtrees* to capture the subset of the subtree rooted at u that is "seen" in the calculation of $z(u)$.

Definition 5.1. The *threholded subtree* rooted at r, denoted $T(r)$, is recursively defined as follows: $r \in T(r)$, and for any $u \in T(r)$, if v is a child of u, and $z(v) \geq z(r)$, then $v \in T(r)$. Equivalently, $u \in T(r)$ if for all v in the path between r and u along the tree, we have $z(v) \geq z(r)$.

Note that the thresholded subtrees form a laminar set family: if $u \in T(r)$, then $T(u) \subseteq T(r)$, since $z(u) \geq z(r)$, and the paths inside $T(u)$ along high-threshold nodes can be extended to r.

From the above discussion, the threshold $z(r)$ is exactly the threshold at which we are indifferent between entering $T(r)$, if it were the entire set of boxes. Furthermore, the return from exploring $T(r)$ is equivalent to Y_r in expectation. Note that we can re-define $z(r)$ as the solution to

$$\mathbb{E}\Big[\big(X_r \vee \max_{u \in T(r)} (X_u \wedge \min_{v \in T(r): u \in T_v} z(v)) - z(r)\big)_+\Big] = c_r \tag{5.10}$$

Theorem 5.7. Algorithm 5.1, following the $z(u)$'s computed above, is optimal in expectation, and the thresholds can be computed in polynomial time and space.

Proof. We begin by arguing that the above discussion gives a space-and-time efficient tree-recursion method for computing the threshold values, so long as expectations can be computed. Since we have argued that the distributions can be assumed to be finitely supported at the cost of small error, this is feasible.

It remains to show that the choice of thresholds leads to the optimality of the algorithm. Analogously to (5.8), let S be the set of opened boxes, $\mathbb{I}_u$ be the indicator that $u \in S$, $\mathbb{A}_u$ be the indicator that $u = \arg\max_{v \in S} X_v$, and $\mathbb{T}_u$ the indicator that $T(u)$ contains the arg-max. Thus, $\mathbb{I}_u \geq \mathbb{T}_u \geq \mathbb{A}_u$. We have

$$\begin{aligned}\mathbb{E}[\max_{u \in S} X_u - \textstyle\sum_{u \in S} c_u] &= \mathbb{E}\left[\textstyle\sum_u \mathbb{A}_u X_u - \mathbb{I}_u\big(X_u \vee \max\limits_{v \in T(u)} (X_v \wedge \min\limits_{w \in T(u): v \in T(w)} z(w)) - z(u)\big)_+\right] \\ &\leq \mathbb{E}\left[\textstyle\sum_u \mathbb{A}_u X_u - \mathbb{T}_u\big(X_u \vee \max\limits_{v \in T(u)} (X_v \wedge \min\limits_{w \in T(u): v \in T(w)} z(w)) - z(u)\big)_+\right]\end{aligned} \tag{5.11}$$

Now, let u^* be the vertex where $\mathbb{A}_u = 1$, *i.e.* the arg-max, and let ℓ be the depth of $T(u^*)$ in the laminar set-system of thresholded subtrees. Call $r_1, r_2, \ldots, r_\ell = u^*$ the roots of these thresholded subtrees, such that $T(r_1) \supset T(r_2) \supset \cdots \supset T(r_\ell)$. Note that the r_i's are exactly the nodes for which $\mathbb{T}_{r_i} = 1$. Thus,

$$\begin{aligned} &\mathbb{E}\left[\textstyle\sum_u \mathbb{A}_u X_u - \mathbb{T}_u\big(X_u \vee \max_{v\in T(u)} (X_v \wedge \min_{w\in T(u):v\in T(w)} z(w)) - z(u)\big)_+\right] && (5.12)\\ &= \mathbb{E}\left[X_{i^*} - \textstyle\sum_{i=1}^{\ell} \left(X_{i^*} \wedge z(r_{i+1}) - z(r_i)\right)_+\right] = \mathbb{E}[X_{i^*} \wedge z(r_1)] && (z(r_{\ell+1}) := \infty)\end{aligned}$$

Thus, the revenue is upper-bounded by the capped value of the arg-max box, capped by the outermost thresholded subtree that contains it. We first argue that the above inequality is tight if and only if our strategy meets two basic conditions, and then that the largest possible value can be attained exactly, similarly to the unconstrained case. We impose the following conditions:

(i) If we open box r, then we will open all of $T(r)$ before moving to another subtree.

(ii) If we open box $i \in T(r)$, and $X_i > z(r)$, then the only other boxes we open will lie in $T(r)$.

It suffices to argue that the multiplicand after $\mathbb{I}_r$ is zero if $r \in S$ but $\mathbb{T}_r = 0$. By (i), if we open r, and find no $X_u > z(r)$ in $T(r)$, then we will continue to another thresholded subtree, but we will also have that $X_u < z(r)$ for all $u \in T(r)$, and so the multiplicand evaluates to zero. If instead we find some $X_u > z(r)$, then by (ii), we will terminate search without leaving $T(r)$, and the arg-max will lie in the subtree, so $\mathbb{T}_r = 1$.

Note that the threshold strategy following the $z(u)$'s satisfies these two conditions. If we open box r, then $z(r)$ was the largest available threshold, but now every box in $T(r)$ has a threshold at least as large. If we find some $X_u > z(r)$, then we will not want to open any of the boxes available when r was opened. It remains to show that we achieve the optimal value for the bound. This is argued like in the unconstrained setting, by considering only the top-level thresholded subtrees (maximal with respect to inclusion in the laminar set system), and the largest X_i's contained within them: The largest opened capped-value must be greater than the largest un-opened threshold, and either all the opened values were small enough that none are capped, or only the last thresholded subtree opened attains the threshold. *Q.E.D.*

5.4 ADAPTIVITY GAPS AND APPROXIMATION BEYOND TREE CONSTRAINTS.

In the previous section, we sought to design exactly optimal policies. As we will see in Section 5.5, we can not hope to do so for more general constraints, as the problem becomes

NP-hard to approximate. For this reason, we seek instead to find approximately optimal solutions, as presented in Section 5.4.1. Then, in Section 5.4.2, following recent literature on stochastic probing, e.g. [115, 116, 129, 130], we go through an *adaptivity gap* route, arguing that for any adaptive strategy, there exists a non-adaptive strategy — *i.e.* pre-computing a fixed set and opening it obliviously — which approximates its performance. Therefore, the optimal non-adaptive strategy is a good approximation of the optimal adaptive strategy. However, as our setting is very broad, and captures many of the complexities of stochastic submodular optimization, it is not likely that an optimal non-adaptive set will be easy to find. Instead, following an approach similar to [135], in Section 5.4.3, we present a simple adaptive strategy which performs, in expectation, better than *any* fixed set, and yields therefore a good approximation for the optimal adaptive strategy for a large class of Pandora's Box Problem with prefix closed constraints.

5.4.1 The approximation benchmark.

As a starting point, notice that the standard notion of approximation — *i.e.* finding a solution whose performance is at least a multiplicative factor of the optimal solution in expectation — is not meaningful in this setting for two reasons. First, any hard example (*e.g.* the hardness proof in Theorem 5.13 and its Corollary 5.1) can be modified by adding a large-cost-no-payoff dummy box at the root of the tree. For the right cost, optimal strategies would have positive revenue, but approximately optimal strategies would have negative revenue. Second, the classical multiplicative approximation notion would imply an unbounded adaptivity gap even for very simple problems. For the sake of completeness and to give a deeper idea of why we need a different definition of approximation, we report in the following Claim a proof of this fact, loosely based on [116] and the extended version of [115].

Claim 5.8. There exists an instance of the Pandora's Box Problem for which the ratio between the expected rewards of the best adaptive policy and the best non-adaptive is unbounded.

Proof. Proof: Let $p > 0$, and consider n identical boxes with cost $c = 1 - p/2$, and reward $\frac{1}{p^2}$ with probability p^2, 0 otherwise. Since $c < 1$, the adaptive optimal strategy is to open boxes until you get the reward, which guarantees reward $\frac{1}{p^2}$ and costs $\frac{c}{p^2}$ in expectation, for a total expected revenue of $\frac{1}{p^2}(1 - c) = \frac{1}{2p}$

We now consider the non-adaptive strategy which opens k boxes. It earns $\frac{1}{p^2}$ with probability $1 - (1 - p^2)^k$ and 0 otherwise, and pays ck. Note that $(1 - p^2)^k$ is convex in k, and so

$\frac{1}{p^2}(1-(1-p^2)^k)$ is concave, and thus has non-increasing derivatives. At $k = \frac{1}{p}$, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}k}\left[-ck + \tfrac{1}{p^2}(1-(1-p^2)^k)\right] &= -c - (1-p^2)^k \cdot \frac{\ln(1-p^2)}{p^2} \\
&\le -c + \frac{-(-p^2)+p^4}{p^2} \cdot (1-p^2)^k && \text{if } p^2 \le \tfrac{1}{2} \qquad (5.13) \\
&\le -c + (1+p/8)(1-p^2)^k && \text{if } p \le \tfrac{1}{2} \\
&\le -c + (1+p/8)e^{-p} \\
&\le -(1-p/2) + (1+p/8)(1-0.632p) && \qquad\qquad (5.14) \\
&\le (\tfrac{1}{2} - 0.632 + 0.125)p = -0.007p < 0
\end{aligned}$$

Where (5.13) holds because $\ln(1+x) \ge x - x^2$ for $x \in [-\frac{1}{2}, 0]$: at $x = 0$, $\ln(1+x) = x - x^2 = 0$, and $\frac{\mathrm{d}}{\mathrm{d}x}\ln(1+x) = \frac{1}{1+x} \le \frac{\mathrm{d}}{\mathrm{d}x}x - x^2 = 1 - 2x$ over the domain, by convexity of the former. (5.14) holds since $e^{-p} \le 1 - (1-1/e)p$ for $p \in [0,1]$, again by convexity. Hence, the derivative is negative, and we conclude the optimum is attained on $1 \le k < 1/p$. However, $(1-p^2)^k \ge 1 - kp^2$, so

$$-ck + \tfrac{1}{p^2}(1-(1-p^2)^k) \le -ck + \tfrac{1}{p^2}kp^2 = (1-c)k \tag{5.15}$$

Since $1 - c = p/2$, and $k \le 1/p$, this upper-bounds the total revenue by $\frac{1}{2}$.

Recalling that the adaptive strategy earned on average $1/2p$, then the ratio of the two revenues is $\frac{1}{p}$. Taking $p \to 0$, this suggests that the adaptivity gap can be arbitrarily large.

We remark that this counterexample works also for the single line constrained case as the boxes are all equal and the order is irrelevant. *Q.E.D.*

With this in mind, we need to somehow decouple the two terms of the objective function: the reward and the sum-of-costs part. We define below a modified notion of approximately optimal solution, which exploits this idea. Note that similar metrics have been used before, in works relative to adaptive submodular maximization, such as [136, 137, 138]. This relation is indeed quite natural since the Pandora's Box Problem can be cast as an adaptive submodular optimization problem.

Definition 5.2 (Approximately Optimal Solutions)**.** In this paper we consider a notion of approximation which considers only a fraction of the reward term of the optimal solution, while paying similar costs. Formally, given a Pandora's Box Problem with prefix closed constraints, a (possibly adaptive) strategy $\hat{\pi}$ is a $C \in (0,1]$ approximation if, for any other rule π, it holds that

$$\mathbb{E}\left[\max_{i\in S(\hat{\pi})} X_i - \sum_{i\in S(\hat{\pi})} c_i\right] \geq C \cdot \mathbb{E}\left[\max_{i\in S(\pi)} X_i\right] - \mathbb{E}\left[\sum_{i\in S(\pi)} c_i\right]. \tag{5.16}$$

5.4.2 The adaptivity gap.

The previous definition translates into the notion of adaptivity gap we use in the rest of the paper. The main result is that, differently from Claim 5.8, this adaptivity gap can be bounded. The proof of this theorem closely follows [130], but it is short, and so we include it here for completeness.

Theorem 5.9. *Consider the Pandora's Box Problem with constraints modeled by some prefix closed family, then for every adaptive strategy π, there exists a non-adaptive strategy, i.e. a feasible set S, such that the following holds:*

$$\mathbb{E}\left[\max_{i\in S} X_i - \sum_{i\in S} c_i\right] \geq \tfrac{1}{2}\mathbb{E}\left[\max_{i\in S(\pi)} X_i\right] - \mathbb{E}\left[\sum_{i\in S(\pi)} c_i\right] \tag{5.17}$$

Proof. Proof: The proof is a relatively simple, but clever, idea introduced in the adaptivity gap upper-bound of [130]. The idea is to show that if we choose a set at random, according to the distribution induced by $S(\pi^*)$ from the randomness on the rewards, then this set will perform well in expectation over both the random set, and the random rewards. It follows that there must exist some set which performs at least as well as this in expectation.

Formally, we wish to show that if we randomly sample the value of each box twice, choose optimal boxes adaptively for one of the samples, but measure revenue using the other samples, we lose only a factor 2 in the expectation of the $\max_{i\in S} X_i$ term. As for the $\sum_{i\in S} c_i$ term, we are opening the same set, so they cost the same. Note that this considers only feasible sets S, by definition.

To this end, let $X_1, \ldots, X_n$ be the random payoff values of the boxes, and let $Z_1, \ldots, Z_n$ be respectively identically distributed copies of the X_i's, sampled independently. Fix an optimal adaptive strategy π, and let $\pi(S, y) \in [n]$ denote the choice of the next box to open after having opened S, and observing largest value y. Let $\mathbb{S}(\pi, X|S, y)$ be the (random) final set that π opens when it chooses to terminate, if it starts with set S and total y. We denote

$$\mu_Z(S, y, y') := \mathbb{E}\left[(-y' + \max\{Z_i : i \in \mathbb{S}(\pi, X|S, y),\, i \notin S\})_+\right] \tag{5.18}$$

the expected future gain when playing according to the X_i values starting in state (S, y), but measuring revenue with the Z_i's from state (S, y'). Note that $\mu_X(\emptyset, 0, 0)$ is the expected revenue of playing according to the adaptive strategy, and $\mu_Z(\emptyset, 0, 0)$ is the expected revenue of randomly picking a set according to the Z_i's.

We wish to show $\mu_X(S, y, y') \leq 2\mu_Z(S, y, y')$, by induction on the set S, as it ranges over all subsets, in decreasing order of cardinality. Note that if (S, y) is such that the policy π will choose to terminate, then both values are $y - y'$. Otherwise, fix S, y, and y', and let $p := \pi(S, y)$. We have

$$\begin{aligned}
\mu_X(S, y, y') &= \mathbb{E}\left[(X_p - y')_+ + \mu(S + p,\ y \vee X_p,\ y' \vee X_p)\right] \\
&\leq \mathbb{E}\left[((X_p \vee Z_p) - y')_+ + \mu(S + p,\ y \vee X_p,\ y' \vee (X_p \vee Z_p))\right] \\
&\leq \mathbb{E}\left[(X_p - y')_+ + (Z_p - y')_+ + \mu(S + p,\ y \vee X_p,\ y' \vee (X_p \vee Z_p))\right] \\
&= \mathbb{E}\left[2(Z_p - y')_+ + \mu(S + p,\ y \vee X_p,\ y' \vee (X_p \vee Z_p))\right] \qquad (5.19)
\end{aligned}$$

Where the first inequality asserts that earning more up front an only help, and the last equality holds by linearity of expectation and the identical distributions of X and Z. Furthermore,

$$\begin{aligned}
\mu_Z(S, y, y') &= \mathbb{E}\left[(Z_p - y')_+ + \mu_Z(S + p,\ y \vee X_p,\ y' \vee Z_p)\right] \\
&\geq \mathbb{E}\left[(Z_p - y')_+ + \mu_Z(S + p,\ y \vee X_p,\ y' \vee (X_p \vee Z_p))\right] \qquad (5.20)
\end{aligned}$$

Since $\mu(S, y, y')$ is non-increasing in y'. By linearity of expectation, and by induction on S, we get $\mu_Z(\emptyset, 0, 0) \geq \frac{1}{2}\mu_X(\emptyset, 0, 0)$, as desired. *Q.E.D.*

Remark 5.1. The previous result has a more general field of application. Consider the problem of maximizing $u(S) - \sum_{i \in S} c_i$, where u is not simply the max but a general (monotone) submodular function. As [130] gives the proof for that case, Theorem 5.9 extends nearly as it is to the submodular-minus-sum-of-costs scenario.

5.4.3 Beating non-adaptive.

With this result in hand, it remains to show that we can develop an *adaptive* strategy which performs at least as well as every non-adaptive strategy, in expectation. We will take advantage of the fact that we are working in a tree-constraint, and label the boxes with a pre-order of the nodes of the tree. We will denote the index of box b as i_b. Recall that, by the properties of a pre-order, we have that for all b, if b has k descendants in the tree, then the descendants of b are exactly those boxes indexed by i_b+1, i_b+2, ..., i_b+k. This allows us to keep track of which boxes can legally be opened if we choose to not open b, since we may simply jump ahead in the pre-order.

We wish to use this fact to design a simple dynamic program computing the best adaptive strategy among all which only consider boxes following the pre-order. The pre-order allows us to use an index in the order to store the tree-constraint information on S, but it remains to efficiently encode information regarding the matroid constraint. To this end, we define here a characterization of all constraints with "oblivious feasibility oracles":

Definition 5.3. A constraint on the feasible sets S of boxes to open is said to have an *oblivious feasibility oracle* if it is characterized by a set function $D(S)$ with the following properties:

1. The cardinality of $\{D(S) : S \subseteq [n]\}$ is polynomial in n,
2. For any S and $u \notin S$, $D(S+u)$ is efficiently computable knowing only $D(S)$ and u, and
3. For any S, it can be efficiently determined whether S is feasible knowing only $D(S)$.

To illustrate this notion, we take as an example a *generalized knapsack constraint*, where every box b is assigned a vector $\boldsymbol{w}_b \in \mathbb{Z}_+^d$, and we have a capacity vector $\boldsymbol{m} \in \mathbb{Z}_+^d$. Here d is a constant. A set S is feasible if $\sum_{b\in S} \boldsymbol{w}_b \preceq \boldsymbol{m}$, taken componentwise. The function $D(S)$ is simply $\sum_{b\in S} \boldsymbol{w}_b$, and we require that the entries of $\boldsymbol{m}$ be polynomial in n.

Note that this generalized knapsack constraint includes, as a special case, knapsack constraints, cardinality constraints, and even partition matroids with $O(1)$ partitions.

We will define the function $\Psi(i, y, D)$ recursively below, which denotes the expected revenue if we start at position i in the sequence, having already collected y, with feasibility oracle value D. As a base case, $\Psi = 0$ when $D(S) = D$ implies S is not feasible, and $\Psi = y$ when $i = n+1$. Otherwise, let $\textsc{next}(i)$ denote the first position after i in the pre-order on the tree such that $\textsc{next}(i)$ is not a descendent of i. Then

$$\Psi(i, y, D(S)) := \max \begin{cases} y \\ \Psi(\textsc{next}(i), y, D(S)) \\ -c_i + \mathbb{E}\left[(X_i - y)_+\right] + \mathbb{E}\left[\Psi(i+1,\ y \vee X_i,\ D(S+i))\right] \end{cases} \tag{5.21}$$

Since the X_i's take only polynomially many values, then this function can be computed in polynomial time, by definition of D. We can also simultaneously compute the associated adaptive policy π as in Algorithm 5.2 below.

Algorithm 5.2: Approximately Optimal Adaptive Strategy

```
Data: Pre-ordering $b_1, \dots, b_n$, Oblivious feasibility oracle $D$, box costs, and
      random payoffs.
for $i \leftarrow n+1$ to $1$ do
    for $y \in Y,\ D \in \mathcal{D}$ do
        if $D$ is infeasible then $\Psi(i, y, D) \leftarrow 0$;
        else if $i = n+1$ then $\theta(n, y) \leftarrow$ TERMINATE and $\Psi(n+1, y, D) \leftarrow y$;
        else
            OPEN $\leftarrow -c_i + \mathbb{E}\left[(X_i - y)_+\right] + \mathbb{E}\left[\Psi(i+1,\ y \vee X_i,\ D(S+i))\right]$
            SKIP $\leftarrow \Psi(\text{NEXT}(i), y, D)$
            $\Psi(i, y, D) \leftarrow \max\{y, \text{OPEN}, \text{SKIP}\}$
            if $\max\{y, \text{OPEN}, \text{SKIP}\} = y$ then $\theta(i-1, y) \leftarrow$ TERMINATE;
            if $\max\{y, \text{OPEN}, \text{SKIP}\} = \text{OPEN}$ then $\theta(i-1, y) \leftarrow i$;
            if $\max\{y, \text{OPEN}, \text{SKIP}\} = \text{SKIP}$ then $\theta(i-1, y) \leftarrow \theta(\text{NEXT}(i), y)$;
        end
    end
end
```

Let Y be the set of all possible values attained by all the X_i's, and $\mathcal{D}$ be all possible values attained by $D(S)$. The θ function returned by the algorithm determines the policy: if we are in state (S, y), and the max index of an element in S is i, then $\pi(S, y) := \theta(i, y)$.

Claim 5.10. The strategy returned by Algorithm 5.2 is at least as good as any non-adaptive strategy.

Proof. Proof: This can be seen by induction on the i variable of the dynamic program. Let $S_{-j} := S \cap \{n-j, \dots, n\}$. We wish to show that

$$\Psi(n-j, y, D(S \setminus S_{-j})) \geq \mathbb{E}\left[\max_{i \in S_{-j}} X_i - c(S_{-j})\right] \quad \text{for all } y\ , \tag{5.22}$$

by induction on j. Note that for $j = 0$, both values are equal to the revenue of set S. For $j > 0$, regardless of whether $n-j+1 \in S$, Ψ takes the max over including it and not including it, and by induction, the following Φ term performs better than $S_{-(j-1)}$ in expectation. *Q.E.D.*

Combining Claim 5.10 with Theorem 5.9 gives us the following result:

Theorem 5.11. *For the Tree-Constrained Pandora's Box Problem augmented with oblivious-feasibility-oracle matroid constraints, we can efficiently compute a policy $\hat{\pi}$ such that $\forall\,\pi$,*

$$\mathbb{E}\left[\max_{i\in S(\hat{\pi})} X_i - \sum_{i\in S(\hat{\pi})} c_i\right] \geq \tfrac{1}{2}\mathbb{E}\left[\max_{i\in S(\pi)} X_i\right] - \mathbb{E}\left[\sum_{i\in S(\pi)} c_i\right]. \tag{5.23}$$

5.5 IMPOSSIBILITY AND HARDNESS RESULTS.

In this Section, we show the impossibility results outlined in Section 5.2. In Section 5.5.1 we prove that, when the precedence graph is not a tree, there may not exist a threshold-like optimal strategy. Then, in Section 5.5.2 show the approximation hardness of solving the Pandora's Box Problem with both general order constraints and $\mathcal{F} = \mathcal{F}_T \cap \mathcal{I}_{\mathcal{M}}$ where T is a tree and $\mathcal{I}_{\mathcal{M}}$ is the collection of independence sets of a matroid $\mathcal{M}$.

5.5.1 Suboptimality of threshold strategies.

Surprisingly, the optimality of threshold strategies is a feature only of Tree-Constrained Pandora's Box Problem. As soon as we consider the simplest order constraint with a single root which is no more a tree, as in Figure 5.1(a), we loose the existence of an optimal strategy based on a fixed ordering of the boxes.

Theorem 5.12. *The Pandora's Box Problem with order constraints need not admit an optimal threshold strategy, when the constraint graph is not a tree. Moreover the same holds for constraints $\mathcal{F}_T \cap \mathcal{I}_{\mathcal{M}}$, where T is a tree and $\mathcal{M}$ a matroid.*

Proof. Proof: Consider graph (a) in Figure 5.1 with the following parameters:

$$\begin{aligned} X_A &= \begin{cases} 2.5 & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases}, & X_B &= 2, & X_C &= \begin{cases} 3 & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases}, & X_D &= \begin{cases} 6 & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases}, \\ c_A &= 0, & c_B &= 1, & c_C &= 1 - \tfrac{\varepsilon}{2}, & c_D &= 0. \end{aligned} \tag{5.24}$$

For $\varepsilon \in [\frac{5}{4}, 2]$ it can be shown that it is optimal to start the exploration of the graph from A, then, depending on the realization of X_A it is optimal to open B (and then D) or to open C (then D and then possibly B). If we now consider an instance of the Pandora's Box Problem with order constraints on $\mathcal{F}_T \cap \mathcal{I}_{\mathcal{M}}$ where $\mathcal{I}_{\mathcal{M}}$ is all subsets of cardinality 4, and T is given by (b) in Figure 5.1, with boxes A, B and C and two copies E and F of D, then we inherit the results from (a). *Q.E.D.*

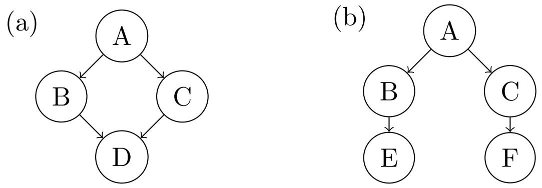

Figure 5.1: The order of optimal adaptive exploration is not fixed

5.5.2 Hardness of approximation.

As previously mentioned, we wish to show that it is NP-hard to approximate an optimal strategy for general order constraints and $\mathcal{F} = \mathcal{F}_T \cap \mathcal{I}_{\mathcal{M}}$ where T is a tree and $\mathcal{I}_{\mathcal{M}}$ are the independent sets of a matroid $\mathcal{M}$. Formally, we prove it is NP-hard to design a policy with approximately optimal rewards, for some constant. Approximation is taken in the sense of Definition 5.2. We remark that since it is a weaker notion than the standard multiplicative one, the results of this Section hold also in that case.

Theorem 5.13. *It is NP-hard to approximate within 0.9997 the optimal strategy to the Pandora's Box Problem with order constraints. It is sufficient for the DAG to have depth 2 and fan-in 3.*

Proof. Proof: It is known that it is NP-hard to approximate the minimum vertex cover of a cubic graph within a factor of $\approx 1.0012 =: 1 + \epsilon_0$, as shown in [139, 140]. Let $G = (V, E)$ be a hard-to-approximate instance, and let $n := |V|$, and $m := |E|$. Let α be such that the optimal vertex cover has size αm. Observe, since G is cubic, that $m := \frac{3}{2}n$, and $\alpha \geq \frac{1}{3}$. Furthermore, any greedy independent set must have at least $\frac{1}{3+1}n$ nodes, which implies that its complement is a vertex cover of size at most $\frac{3}{4}n = \frac{1}{2}m$. Thus, $\alpha \in [\frac{1}{3}, \frac{1}{2}]$.

We construct, now, the constraint graph D. The nodes of D will be labelled by $V \cup E$, where the V nodes will be the sources of the DAG, each having cost 1 and reward 0, and the E nodes will be the sinks of the DAG, each having cost 0 and reward βm with probability $\frac{c}{m}$ and 0 otherwise, for constants $\beta, c > 0$ which we will choose later. There is an edge connecting any vertex-box v to each edge-box e such that e in incident to v. Since G is cubic, this implies that D has depth 2 and fan-in 3, as required in the theorem statement.

Any optimal strategy must take the following form: (1) Fix an ordering on the boxes labelled by V, (2) Pay to open the next vertex-box in the order, then reveal the ≤ 3 unopened edge-boxes which it reveals. (3) Repeat until the reward has been collected. Suppose that the i-th vertex-box we pay for allows us to open $0 \leq n_i \leq 3$ new edge-boxes, and $N_i := \sum_{j=1}^{i-1} n_j$.

Then the expected max reward will be $\beta m \cdot (1 - (1 - \frac{c}{m})^m)$, and the expected cost will be

$$\mathbb{E}\left[\#\ V \text{ boxes opened}\right] \;=\; \sum_{i=1}^{n} \Pr \text{opening} \geq i \text{ boxes} \;=\; \sum_{i=1}^{n} (1 - \tfrac{c}{m})^{N_i} \tag{5.25}$$

Observe, without loss of generality, $n_{i-1} \geq n_i$ for all i, as swapping the $(i-1)$-st and i-th boxes will only increase N_i and leave N_{i+1} and onwards unchanged, reducing the expected cost. Thus, in any fixed order, after this swapping, there must exist numbers k_3, k_2, and k_1, such that

$$\begin{aligned} &n_1 = n_2 = \cdots = n_{k_3} = 3, \\ &n_{k_3+1} = n_{k_3+2} = \cdots = n_{k_3+k_2} = 2, \\ &n_{k_3+k_2+1} = \cdots = n_{k_3+k_2+k_1} = 1 \end{aligned} \tag{5.26}$$

Note that $3k_3 + 2k_2 + k_1 = m$, and that the vertex cover has size $k_3 + k_2 + k_1$. Setting $r = (1 - \frac{c}{m})$, we have that the expected cost becomes

$$\begin{aligned} &\sum_{i=1}^{k_3} r^{3i} + \sum_{i=1}^{k_2} r^{3k_3+2i} + \sum_{i=1}^{k_1} r^{3k_3+2k_2+i} \\ &\qquad = \tfrac{r^3}{1-r^3}\left(1 - r^{3k_3}\right) + \tfrac{r^2}{1-r^2}\left(r^{3k_3} - r^{3k_3+2k_2}\right) + \tfrac{r}{1-r}\left(r^{3k_3+2k_2} - r^m\right) \end{aligned} \tag{5.27}$$

In the remainder of the proof, we will bound the values of k_3, k_2, and k_1, for optimal and sub-optimal vertex covers, and show that the difference in expected cost is at least a constant factor of the expected reward. Since it is NP-hard to approximate the vertex cover, this will imply that is it NP-hard to approximate the optimal strategy for the Pandora's Box Problem on D.

Let S^* be an optimal vertex cover of size αm, and let S' be any vertex cover of size $\geq (1+\epsilon_0)\alpha m$. Let k_3^*, k_2^*, and k_1^*, be as above for the set S^*, and k_3', k_2', and k_1' be similarly for S'. We wish to lower-bound the cost of opening S^*, and upper-bound the cost of opening S', by bounding the possible values of the k^*'s and k''s obtained by sub- and super-optimal orderings, respectively. Note that we can trade off $k_3 + k_1$ for $2k_2$ to increase the expected cost, and vice versa. Since $\alpha \leq \frac{1}{2}$, then for S^*, it will suffice to assume $k_1^* = 0$, and increase k_3^* as α approaches $\frac{1}{3}$. For S', it will suffice to assume $k_2' = 0$. With the constraints on the

k^*'s and the k''s, this gives

$$\begin{cases} k_3^* = (1-2\alpha)m \\ k_2^* = (3\alpha - 1)m \end{cases} \qquad \begin{cases} k_3' = \frac{1}{2}(1-(1+\epsilon_0)\alpha)m \\ k_1' = \frac{1}{2}(3(1+\epsilon_0)\alpha - 1)m \end{cases} \tag{5.28}$$

Furthermore, in the expected cost expression above, we get

$$\begin{aligned} \mathbb{E}\left[\mathrm{cost}(S^*)\right] &\leq \tfrac{r^3}{1-r^3}\left(1 - r^{3k_3^*}\right) + \tfrac{r^2}{1-r^2}\left(r^{3k_3^*} - r^m\right) \\ &= \tfrac{r^3}{1-r^3} - \tfrac{r^2}{1-r^2}(r^m) + r^{3k_3^*}\left[\tfrac{r^2}{1-r^2} - \tfrac{r^3}{1-r^3}\right] \end{aligned} \tag{5.29}$$

$$\begin{aligned} \mathbb{E}\left[\mathrm{cost}(S')\right] &\geq \tfrac{r^3}{1-r^3}\left(1 - r^{3k_3'}\right) + \tfrac{r}{1-r}\left(r^{3k_3'} - r^m\right) \\ &= \tfrac{r^3}{1-r^3} - \tfrac{r}{1-r}(r^m) + r^{3k_3'}\left[\tfrac{r}{1-r} - \tfrac{r^3}{1-r^3}\right] \end{aligned} \tag{5.30}$$

Combining (5.29) and (5.30), the difference $\Delta := \mathbb{E}\left[\mathrm{cost}(S') - \mathrm{cost}(S^*)\right]$ is at least

$$\begin{aligned} \Delta &\geq -r^m\left[\tfrac{r}{1-r} - \tfrac{r^2}{1-r^2}\right] - \tfrac{r^3}{1-r^3}\left(r^{3k_3'} - r^{3k_3^*}\right) + \tfrac{r}{1-r}r^{3k_3'} - \tfrac{r^2}{1-r^2}r^{3k_3^*} \\ &= \left(r^{3k_3'} - r^{3k_3^*}\right)\left[\tfrac{r^2}{1-r^2} - \tfrac{r^3}{1-r^3}\right] + \left[\tfrac{r}{1-r^2}\right]\left(r^{3k_3'} - r^m\right) \end{aligned} \tag{5.31}$$

Recalling the values of k_3^* and k_3' above, expanding $r = 1 - \frac{c}{m}$, and first taking MacLaurin series around "$\frac{c}{m}$"$= 0$ for the terms in square brackets, then Taylor series for the terms in round brackets, we have

$$\begin{aligned} \Delta &\geq \left(\tfrac{m}{6c} + O(1)\right)\left((1-\tfrac{c}{m})^{3k_3'} - (1-\tfrac{c}{m})^{3k_3*}\right) + \left(\tfrac{m}{2c} + O(1)\right)\left((1-\tfrac{c}{m})^{3k_3'} - (1-\tfrac{c}{m})^m\right) \\ &= \tfrac{m}{6c}\left(e^{-3c(1-\alpha-\epsilon_0\alpha)/2} - e^{-3c(1-2\alpha)}\right) + \tfrac{m}{2c}\left(e^{-3c(1-\alpha-\epsilon_0\alpha)/2} - e^{-c}\right) + O(1) \\ &= \tfrac{m}{6c}\left(4e^{-3c(1-\alpha-\epsilon_0\alpha)/2} - 3e^{-c} - e^{-3c(1-2\alpha)}\right) + O(1) \end{aligned} \tag{5.32}$$

Setting $c = (2\epsilon_0)/(3\alpha)$, and for convenience, denoting $A := 1/\alpha$, we have

$$\Delta \;\geq\; \tfrac{m}{4A\epsilon_0}\left(4e^{-\epsilon_0(A-1-\epsilon_0)} - 3e^{-c} - e^{-3c(1-2\alpha)}\right) + O(1) \tag{5.33}$$

Recalling that $A \in [2,3]$. For $\epsilon_0 = 0.0012$ as in [139, 140], it can be shown that the function is non-increasing in A on its domain, and plugging $A = 3$, we numerically have $\Delta \geq 0.000399 \cdot m$. More generally, taking the second derivative in ϵ_0 suffices to show that the right hand side is strictly convex in ϵ_0, since $A \in [2,3]$, and its derivative is 0 when $\epsilon_0 = 0$. This ensures the constant is a positive function of ϵ_0.

It remains to determine the ratio of the difference in expected cost to the expected reward.

Recall that we have set the reward to be βm with probability $\frac{c}{m}$, and 0 otherwise. Since it costs 1 to open a box, and we wish to ensure that even when there is a single edge-box remaining, it is in our interest to open the box, we must set $\beta = \frac{1}{c}$. Recall, then, that the expected reward will be

$$\beta m \cdot (1 - (1 - \tfrac{c}{m})^m) = m \cdot \tfrac{1}{c}(1 - e^{-c} + O(\tfrac{1}{m})) = m + O(1) \tag{5.34}$$

Thus, an approximation for the Pandora's Box Problem which additively approximates the cost within a 1.00039 factor of the revenue implies an approximation algorithm for vertex cover on cubic graphs within a factor of < 1.0012, which is not possible unless $P = NP$. This concludes the proof with a multiplicative constant of $1 - 0.00039 < 0.9997$ in the sense of Definition 5.2. *Q.E.D.*

We argue here that, since the constraint graph G of the previous proof has depth 2 and fan-in 3, this also implies hardness for tree-and-matroid constraints, $\mathcal{F}_T \cap \mathcal{I}$. It suffices to replace each sink with 3 identical copies (including costs and rewards), assign one to each parent, and restrict that at most one copy of each is opened. This is exactly a partition matroid constraint, and the resulting graph is a forest of depth-2 trees, and is equivalent in terms of exploration costs and rewards to the constraint graph G. We have then proved the following Corollary.

Corollary 5.1. It is NP-hard to find the optimal strategy to the Pandora's Box Problem with order constraints with constraint $\mathcal{F} = \mathcal{F}_G \cap \mathcal{I}_\mathcal{M}$ where $\mathcal{M}$ can be any matroid on B even if G is restricted to be a tree. It suffices for $\mathcal{M}$ to be a partition matroid.

5.6 ROBUSTNESS TO APPROXIMATE DISTRIBUTIONAL INFORMATION, OR SAMPLE ACCESS

In this section we show how a polynomial number of samples from the random variables $\{X_i\}_{i=1}^n$ is enough to approximately solve the Pandora's Box Problem with any general prefix-closed constraint. Furthermore we prove that, for the Tree-Constrained Pandora's Box Problem, a linear number of samples is enough and that this is tight. These results are based on the techniques of [113].

We begin by remarking on the assumptions made in Section 5.1. First, we may safely assume that all variables have bounded support: if not, either there exists X_i with $\mathbb{E}[X_i] = \infty$, and the optimal solution is to go for that box no matter the cost. Conversely we may truncate the random variable without greatly affecting the problem. For any $\delta > 0$, there must exist a sufficiently large M_δ such that $\mathbb{E}\left[\max_i X_i | X_i \leq M_\delta \ \forall i\right] \geq (1-\delta) \cdot \mathbb{E}\left[\max_i X_i\right]$, and therefore $\Pr \max_i X_i > M_\delta \leq \delta$. Taking $\delta < n^{-1}$, this only affects the problem with vanishingly small probability, and we may choose δ arbitrarily smaller. Let M denote M_δ for our choice of $\delta > 0$. Thus, we may replace X_i with the random variable $\min\{X_i, M\}$. Now, up to scaling by this factor M — the largest reward achievable — we can assume the random variables are supported on $[0, 1]$.

At first glance, this M factor may be arbitrarily large. However, if it were the case that *e.g.* $M_{1/n} > n \cdot \mathbb{E}\left[\max_i X_i\right]$, meaning that a near-totality of the weight of the random variables occurred in this tail compared to the bounds expected from Markov's inequality, then it seems reasonable to treat this quantity as our scale.

5.6.1 The learning procedure.

Following [113], we use the *Product empirical revenue maximization* (PERM) paradigm: for any fixed $\epsilon > 0$, we take an $O(\epsilon)$-grid of the interval $[0, 1]$, and for sufficiently many samples, learn the empirical distribution on these grid points. We then compute the optimal search policy using this empirical distribution. Formally,

- For each box b, let X_b^ϵ be the random variable obtained by rounding down the reward X_b to the nearest multiple of ϵ;
- Given N *i.i.d.* samples of X_b^ϵ, let $\hat{X}_b$ be the random variable distributed according to the empirical distribution;
- Output the strategy $\hat{\pi}$ which is optimal with respect to the $\hat{X}_b$'s.

Note that here we do not consider the empirical joint distribution on the whole set of boxes, because it may not be independent, but we exploit our knowledge on the actual independence of the X_i and consider each box separately. We are now ready for the main result of the Section, which follows straightforward by Theorem 1 and 7 in [113].

That this is an ϵ-approximation is a straightforward application of standard techniques, and a proof is given in Appendix C2 of [113].

Theorem 5.14 ([113], Theorem 13). *For any $\epsilon > 0$ and $\delta > 0$ the PERM is a $1+\epsilon$ multiplicative approximation of the optimal strategy of a Pandora's Box Problem with prefix closed constraints with n boxes, as soon as the number of samples N satisfies*

$$N \geq C_1 \cdot \tfrac{n^3}{\epsilon^3} \log(\tfrac{n}{\epsilon\delta}) \tag{5.35}$$

for some universal constant $C_1 > 0$.

5.6.2 Tree constraints.

We prove here that linearly many samples are sufficient to learn the Tree-Constrained Pandora's Box Problem. The n^3 term in the previous bound comes from the fact that, when rewards and costs are bounded in $[0, 1]$, the total performance of a strategy may lie in the entire interval $[-n, 1]$, requiring the ϵ value to be normalized by $n + 1$. Following [113], we need two ingredient: a concentration result to show that with good probability an optimal strategy is never going to be "very negative", and a monotonicity property. We then conclude the following Theorem, using Lemma 25 from [113].

Theorem 5.15. *For the Tree-Constrained Pandora's Box Problem, it suffices to take*

$$N \geq C_2 \cdot \tfrac{n}{\epsilon^2} \log^2(\tfrac{1}{\epsilon}) \log(\tfrac{n}{\epsilon}) \log(\tfrac{n}{\epsilon\delta}) \tag{5.36}$$

for some universal constant C_2.

This latter result is tight up to $\operatorname{poly}\log(\frac{n}{\epsilon\delta})$ terms: [113] show that it takes at least $\Omega(\frac{n}{\epsilon^2})$ samples to get the desired degree of accuracy.

Let's then focus on the first ingredient: concentration. The goal is to show that, for an optimal algorithm, the performance over time forms a *submartingale.* Equivalently, one should only open a box if, in expectation, the revenue is increasing. This is not true at face value, as it is often necessary to open bad boxes to allow us to move onto better boxes. We use the notion of *thresholded subtrees* introduced in Section 5.3.

Claim 5.16. Let $S_i(\pi^*)$ denote the (random) set obtained by following the optimal policy π^* on the thresholded subtree $T(r)$, and define the random variable

$$M_i := \max_{\ell \in S_i(\pi^*)} X_\ell - \sum_{\ell \in S_i(\pi^*)} c_\ell. \tag{5.37}$$

Then the M_i's form a submartingale.

Proof. Proof: We must show that for all $i \geq 1$,

$$\mathbb{E}\left[M_{i+1} | M_i, \ldots, M_1\right] \geq M_i. \tag{5.38}$$

By the discussion in Section 5.3, we have that $z(u)$ remains unchanged for all $u \in T(r)$ if we remove nodes outside $T(r)$. But in this case, M_{i+1} is simply the performance of π^* on the whole set. By definition of reservation values, we then have that $\mathbb{E}\left[M_{i+1} | M_i\right] \geq M_i$. *Q.E.D.*

Turn now our attention to monotonicity, in particular to the following notion of *strong monotonicity* of the problem.

Definition 5.4 (First-Order Stochastic Dominance)**.** Random vector $\boldsymbol{X'}$ *stochastically dominates* $\boldsymbol{X}$ if, for every component i, and every $x \in \mathbb{R}$, we have $\Pr X_i' \geq x \geq \Pr X_i \geq x$.

Definition 5.5 (Strong Monotonicity)**.** A problem is *strong monotone* if for any random variable $\boldsymbol{X}$, and any random variable $\boldsymbol{X'}$ which dominates $\boldsymbol{X}$, Letting π^* be the optimal policy for the distribution on $\boldsymbol{X}$, we have that the performance of π^* over $\boldsymbol{X'}$ is at most the performance over $\boldsymbol{X}$.

The following is a direct corollary of Appendix C3 in [113], when viewed over the macro-boxes, as we have a fixed order of exploration, and reservation prices.

Claim 5.17. The Tree-Constrained Pandora's Box Problem is strongly monotone.

This concludes this Section. We have shown that for vanishingly small approximation guarantees, it suffices to assume the X_i's are supported on polynomial in n many values.

CHAPTER 6: BINARY PERFORMANCE METRIC ELICITATION FROM PAIRWISE CLASSIFIER COMPARISONS[1]

In this Chapter, we give efficient procedures for finding optimal binary classifiers, when the objective function is not known, but is accessible via a noisy comparison oracle. In this model, there is an underlying "population" random variable $X \in \Omega$, and a random *label* $Y \in \{0,1\}$ which is jointly distributed with X. A binary *classifier* h is a random variable which we construct, that is allowed to be jointly distributed with X as well, but must be independent of Y, when both are conditioned on X.

Formally, let $X \in \mathcal{X}$ and $Y \in \{0,1\}$ represent input and output random variables respectively. We denote a classifier by h, and let $\mathcal{H} = \{h : \mathcal{X} \to [0,1]\}$ be the set of all classifiers. We assume the existence of a data generating distribution $(X,Y) \overset{i.i.d.}{\sim} \mathcal{D}$. Let f_X be the marginal density on $\mathcal{X}$ given by f, and let $\eta(x) = \Pr[Y=1|X=x]$ and $\pi = \Pr[Y=1]$ represent the conditional and the unconditional probability of the positive class, respectively.

A confusion matrix for a classifier h is denoted by $C(h,\mathcal{D}) \in [0,1]^{2\times 2}$, comprising true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) and is given by:

$$\begin{aligned} C_{11} &= TP(h,\mathcal{D}) = \Pr[Y=1,h=1], \qquad & C_{01} &= FP(h,\mathcal{D}) = \Pr[Y=0,h=1], \\ C_{10} &= FN(h,\mathcal{D}) = \Pr[Y=1,h=0], \qquad & C_{00} &= TN(h,\mathcal{D}) = \Pr[Y=0,h=0]. \end{aligned} \tag{6.1}$$

Clearly, $\sum_{i,j} C_{ij} = 1$. We denote the set of all confusion matrices by $\mathcal{C} = \{C(h,\mathcal{D}) : h \in \mathcal{H}\}$. In fact, we can reduce dimension by noting that:

$$FN(h,\mathcal{D}) = \pi - TP(h,\mathcal{D}), \qquad FP(h,\mathcal{D}) = 1 - \pi - TN(h,\mathcal{D}). \tag{6.2}$$

Thus, the set of confusion matrices can be parametrized as $\mathcal{C} = \{(TP(h,\mathcal{D}), TN(h,\mathcal{D})) : h \in \mathcal{H}\}$. For clarity, we will suppress the dependence on $\mathcal{D}$ in our notation. In addition, we will subsume the notation h if it is implicit from the context and denote the confusion matrix by $C = (TP, TN)$. We represent the boundary of the set $\mathcal{C}$ by $\partial\mathcal{C}$.

Optimal Classifiers. The goal is to choose an h which is a "good" estimate for Y. The quality of a classifier is given as a function of its confusion matrix. A classifier h has score $\phi(C(h))$, where $\phi : [0,1]^{2\times 2} \to \mathbb{R}$ is a function on the confusion matrix, and does not otherwise depend on h. The binary classification problem requires us to find a ϕ-maximizing

[1] This chapter is based on collaboration with Gaurush Hiranandani, Ruta Mehta, and Sanmi Koyejo [98].

classifier h. For most "reasonable" objectives, it is not hard to show that the optimal classifier h must be a deterministic function of X, as this is often a Pareto improvement in C. Without loss of generality, we assume that ϕ is a utility, so that larger true positive and true negative rates values are better.

Problem Definition. In this model, ϕ is not given, but instead we have access to a *comparison oracle*: given two classifiers h, h', the oracle $\Gamma(h, h')$ returns whether or not $\phi(h) > \phi(h')$. However, we assume here that the oracle is *noisy*, *i.e.* when $|\phi(h) - \phi(h')| < \epsilon_\Gamma$, the oracle answers randomly. This models evaluating $C(h)$ from samples. Our goal is to, first, find a classifier h which approximately maximizes ϕ, and second, approximately learn ϕ.

As we can only access ϕ from pairwise comparisons, this goal is hopeless without further assumptions. We assume here that ϕ is a *linear fractional* function of $C(h)$, *i.e.* that it is the ratio of two affine functions of the entries of $C(h)$. This class is broad, and many performance metrics used in practice lie in this class.

Related Work. This problem can be seen to lie in the field of *derivative-free optimization*, see *e.g.* [141], though our work must develop a robust picture of the search space, due to our very limited information model. The work nearest to ours is that of [142], who pose a similar question and develop of theory for searching in their domain, though they do not give any theoretical performance guarantees. Furthermore, the problem of optimizing and estimating linear objectives from comparisons has been studied at length, including passive, or non-adaptive, settings, such as [68, 143, 144], or a more active-learning approach, or adaptive setting, such as [145, 146]. See [147] for a survey on active learning.

We seek to characterize the space of possible confusion matrices, and reason in confusion-matrix space, when we are querying classifiers. Note that optimizing allows us to learn the parameters of ϕ: Since the level-sets of a linear-fractional function consist of hyperplanes, all passing through some point outside of the domain, then finding the maximizer and the minimizer of ϕ gives you two points on the boundary of the region, and it is possible to determine their supporting hyperpelanes. Our procedure effectively "binary searches" along the boundary of the feasible region of confusion matrices — whose shape is determined by the randomness in $Y|X$. We show the following:

Theorem 6.1. For binary classification from noisy comparison queries to a linear-fractional metric, $O(\log(\frac{1}{\epsilon}))$ queries suffice to estimate ϕ with error at most $O(\epsilon + \sqrt{\epsilon_\Gamma})$. The hidden constants depend on the Lipschitz constants of ϕ, and on regularity assumptions on the distribution on (X, Y).

Performance Metrics in Practice. Selecting an appropriate performance metric is crucial to the real-world utility of predictive machine learning. This fundamental importance may explain the wide variety of performance metrics employed in practice, including accuracy and AUC [148] for classification, mean squared error for regression, and normalized discounted cumulative gain (NDCG) [149] for ranking, among several others. Interestingly, default metrics are the norm within the academic literature. For instance, applications with highly imbalanced binary classification problems, such as fraud detection [150], or applications where real-world costs are asymmetric with respect to predictions [151], often opt for the F_β-measure, the Jaccard Similarity Coefficient, and related default metrics which ideally reflect problem-specific tradeoffs [152].

Our result allows for *metric elicitation*, *i.e.* determining the performance metric from user feedback. This goal is motivated by the principle that the performance metric should reflect implicit user tradeoffs. This way, the resulting learning models best reflect the user preferences [151]. Our approach is inspired by a large literature in economics and psychology on *preference elicitation* [153, 154, 155, 156] which can be applied to learn user preferences in a marketplace. In these studies, the goal is to learn from buyers' purchases at posted prices – the prices may be posted by the mechanism or may come from data. Since there is no notion of prices or purchases in *metric elicitation* for machine learning, standard approaches from preference elicitation do not apply.

We additionally observe that large families of metrics are best characterized as functions of the confusion matrix [152, 157, 158]. This includes almost all modern metrics in common use. Particularly, since most of these common metrics are linear or ratio-of-linear functions of confusion matrices, this paper focuses on this important setting. As a result, pairwise classifier comparisons may be conceptually represented by their associated pairwise confusion matrix comparisons. Despite this apparent simplification, the problem becomes challenging because one can only query feasible confusion matrices, i.e, confusion matrices for which there exists a classifier. As we show, our characterization of feasible confusion matrices enables the design of a simple binary search procedure that identifies the underlying performance metric. While classifier comparisons may introduce additional noise, our procedure remains robust, both to noise from classifier estimation and to noise in the pairwise comparisons. Thus, our work directly results in a practical algorithm.

Comparison Oracles. On its face, metric elicitation simply requires querying a user to determine quality assigned to classifiers. Unfortunately, direct quality feedback may be ineffective, as humans are often inaccurate when asked to provide absolute preferences [159]. Instead, the user is asked to compare two classifiers and provide an indicator of relative preference. This is common practice and is applied by many web companies in the form of A/B testing [160], where the whole population of users acts as an oracle.[2] Moreover, comparisons of classifiers is becoming commonplace for a single expert. In fact, the ability to compare two classifiers is one of the primary contributions of the field of interpretable machine learning [161, 162].

6.1 PRELIMINARIES

6.1.1 Types of Performance Metrics

We consider two of the most common families of classification metrics, namely linear and linear-fractional functions of the confusion matrix (6.1).

Definition 6.1. Linear Performance Metric (LPM): We denote this family by φ_{LPM}. Given constants (representing costs or weights) $\{a_{11}, a_{01}, a_{10}, a_{00}\} \in \mathbb{R}^4$, we define the metric as:

$$\phi(C) = a_{11}TP + a_{01}FP + a_{10}FN + a_{00}TN = m_{11}TP + m_{00}TN + m_0, \tag{6.3}$$

where $m_{11} = (a_{11} - a_{10})$, $m_{00} = (a_{00} - a_{01})$, and $m_0 = a_{10}\pi + a_{01}(1 - \pi)$.

Example 6.2. Given a loss matrix $\mathbb{L} \in [0,1]^{2\times 2}$, such that L_{ij}, for $i, j \in \{0, 1\}$, denotes the loss incurred on predicting class j when the true class is i ($\mathbb{L}$ can be shifted and scaled to $[0,1]^{2\times 2}$ without changing the learning problem) [158], the performance metric is defined as:

$$\phi(C) = \sum_{i,j}(1 - L_{ij})C_{ij}. \tag{6.4}$$

For example, for the *0-1* loss given by $L_{ij} = \mathbb{I}(i \neq j)$, we have $\phi(C) = TP + TN$ (0-1 accuracy).

Definition 6.2. Linear-Fractional Performance Metric (LFPM): Given constants (representing costs or weights) $\{a_{11}, a_{01}, a_{10}, a_{00}, b_{11}, b_{01}, b_{10}, b_{00}\} \in \mathbb{R}^8$, we define the metric as:

[2] In A/B testing, sub-populations of users are shown classifier A vs. classifier B and their responses are used to determine the overall preference. Interestingly, while each person is shown a sample output from one of the two classifiers, the entire user population serves as the oracle for comparing classifiers.

$$\phi(C) = \frac{a_{11}TP + a_{01}FP + a_{10}FN + a_{00}TN}{b_{11}TP + b_{01}FP + b_{10}FN + b_{00}TN} = \frac{p_{11}TP + p_{00}TN + p_0}{q_{11}TP + q_{00}TN + q_0}, \tag{6.5}$$

where $p_{11} = (a_{11} - a_{10})$, $p_{00} = (a_{00} - a_{01})$, $q_{11} = (b_{11} - b_{10})$, $q_{00} = (b_{00} - b_{01})$, $p_0 = a_{10}\pi + a_{01}(1 - \pi)$, $q_0 = b_{10}\pi + b_{01}(1 - \pi)$. We denote this family by φ_{LFPM}.

Example 6.3. φ_{LFPM} includes the F_β measure [163] and the Jaccard similarity coefficient (JAC) [152]:

$$F_\beta = \frac{(1 + \beta^2)TP}{TP - TN + \beta^2\pi + 1 - \pi} \quad , \qquad JAC = \frac{TP}{1 - TN}. \tag{6.6}$$

6.1.2 Oracle Query

We query the oracle to determine relative preference between two classifiers. The space of classifiers $\mathcal{H}$ is infinite dimensional for Euclidean $\mathcal{X}$, because classifiers are functions, *i.e.* they assign a value for each $x \in \mathcal{X}$. However, the surjective mapping $\mathcal{H} \to \mathcal{C}$ results in a reduction from the infinite dimensional classifier space $\mathcal{H}$ to the finite dimensional confusion matrix space $\mathcal{C}$. It is clear that the oracle's preference may depend on factors such as interpretability and complexity; however, in this paper, we focus on the most common performance metrics which are functions of the confusion matrix as defined in Section 6.1.1. Therefore, a comparison query over classifiers boils down to a comparison query over confusion matrices which is formally defined below.

Definition 6.3. Given two classifiers h, h' (equivalent to confusion matrices C, C' respectively), a query to the Oracle (user) is represented by:

$$\Gamma(h, h') = \Omega(C, C') = \mathbb{I}[\phi(C) > \phi(C')] =: \mathbb{I}[C \succ C'], \tag{6.7}$$

where $\Gamma : \mathcal{H} \times \mathcal{H} \to \{0, 1\}$ and $\Omega : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$. This denotes whether h is preferred to h' (equivalent to C is preferred to C') as measured according to ϕ.

6.1.3 Bayes Optimal and Inverse Bayes Optimal Classifiers

Given a performance metric ϕ, the Bayes utility $\overline{\tau}$ is the optimal value of the performance metric over all classifiers, *i.e.* $\overline{\tau} = \sup_{h \in \mathcal{H}} \phi(C(h)) = \sup_{C \in \mathcal{C}} \phi(C)$. The Bayes classifier $\overline{h}$ is the classifier that optimizes the performance metric, so $\overline{h} = \arg\max_{h \in \mathcal{H}} \phi(C(h))$. Similarly, the Bayes confusion matrix is given by $\overline{C} = \arg\max_{C\in\mathcal{C}} \phi(C)$. We further define the inverse Bayes utility $\tilde{\tau} = \inf_{h\in\mathcal{H}} \phi(C(h)) = \inf_{C\in\mathcal{C}} \phi(C)$. The inverse Bayes classifier is given by $\underline{h} = \arg\min_{h\in\mathcal{H}} \phi(C(h))$. Similarly, the inverse Bayes confusion matrix is given by $\underline{C} = \arg\min_{C\in\mathcal{C}} \phi(C)$. Notice that for $\phi \in \varphi_{LPM}$ (6.3), the Bayes classifier predicts the label which maximizes the expected utility conditioned on the instance. We formalize this below.

Proposition 6.4. ([157]) Let $\phi \in \varphi_{LPM}$. Then, a classifier $\overline{h}$ of the form:

$$\overline{h}(x) = \left\{ \begin{array}{lr} \mathbb{I}[\eta(x) \geq \frac{m_{00}}{m_{11}+m_{00}}], & m_{11}+m_{00} \geq 0 \\ \mathbb{I}[\frac{m_{00}}{m_{11}+m_{00}} \geq \eta(x)], & o.w. \end{array} \right\} \tag{6.8}$$

is a Bayes optimal classifier *w.r.t.* ϕ. Further, the inverse Bayes classifier is given by $\underline{h} = 1 - \overline{h}$.

Proof. Note, we are maximizing a linear function on a convex set. Consider the following cases: (a) if $m_{11}, m_{00} \geq 0$, then the maximum is attained on $\partial\mathcal{C}_+$, and the proof below gives the desired result, (b) if $m_{11}, m_{00} < 0$, then the maximum is attained on $\partial\mathcal{C}_-$, and an argument identical to the proof below gives the desired result, and (c) otherwise their signs differ, and the maximum is attained either at $(0, 1-\pi)$ or $(\pi, 0)$, as per the discussion below. Which of the two it is depends on whether $|m_{11}| \geq |m_{00}|$, *i.e.* on the sign of $m_{11} + m_{00}$. It can be verified that in all 4 possible cases, the statement holds, noting that in all 4 cases, $m_{00}/(m_{11} + m_{00})$ is either ≤ 0 or ≥ 1.

We prove below, in lemma 6.1 and its following remark, that $\overline{h}$ must be of the form $\mathbb{I}[\eta(x) \geq t]$ for some t. It suffices to find t. Thus, we wish to maximize $m_{11}TP(h_t) + m_{00}TN(h_t)$. Now, let $Z := \eta(X)$ be the random variable obtained by evaluating η at random X. Note,

$$TP(h_t) \ = \int_{x:\eta(x)\geq t} \eta(x)\,\mathrm{d}f_X \ = \int_t^1 z\,\mathrm{d}f_Z \tag{6.9}$$

Similarly, $TN(h_t) = \int_0^t (1-z)\,\mathrm{d}f_Z$. Therefore,

$$\tfrac{\partial}{\partial t}\big(m_{11}TP(h_t) + m_{00}TN(h_t)\big) \ = -m_{11}tf_Z(t) + \cdot m_{00}(1-t)f_Z(t). \tag{6.10}$$

Thus, the critical point is attained at $t = m_{00}/(m_{11} + m_{00})$, as desired. A similar argument gives the converse result for $m_{11} + m_{00} < 0$. *Q.E.D.*

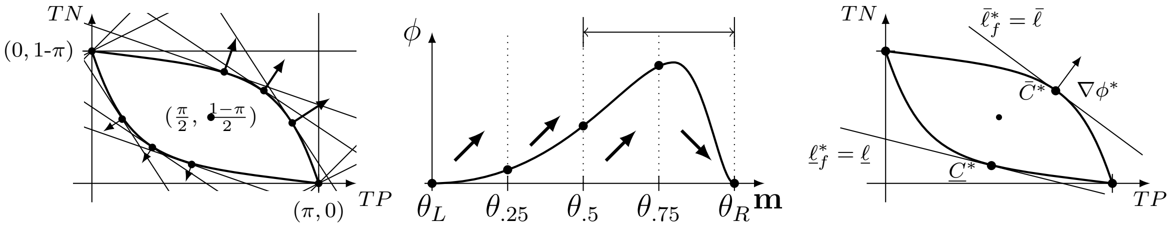

Figure 6.1: (a) Supporting hyperplanes and resulting geometry of $\mathcal{C}$; (b) Sketch of Algorithm 6.1; (c) Maximizer $\overline{C}^*$ and minimizer $\underline{C}^*$ along with the supporting hyperplanes for LFPMs.

6.2 CHARACTERIZING THE SPACE OF CONFUSION MATRICES

Our procedure for metric elicitation will require comparing feasible confusion matrices, *i.e.* confusion matrices which can be achieved by classifiers under the population law $\mathcal{D}$. Thus, it becomes necessary to properly characterize the set $\mathcal{C}$ in a way which is useful for the task of metric elicitation.

Assumption 6.5. We assume $g(t) = \Pr[\eta(X) \geq t]$ is continuous and strictly decreasing for $t \in [0, 1]$.

This is natural, and is equivalent to standard assumptions [157] that the event $\eta(X) = t$ has positive density but zero probability for all $t \in [0, 1]$. This requires X to have no point mass. The following are some basic properties of the space confusion matrices $\mathcal{C}$. See Figure 6.1(a) for intuition.

1. For all $(TP, TN) \in \mathcal{C}$, $0 \leq TP \leq \pi$, and $0 \leq TN \leq 1 - \pi$: No classifier can out-perform the best guess from the true distribution, *i.e.* $0 \leq \Pr[h = Y = 1] \leq \Pr[Y = 1] = \pi$. The rest is shown similarly.

2. $(\pi, 0) \in \mathcal{C}$ and $(0, 1 - \pi) \in \mathcal{C}$: these values are attained when h is identically 1 or 0, respectively.

3. For all (TP, TN) in $\mathcal{C}$, $(\pi - TP, 1 - \pi - TN) \in \mathcal{C}$: Let h be such that $TP(h) = TP$, $TN(h) = TN$. Then $TP(1 - h) = \Pr[(1 - h) = Y = 1] = \Pr[Y = 1] - \Pr[h = Y = 1] = \pi - TP(h)$, and a similar argument gives $TN(1 - h) = 1 - \pi - TN(h)$.

4. $\mathcal{C}$ is convex: Consider any two values (TP_1, TN_1), $(TP_2, TN_2) \in \mathcal{C}$, attained by h_1 and h_2, respectively. Let $0 \leq \lambda \leq 1$. Define h' as h_1 with probability λ, and h_2 otherwise.

Then,

$$\begin{aligned} TP(h') &= \Pr[h' = Y = 1] \\ &= \Pr[h_1 = Y = 1 | h = h_1] \Pr[h = h_1] + \Pr[h_2 = Y = 1 | h = h_2] \Pr[h = h_2] \\ &= \lambda TP(h_1) + (1 - \lambda) TP(h_2) \qquad (6.11) \end{aligned}$$

and a similar argument gives the convex combination for TN.

5. $\mathcal{C}$ has a supporting hyperplane associated to every normal vector. This follows from convexity and boundedness.

6. Any supporting hyperplane with positive slope is tangent to $\mathcal{C}$ at $(0, 1-\pi)$ or $(\pi, 0)$: This is true of the bounding region $[0, \pi] \times [0, 1-\pi]$ and $\mathcal{C}$ contains the points $(\pi, 0)$ and $(0, 1-\pi)$.

In the remainder of this section, we give geometric characterizations of the space of confusion matrices, which will be essential to the correctness of our algorithms.

Lemma 6.1. The boundary of $\mathcal{C}$ is exactly the confusion matrices of estimators of the form $\lambda\mathbb{I}[\eta(x) \geq t] + (1-\lambda)\mathbb{I}[\eta(x) > t]$ and $\lambda\mathbb{I}[\eta(x) < t] + (1-\lambda)\mathbb{I}[\eta(x) \leq t]$, where $\mathbb{I}$ is the indicator function.

Proof. To prove that the boundary is attained by estimators of these forms, consider solving the problem under the constraint $\Pr[h = 1] = c$. We have $\Pr[h = 1] = TP + FP$, and $\pi = \Pr[Y = 1] = TP + FN$, so we get

$$TP - TN \;=\; c + \pi - TP - TN - FP - FN \;=\; c + \pi - 1 \qquad (6.12)$$

a constant. This effectively partitions $\mathcal{C}$, since all confusion matrices are attained by varying c from 0 to 1, and no confusion matrix has two values of $TP - TN$. Furthermore, since restricting c restricts the search-space to an affine space A, then $\mathcal{C} \cap A$ has (by convexity and boundedness) exactly two endpoints, which are exactly those boundary points of $\mathcal{C}$ that are contained in A. (Unless $\mathcal{C} \cap A$ is a single point, in which case A is a hyperplane tangent to $\mathcal{C}$ at $(0, 1-\pi)$ or $(\pi, 0)$, from the above discussion.)

Since the affine space A has positive slope, we claim that the two endpoints are attained by maximizing or minimizing $TP(h)$ subject to $\Pr[h = 1] = c$. It remains to show that this happens for estimators of the form $h^{\lambda}_{t+} := \lambda\mathbb{I}[\eta(x) \geq t] + (1-\lambda)\mathbb{I}[\eta(x) > t]$ and $h^{\lambda}_{t-} := \lambda\mathbb{I}[\eta(x) < t] + (1-\lambda)\mathbb{I}[\eta(x) \leq t]$, respectively.

Let h be any estimator, and recall

$$TP(h) := \int_{\mathcal{X}} \eta(x) \Pr[h = 1 | X = x] \, \mathrm{d}f_X. \tag{6.13}$$

It should be clear that under a constraint $\Pr[h = 1] = c$, the optimal choice of h puts all the weight onto the larger values of η. Let t such that $\Pr[h^0_{t+} = 1] \leq t \leq \Pr[h^1_{t+} = 1]$, and let λ be chosen such that $\Pr[h^\lambda_{t+} = 1] = c$, then h^λ_{t+} must maximize $TP(h)$ subject to $\Pr[h = 1] = c$.

A similar argument show that all TP-minimizing boundary points are attained by the h_{t-}'s. *Q.E.D.*

Remark 6.1. Under Assumption 6.5, $\mathbb{I}[\eta(x) > t] = \mathbb{I}[\eta(x) \geq t]$ and $\mathbb{I}[\eta(x) < t] = \mathbb{I}[\eta(x) \leq t]$. Thus, the boundary of $\mathcal{C}$ is the confusion matrices of estimators of the form $\mathbb{I}[\eta(x) \geq t]$ and $\mathbb{I}[\eta(x) \leq t]$.

Proposition 6.6. (Properties of $\mathcal{C}$ — Figure 6.1(a).) The set of confusion matrices $\mathcal{C}$ is convex, closed, contained in the rectangle $[0, \pi] \times [0, 1 - \pi]$ (bounded), and 180° rotationally symmetric around the center-point $(\frac{\pi}{2}, \frac{1-\pi}{2})$. Furthermore, under Assumption 6.5, $(0, 1 - \pi)$ and $(\pi, 0)$ are the only vertices of $\mathcal{C}$, and $\mathcal{C}$ is strictly convex. Thus, any supporting hyperplane of $\mathcal{C}$ is tangent at only one point.

Proof. We argued above that $\mathcal{C}$ is convex and bounded. To see that $\mathcal{C}$ is closed, note that, from Lemma 6.1, every boundary point is attained. It remains then to prove that, under Assumption 6.5 — *i.e.* $g(t) = \Pr_X[\eta(X) \geq t]$ is continuous and continuously invertible — then $\mathcal{C}$ is strictly convex.

To see this, recall every boundary point of $\mathcal{C}$ can be attained by a thresholding estimator. By the discussion in Section 6.2, every boundary point is the optimal classifier for some linear performance metric, and the vector defining this linear metric is exactly the normal vector of the supporting hyperplane at the boundary point.

A vertex exists if (and only if) some point is supported by more than one tangent hyperplane. This means it is optimal for more than one linear metric. We know from Proposition 6.4 that optimal classifiers for linear metrics are threshold classifiers. Therefore there exist more than one threshold classifier with the same confusion matrix, and so there are multiple values of η which are never attained! This contradicts that g is continuously invertible.

A flat region exists if (and only if) some supporting hyperplane is tangent at multiple points. This means there exist two threshold classifiers with arbitrarily close threshold values, but confusion matrices that are well-separated. Therefore, there must exist some value of η which exists with non-zero probability, contradicting the continuity of g. *Q.E.D.*

For a linear performance metric ϕ (6.3), Proposition 6.6 guarantees the existence of a unique Bayes confusion matrix on the boundary $\partial\mathcal{C}$. This is because optimizing a linear function over a convex set results in the optimal being on the set boundary [164]. Note from Example 6.2 that any linear function with the same trade-offs for *TP* and *TN* is maximized at the same boundary point regardless of the bias term m_0. Thus, different LPMs can be generated by varying trade-offs $\boldsymbol{m} = (m_{11}, m_{00})$ such that $\|\boldsymbol{m}\| = 1$ and $m_0 = 0$. Further notice that $\|\boldsymbol{m}\| = 1$ does not affect the learning problem as discussed in Example 6.2. This allows us to represent the family of linear metrics φ_{LPM} by a single parameter $\theta \in [0, 2\pi]$:

$$\varphi_{LPM} = \{\boldsymbol{m} = (\cos\theta, \sin\theta) : \theta \in [0, 2\pi]\}. \tag{6.14}$$

Given $\boldsymbol{m}$ (equivalent to θ), it is straightforward to recover the Bayes classifier using Proposition 6.4, which further enables us to compute the Bayes confusion matrix $\overline{C}_\theta = \overline{C}_{\boldsymbol{m}} = (\overline{TP}_{\boldsymbol{m}}, \overline{TN}_{\boldsymbol{m}})$ using (6.1). Under Assumption 6.5, the Bayes confusion matrix $\overline{C}_{\boldsymbol{m}}$ is unique; therefore, we have that

$$\langle \boldsymbol{m}, C \rangle < \langle \boldsymbol{m}, \overline{C}_{\boldsymbol{m}} \rangle \qquad \forall\, C \in \mathcal{C}, C \neq \overline{C}_{\boldsymbol{m}}. \tag{6.15}$$

Notice the connection between the linear performance metrics and the supporting hyperplanes. Given $\boldsymbol{m}$, there exists a supporting hyperplane tangent to $\mathcal{C}$ at only $\overline{C}_{\boldsymbol{m}}$ defined as follows:

$$\bar{\ell}_{\boldsymbol{m}} = m_{11} \cdot tp + m_{00} \cdot tn - (m_{11}\overline{TP}_{\boldsymbol{m}} + m_{00}\overline{TN}_{\boldsymbol{m}}) = 0. \tag{6.16}$$

It is clear that if m_{11} and m_{00} are of opposite sign, then $\overline{h}_{\boldsymbol{m}}$ is the trivial classifier predicting either 1 or 0 everywhere. In other words, if the slope of the hyperplane is positive, then it touches the set $\mathcal{C}$ either at $(\pi, 0)$ or $(0, 1-\pi)$. When $m_{11}, m_{00} \neq 0$ and have the same sign, which means when linear ϕ is strictly monotonically increasing (or decreasing) in both *TP* and *TN*, then the Bayes confusion matrix is away from these two vertices. We can further break $\partial\mathcal{C}$ into two parts as follows:

Definition 6.4. The Bayes confusion matrices corresponding to $m_{11}, m_{00} \geq 0$ form the upper boundary, denoted by $\partial\mathcal{C}_+$. Similarly, the Bayes confusion matrices corresponding to $m_{11}, m_{00} < 0$ form the lower boundary, denoted by $\partial\mathcal{C}_-$. It follows that $\partial\mathcal{C}_+$ and $\partial\mathcal{C}_-$ correspond to the classifiers of the form $\mathbb{I}[\eta(x) \geq \delta]$ and $\mathbb{I}[\delta \geq \eta(x)]$, respectively, for some $\delta \in [0, 1]$ (see Propostion 6.4).

6.3 ALGORITHMS

In real-world scenarios, it is reasonable to assume that the metrics are monotonically increasing in both *TP* and *TN*, as we typically prefer metrics which reward correct decisions. While we first discuss this case, our algorithms also apply to the monotonically decreasing case as well. It suffices to query two classifiers whose relative performance is trivial to determine whether the objective is increasing or decreasing, and then the methods presented in this chapter can be easily modified to search on $\partial\mathcal{C}_-$ rather than $\partial\mathcal{C}_+$.

In addition, we assume an accuracy threshold $\epsilon_\Omega > 0$, such that the oracle answers correctly as long as $|\phi(C) - \phi(C')| > \epsilon_\Omega$. Otherwise, it may provide wrong answers. This means if the confusion matrices are close as measured by ϕ, then the responses to the comparison queries (6.7) can be wrong. We denote as ϵ_Ω the oracle's feedback noise. Later, we will show that our algorithms are robust enough to work well even with this noisy feedback, but first, we discuss the behavior of quasiconcave (quasiconvex) performance metrics, which are a much broader class of metrics than linear and linear-fractional, on the upper (lower) boundary of the set $\mathcal{C}$. A function is said to be quasiconcave (quasiconvex) if its super-level sets (sub-level sets) are convex. Formally, f is quasiconcave (quasiconvex) if $\{x : f(x) \geq t\}$ ($\{x : f(x) \leq t\}$) is convex for all t. These are monotone re-scalings of concave (convex) functions.

Lemma 6.2. Let $\rho^+ : [0,1] \rightarrow \partial\mathcal{C}_+$ and $\rho^- : [0,1] \rightarrow \partial\mathcal{C}_-$ be continuous, bijective, parametrizations of the upper and lower boundary, respectively. Let $\phi : \mathcal{C} \rightarrow \mathbb{R}$ be a quasiconcave function, and $\psi : \mathcal{C} \rightarrow \mathbb{R}$ be a quasiconvex function, which are monotone increasing in both TP and TN. Then the composition $\phi \circ \rho^+ : [0,1] \rightarrow \mathbb{R}$ is quasiconcave (and therefore unimodal) on the interval $[0,1]$, and $\psi \circ \rho^- : [0,1] \rightarrow \mathbb{R}$ is quasiconvex (and therefore unimodal) on the interval $[0,1]$.

Proof. We will prove the result for $\phi \circ \rho^+$ on $\partial\mathcal{C}^+$, and the argument for $\psi \circ \rho^-$ on $\partial\mathcal{C}^+$ is essentially the same. For simplicity, we drop the + symbols in the notation. A function is quasiconcave if and only if its superlevel sets are convex. Let S be some superlevel set of ϕ: since ϕ is monotone increasing, then $x \in S \implies y \in S$ for all $y \geq x$ componentwise. We want to show that for any $r < s < t$, if $\rho(r) \in S$ and $\rho(t) \in S$, then $\rho(s) \in S$. Since ρ is a continuous bijection, we must have — without loss of generality — $TP(\rho(r)) < TP(\rho(s)) < TP(\rho(t))$, and $TN(\rho(r)) > TN(\rho(s)) > TN(\rho(t))$. (otherwise swap r and t). Since the set is strictly convex and the image of ρ is $\partial\mathcal{C}$, then $\rho(s)$ must dominate some convex combination of $\rho(r)$ and $\rho(t)$, so by the convexity of S, $\rho(s) \in S$.

This implies that $\rho^{-1}(\partial\mathcal{C} \cap S)$ is an interval, and is therefore convex. Thus, the superlevel sets of $\phi \circ \rho$ are convex, so it is quasiconcave, as desired. (To see that this implies unimodaltiy,

Algorithm 6.1: Quasiconcave Maximization

```
Data: $\epsilon > 0$ and oracle $\Omega$.
Initialize $\theta_L \leftarrow 0$,  $\theta_R \leftarrow \frac{\pi}{2}$;
while $|\theta_R - \theta_L| > \epsilon$ do
    Set $\theta_{.25} \leftarrow \frac{3\theta_L+\theta_R}{4}$, $\theta_{.5} \leftarrow \frac{\theta_L+\theta_R}{2}$, and $\theta_{.75} \leftarrow \frac{\theta_L+3\theta_R}{4}$, the quarter interval boundaries;
    Set corresponding slopes ($\boldsymbol{m}$'s) using (6.14).;
    Obtain $\overline{h}_{\theta_L}$,$\overline{h}_{\theta_{.25}}$,$\overline{h}_{\theta_{.5}}$, $\overline{h}_{\theta_{.75}}$, $\overline{h}_{\theta_R}$ using Proposition 6.4;
    Compute $\overline{C}_{\theta_L}$,$\overline{C}_{\theta_{.25}}$,$\overline{C}_{\theta_{.5}}$,$\overline{C}_{\theta_{.75}}$, $\overline{C}_{\theta_R}$ using (6.1);
    Query $\Omega(\overline{C}_{\theta_{.25}}, \overline{C}_{\theta_L})$, $\Omega(\overline{C}_{\theta_{.5}}, \overline{C}_{\theta_{.25}})$, $\Omega(\overline{C}_{\theta_{.75}}, \overline{C}_{\theta_{.5}})$, and $\Omega(\overline{C}_{\theta_R}, \overline{C}_{\theta_{.75}})$;
    // Handle oracle error:
    if $\overline{C}_{\theta} \succ \overline{C}_{\theta'} \prec \overline{C}_{\theta''}$ for consecutive $\theta < \theta' < \theta''$ then replace with $\overline{C}_{\theta} \prec \overline{C}_{\theta'} \prec \overline{C}_{\theta''}$;
    // Recurse:
    if $\overline{C}_{\theta_L} \succ \overline{C}_{\theta_{.25}}$ then $\theta_R \leftarrow \theta_{.5}$;
    if $\overline{C}_{\theta_L} \prec \overline{C}_{\theta_{.25}} \succ \overline{C}_{\theta_{.5}}$ then $\theta_R \leftarrow \theta_{.5}$;
    if $\overline{C}_{\theta_{.25}} \prec \overline{C}_{\theta_{.5}} \succ \overline{C}_{\theta_{.75}}$ then $\theta_L \leftarrow \theta_{.25}$ and $\theta_R \leftarrow \theta_{.75}$;
    if $\overline{C}_{\theta_{.5}} \prec \overline{C}_{\theta_{.75}} \succ \overline{C}_{\theta_R}$ then $\theta_L = \theta_{.5}$;
    if $\overline{C}_{\theta_R} \succ \overline{C}_{\theta_{.75}}$ then $\theta_L = \theta_{.5}$;
end
return slope $\boldsymbol{m}_{.5}$ at $\theta_{.5}$ as $\overline{\boldsymbol{m}}$;  $\overline{C}_{\theta_{.5}}$ as $\overline{C}$;  and line $\langle \overline{\boldsymbol{m}}, (\mathrm{TP}, \mathrm{TN}) \rangle = \langle \overline{\boldsymbol{m}}, \overline{C} \rangle$ as $\bar{\ell}$
```

a function over the real line which has more than one local maximum can not be quasiconcave: consider the super-level set for some value slightly less than the lowest of the two peaks.) *Q.E.D.*

The unimodality of quasiconcave (quasiconvex) metrics on the upper (lower) boundary of the set $\mathcal{C}$ along with the one-dimensional parametrization of $\boldsymbol{m}$ using $\theta \in [0, 2\pi]$ (Section 6.2) allows us to devise binary-search-type methods to find the maximizer $\overline{C}$, the minimizer $\underline{C}$, and the first order approximation of ϕ at these points, *i.e.* the supporting hyperplanes at $\overline{C}$ and $\underline{C}$.

Maximization on the Upper Boundary. Algorithm 6.1 maximizes quasiconcave metrics and finds supporting hyperplanes at the optimum. Since ϕ is monotonically increasing in both *TP* and *TN*, and $\mathcal{C}$ is convex, the maximizer must be on the upper boundary. Hence, we start with the interval $[\theta_L = 0, \theta_R = \frac{\pi}{2}]$. We divide it into four equal parts and set slopes using (6.14) in lines 1 and 2 (See Figure 6.1(b) for visual intuition). Then, we compute the Bayes classifiers using Proposition 6.4 and the associated Bayes confusion matrices in lines 3 and 4. We pose four pairwise queries to the oracle in line 5. To handle oracle and finite sample noise, we re-order the queries if needed, in line 6. In the remaining, we shrink the search interval by half based on the responses from the oracle. We stop when the search interval becomes smaller than a given $\epsilon > 0$ (tolerance). Lastly, we output the slope $\overline{\boldsymbol{m}}$, Bayes confusion matrix $\overline{C}$, and the supporting hyperplane $\bar{\ell}$ at that point.

Optimization on the Lower Boundary. We can modify Algorithm 6.1 to minimize quasiconvex metrics and find supporting hyperplane at the minimum: we modify it by beginning in the range $\theta \in [\pi, \frac{3}{2}\pi]$. Furthermore, we check for $C \prec C'$ whenever Algorithm 6.1 checks for $C \succ C'$, and vice versa. These changes are sufficient to minimize a quasiconvex metrics. Here, we output the counterparts, *i.e.* slope $\tilde{\mathbf{m}}$, inverse Bayes Confusion matrix $\underline{C}$, and the supporting hyperplane $\tilde{\ell}$ at that point.

We prove below performance guarantees for Algorithm 6.1. We must first make an assumption on the distribution, and prove a lemma.

Assumption 6.7. For quasiconcave ϕ recall that the Bayes classifier is of the form $h = \mathbb{I}[\eta(x) \geq \delta]$. Let δ^* be the threshold that maximizes ϕ. We assume there exists $k_1 > 0$ such that $\Pr[|\eta(X) - \delta^*| \leq \epsilon] \leq k_1\epsilon$ for any $\epsilon > 0$. Further, we assume there exists $k_0 > 0$ such that $\Pr[(\delta^* - \eta(X)) \in [0, \epsilon]]$ and $\Pr[(\eta(X) - \delta^*) \in [0, \epsilon]] \geq k_0\epsilon$ for any $0 < \epsilon < \frac{2}{k_0}\sqrt{k_1\epsilon_\Omega}$.

This assumption ensures that near the optimal threshold, the values of η have bounded density, *i.e.* they are well spread. As an example, this holds for η logistic, so long as ϕ has positive weight on *TP* and *TN*, and X has no point-mass.

Theorem 6.8. Given $\epsilon, \epsilon_\Omega \geq 0$ and a metric ϕ that is monotonically increasing in TP and TN, if it is quasiconcave then Algorithm 6.1 finds an approximate maximizer $\overline{C}$. Whether we use it as-is, or use the above modifications for minimization, the following hold: (i) the algorithm also returns the supporting hyperplane at that point, (ii) under Assumption 6.7 from the next section, the value of ϕ at that point is within $\epsilon_\Omega + k_1\epsilon$ of the optimal value, and (iii) the number of queries is $O(\log \frac{1}{\epsilon})$.

Proof. (i) is a direct consequence of our representation of the points on the boundary via their supporting hyperplane. For (ii), by the nature of binary search, we are effectively narrowing our search interval around some target angle θ_0. Furthermore, since the oracle queries are correct unless the ϕ values are within ϵ_Ω, we must have $|\phi(\bar{\theta}) - \phi(\theta_0)| < \epsilon_\Omega$, and we output θ' such that $|\theta_0 - \theta'| < \epsilon$.

Recall, the Bayes classifier due to θ will report positive if $\cos\theta \cdot \eta(x) \geq \sin\theta \cdot (1 - \eta(x))$, and negative otherwise. Now, by Assumption 6.7, a small change in θ, (*i.e.* a small change in the threshold value, since cos and sin are Lipschitz) will only affect a small measure of the space. In other words

$$|TP(\theta_0) - TP(\theta')| = \left| \int_{x:\theta_0 < \eta(x) < \theta'} \eta(x)\, \mathrm{d}f_X \right| < k_1|\theta_0 - \theta'| < k_1\epsilon \ , \tag{6.17}$$

since $\eta(x) \leq 1$. A similar result applies to the true negative rate. This allows us to bound $\|C - C'\|_\infty$, since we are bounding each entry.

Since $\phi = \langle \boldsymbol{m}, C \rangle$ is linear with $\|\boldsymbol{m}\| = 1$, we have $|\phi(C) - \phi(C')| \leq 1 \cdot \|C - C'\|$, but

$$\|C(\theta_0) - C(\theta')\|_\infty \leq k_1|\theta_0 - \theta'| < k_1\epsilon \ . \tag{6.18}$$

Therefore, $|\phi(\bar{\theta}) - \phi(\theta')| < \epsilon_\Omega + k_1\epsilon$, as desired.

We needed only, for part (ii), that the interval of possible values be at most ϵ. This is obtained by making at least $\log_2(1/\epsilon)$ rounds of the algorithm, each of which is a constant number of pairwise queries. From this, we conclude (iii). *Q.E.D.*

Lemma 6.3. Under our model, no algorithm can find the maximizer in fewer than $O(\log \frac{1}{\epsilon})$ queries.

Proof. For any fixed ϵ, divide the search space θ into bins of length ϵ, resulting in $\lceil \frac{1}{\epsilon} \rceil$ classifiers. When the only operation allowed is pairwise comparison, the optimal worst case complexity for finding the maximum is $O(\log \frac{1}{\epsilon})$ [165], which is achieved by binary search. *Q.E.D.*

Since binary search always tends towards the optimal whenever responses are correct, we necessarily stop within a confidence interval of the true value. Thus, we can take ϵ sufficiently small so that the only error arises when we are in fact querying confusion matrices near the true optimal confusion matrix. Such details are discussed formally in the next section. Furthermore, the optimality of binary search in one-dimension given pairwise queries results in achieving the lower bound of Lemma 6.3.

6.3.1 Estimating Confusion Matrices — Sources of Noise

In all the algorithms above, we assumed knowledge of the confusion matrices, and made implicit assumptions regarding the noisiness of the samples. We seek to quantify these ideas. We begin by recalling that, as a standard consequence of Chernoff-type bounds [166], sample estimates of true-positive and true-negative are consistent estimators. Therefore, with high probability, we can estimate the confusion matrix within any desired sup-norm tolerance, provided we have sufficient samples. Recall that the oracle $\Omega(C, C')$ is only accurate so long as $|\phi(C) - \phi(C')| > \epsilon_\Omega$. Thus, when comparing two confusion matrices C and C', so long as ϕ is scaled to be 1-Lipschitz (true for linear and linear-fractional metrics), then if $|\phi(C) - \phi(C')| > \epsilon_\Omega + \epsilon$, we can compute sample estimates $\hat{C}$ and $\hat{C}'$ within $\epsilon/2$, and the triangle inequality gives $|\phi(\hat{C}) - \phi(\hat{C}')| > (\epsilon_\Omega + \epsilon) - 2 \cdot \epsilon/2 = \epsilon_\Omega$. Thus, with high probability, the oracle will accurately report the better of the two. We need the following assumption:

Assumption 6.9. Let $\hat{\eta}_n$ be a sequence of estimates of η. We assume that $\|\eta - \hat{\eta}_n\|_\infty \xrightarrow{P} 0$.

This assumption is arguably natural, as most estimation is parametric regression, where the function classes are sufficiently "well behaved". With this assumption and Assumption 6.7, we can control the error in optimal classifiers from using $\hat{\eta}$ rather than η. This allows us to prove the following lemma which is a key for showing the correctness of Algorithm 6.1.

Lemma 6.4. Let $\hat{h}_\theta$ be defined similarly as h_θ, but *w.r.t.* the estimated $\hat{\eta}$ rather than the true η. Then for any θ, there exists a θ' such that $TP(\hat{h}_\theta) \leq TP(h_{\theta'})$, $TN(\hat{h}_\theta) \leq TN(h_{\theta'})$, and $\|C(\hat{h}_\theta) - C(h_{\theta'})\|_\infty \leq \|\eta - \hat{\eta}_n\|_1$. Furthermore, let $\bar{\theta}$ be the parameter θ such that $h_{\bar{\theta}} = \arg\max_\theta \phi(h_\theta)$. Then $\|C(\hat{h}_{\bar{\theta}}) - C(h_{\bar{\theta}})\|_\infty = O(\|\hat{\eta}_n - \eta\|_\infty)$, where the constants come from Assumption 6.7.

Proof. Recall the proof of Lemma 6.1: let $\Pr[h_\theta = 1] = c$, then there must exist some θ' such that $\Pr[\hat{h}_{\theta'} = 1] = c$. For simplicity of notation, let $h_\theta = \mathbb{I}[\eta \geq t]$ and $h_{\theta'} = \mathbb{I}[\hat{\eta} \geq t']$. Then we have,

$$\begin{aligned}
&TP(h_\theta) - TP(h_{\theta'}) \\
&= \int_{\eta \geq t} \eta \, \mathrm{d}f - \int_{\hat{\eta} \geq t'} \eta \, \mathrm{d}f_X \qquad (6.19) \\
&= \int_{\substack{\eta \geq t \\ \hat{\eta} < t'}} \eta - t \, \mathrm{d}f_X + \int_{\substack{\eta \geq t \\ \hat{\eta} < t'}} t \, \mathrm{d}f_X - \int_{\substack{\eta < t \\ \hat{\eta} \geq t'}} t \, \mathrm{d}f_X + \int_{\substack{\eta < t \\ \hat{\eta} \geq t'}} t - \eta \, \mathrm{d}f_X
\end{aligned}$$

$$\leq \int_{\substack{\eta \geq t \\ \hat{\eta} < t'}} \eta - t + t' - \hat{\eta} \, \mathrm{d}f_X + 0 + \int_{\substack{\eta < t \\ \hat{\eta} \geq t'}} \hat{\eta} - t' + t - \eta \, \mathrm{d}f_X$$

$$\leq \|\eta - \hat{\eta}\|_1 \tag{6.19, cont'd.}$$

Where the 0 in the third line (and implicitly in the 5th) comes from the fact that the two regions have the same measure, by assumption on t and t'. This completes the first part of the statement.

Furthermore, we have assumed (Assumption 6.7) that for $h_{\overline{\theta}} = \mathbb{I}[\eta \geq \delta^*]$, $\Pr[|\eta(X) - \delta^*| \leq \epsilon] \leq k_1 \varepsilon$ for some k_1 and for all $\epsilon < \epsilon_\Omega$. Now, by Assumption 6.9, we can take n sufficiently large so that $\|\eta - \hat{\eta}_n\|_\infty < \epsilon_\Omega$. Thus, if $\eta(x) \geq \delta^* + \|\eta - \hat{\eta}_n\|_\infty$, we have $\hat{\eta}(x) \geq \delta^*$, so

$$\begin{aligned} TP(\hat{h}_{\theta^*}) &= \int_{\hat{\eta} \geq \delta^*} \eta \, \mathrm{d}f_X \geq \int_{\eta \geq \delta^*} \eta \, \mathrm{d}f_X \\ &\geq TP(h_{\theta^*}) - \Pr\left[\delta^* \leq h \leq \delta^* + \|\eta - \hat{\eta}_n\|_\infty\right] \\ &\geq TP(h_{\theta^*}) - k_1 \|\eta - \hat{\eta}_n\|_\infty \end{aligned} \tag{6.20}$$

Similar arguments apply for TN, which gives us the desired result. *Q.E.D.*

6.4 ELICITING LINEAR PERFORMANCE METRICS

It remains to discuss the elicitation of the performance metric ϕ. We use ‘ $*$ ’ and ‘ ^ ’ to denote entities corresponding to the oracle's true metric and the elicited metric, respectively. Suppose that the oracle's metric belongs to φ_{LPM} (Section 6.1.1). Again, without loss of generality, we may assume $\|\bar{\boldsymbol{m}}\| = 1$ and $\bar{m}_0 = 0$ (Section 6.2). According to Theorem 6.8, Algorithm 6.1 returns an approximate maximizer and the supporting hyperplane at that point. Since the performance metric is linear, the slope of the supporting hyperplane is the elicited performance metric, which is close to the true metric in the sup-norm. Given Assumption 6.7, we may have the following result.

Lemma 6.5. Let $\varphi_{LPM} \ni \phi^* = \mathbf{m}^*$ be the true performance metric. Given any $\epsilon > 0$, Algorithm 6.1 outputs a performance metric $\hat{\phi} = \hat{\boldsymbol{m}}$, such that $\|\mathbf{m}^* - \hat{\boldsymbol{m}}\|_\infty < \epsilon + \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}$, where k_0 and k_1 are the lower bound and upper bound constants, respectively, from Assumption 6.7.

Proof. Recall, the threshold estimator h_δ returns positive if $\eta(x) \geq \delta$, and negative otherwise. Let $\overline{\delta}$ be the threshold which maximizes performance with respect to ϕ, and $\overline{C}$ its confusion matrix. Following the framework of the proof of Theorem 6.8, we wish to show that if $|\delta' - \bar{\delta}| > \epsilon$, then $\|C(\delta') - \overline{C}\|_\infty > \frac{k_0^2 \epsilon^2}{4k_1}$. Suppose, without loss of generality, that $\delta' < \bar{\delta}$, and $\epsilon > \bar{\delta} - \delta'$. Recall, from assumption 6.7 that $\Pr[\eta(X) \in [\bar{\delta} - \frac{k_0}{2k_1}\epsilon, \bar{\delta}]] \leq k_0\epsilon/2$, but $\Pr[\eta(X) \in [\bar{\delta} - \epsilon, \bar{\delta}]] \geq k_0\epsilon$, and therefore

$$\Pr\left[\eta(X) \in [\bar{\delta} - \epsilon, \bar{\delta} - \tfrac{k_0}{2k_1}\epsilon]\right] \geq k_0\epsilon/2 \tag{6.21}$$

Denoting $\phi(C) = \langle \boldsymbol{m}, C \rangle$, and recalling that $\bar{\delta} = m_{00}/(m_{11} + m_{00})$, expanding the integral, we get

$$\begin{aligned}\phi(\bar{\delta}) - \phi(\delta') &= \int_{x:\delta' \leq \eta(x) \leq \bar{\delta}} m_{00}(1 - \eta(x)) - m_{11}\eta(x) \, \mathrm{d}f_X \\ &\geq \tfrac{k_0}{2}|\bar{\delta} - \delta'| \cdot \tfrac{k_0}{2k_1}|\bar{\delta} - \delta'| = \frac{k_0^2}{4k_1}|\bar{\delta} - \delta'|^2 \end{aligned} \tag{6.22}$$

where the first term is a lower bound on the area where the loss occurred is at least the amount of the second term. Therefore, if we have $|\phi(\overline{C}) - \phi(C(\delta'))| < \epsilon_\Omega$, then we must have $|\bar{\delta} - \delta'| < \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}$. Thus, if we are in a regime where the oracle is mis-reporting the preference ordering, it must be the case that the thresholds are sufficiently close to the optimal threshold.

Again, as in the proof of Theorem 6.8, our binary search closes in on a parameter θ' which has $\phi(C(\theta'))$ within ϵ_Ω of the optimum, but from the above discussion, this also implies that the search interval itself is close to the true value, and thus, the total error in the threshold is at most $\epsilon + \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}$. Since $\bar{\delta} = m_{00}/(m_{11} + m_{00})$, this bound extends to the cost vector. *Q.E.D.*

Since Assumption 6.7 determines that the feasible region is sufficiently well behaved near the optimum, k_0 is not too small relative to $\sqrt{k_1 \epsilon_\Omega}$, and thus the above bound is good given that we can never beat the oracle noise ϵ_Ω. A solution to this is given in Algorithm 6.2 which does not rely on Assumption 6.7 and depends only on on the properties of the feasible space. Its details will be presented in the next section.

6.4.1 Low-error LPM elicitation

Note, Algorithm 6.1 does not provide any guarantees on the output θ, only on $\phi(\theta)$. One example of where this breaks is when the region closely resembles that of a polygon. Note that we have assumed it is strictly convex. In this case, many possible choices of θ have similar values of $\phi(C_\theta)$, if the near-polygon has a large region which coincides with a level

Algorithm 6.2: Low-error LPM elicitation

```
Data: $\epsilon > 0$.
Query $\Omega(C_0, C_1)$;
Initialize: $\theta_L \leftarrow 0$, $\theta_R \leftarrow \frac{\pi}{2}$. ;        // Thus, $C(\theta_L) = C_1$ and $C(\theta_R) = C_0$.
while $\|\hat{C}(\theta_R) - \hat{C}(\theta_L)\|_\infty > \epsilon$ do
    Set $\theta \leftarrow \frac{\theta_L + \theta_R}{2}$.;
    Obtain the hyperplane $\boldsymbol{m}_\theta = (\sin\theta, \cos\theta)$.;
    Estimate the confusion matrix of the Bayes optimal classifier $h^*_{\boldsymbol{m}_\theta}$ from
     Proposition 6.4, and call it $\hat{C}(\theta)$.;
    Query $\Omega(\hat{C}^*(\theta), \overline{\hat{C}^*(\theta)})$, where $\overline{C} = (\pi, 1-\pi) - C$.;
    if $C_1 \succ C_0$ and $\hat{C}^*(\theta) \succ \overline{\hat{C}^*(\theta)}$  or  $C_1 \prec C_0$ and $\hat{C}^*(\theta) \prec \overline{\hat{C}^*(\theta)}$ then
        $\theta_L \leftarrow \theta$;
    else
        Set $\theta_R \leftarrow \theta$
    end
end
return $(m_{11}, m_{00})$ normal to the line spanned by $\hat{C}^*(\theta)$ and $\overline{\hat{C}^*(\theta)}$.
```

set of ϕ.

We introduce here an error-minimal algorithm which elicits a linear performance metric, without finding optimal classifiers. Recall, $h_1 \equiv 1$ is the classifier with $TP = \pi$, $TN = 0$ which we denote C_1, and $h_0 \equiv 0$ is the classifier with $TP = 0$ and $TN = 1 - \pi$, which we denote C_0. The idea behind the following algorithm is that, to elicit the linear metric with as much precision as possible, we try and find two classifiers with opposite confusion matrices which evaluate to the same value, and consider the line spanned by these confusion matrices as being a level set of ϕ.

It remains to prove that this minimizes the error for elicitation via estimating level sets, and analyze the accuracy of the algorithm.

Claim 6.10. Algorithm 6.2 finds the two points in $\mathcal{C}$ which are the furthest apart while being on the same level set of ϕ.

Proof. Note that since ϕ is linear, its level sets are hyperplanes. By the 180°-rotational symmetry and strict convexity of $\mathcal{C}$, the level set of ϕ with maximum length is the one passing through the center point of $\mathcal{C}$. Since we are estimating a linear performance metric by finding two points on some level set, the best we can do is take exactly these two opposite-angle points. Thus, of all algorithms which estimate ϕ via finding two points on some level set, 6.2 must be optimal. *Q.E.D.*

It remains to quantify the error rate. The remainder of this section gives the proof for the following theorem:

Theorem 6.11. Let $\varphi_{LPM} \ni \phi^* = \mathbf{m}^*$ be the true performance metric. Given any $\epsilon > 0$, Algorithm 6.2 outputs a performance metric $\hat{\phi} = \hat{\boldsymbol{m}}$, such that $\|\mathbf{m}^* - \hat{\boldsymbol{m}}\|_\infty < (2\epsilon + \epsilon_\Omega)/w$ where $0 < w < 1$ is a geometric property denoting 'width' of the feasible space $\mathcal{C}$.

Note that the parameter w is not small in practice, unless the feasible space is degenerate (η constant). To prove this result, we define a parameter $w(\mathcal{C})$ to be the "width" of $\mathcal{C}$, namely,

$$w(\mathcal{C}) := \inf \{w \geq 0 : \exists \mathbf{a}, b \text{ s.t. } \|\mathbf{a}\|_2 = 1 \text{ and } \forall C \in \mathcal{C}, b \leq \langle \mathbf{a}, C \rangle \leq b + w\} \tag{6.23}$$

Intuitively, this is the least distance between two parallel hyperplanes which bound $\mathcal{C}$. We state here a corollary of Höffding's inequality [167], which we will use in the remainder of this chapter.

Proposition 6.12. Let $(y_1, x_1, h(x_1))$, ..., $(y_n, x_n, h(x_n))$ be n i.i.d. samples from the joint distribution on Y, X, and $h(X)$. Then by Höffding's inequality,

$$\Pr\left[\left|\tfrac{1}{n}\textstyle\sum_{i=1}^n \mathbb{I}[h_i = y_i = 1] - TP(h)\right| \geq \epsilon\right] \leq 2e^{-2n\epsilon^2} . \tag{6.24}$$

The same holds for the analogous estimator on TN.

This proposition follows immediately from Höffding's bound.

Lemma 6.6. By Proposition 6.12, we can sample sufficiently so that $\|C(h_\theta) - \hat{C}(h_\theta)\|_\infty < \epsilon/2$. Doing so, algorithm 6.2 returns some $\mathbf{m}$ with $\|\mathbf{m} - \mathbf{m}^*\|_\infty < (2\epsilon + \epsilon_\Omega)/w(\mathcal{C})$, under the assumption $\|\mathbf{m}\|_2 = \|\mathbf{m}^*\|_2 = 1$ and $\|\nabla\phi\|_1 = 1$.

Proof. Without loss of generality, assume $\|\phi\|_1 = 1$, since everything is scale-invariant. Since the oracle $\Omega(C, C')$ returns correct values as long as $|\phi(C) - \phi(C')| > \epsilon_\Omega$, and we estimate $\|\hat{C} - C\|_\infty < \epsilon/2$, then, since ϕ is 1-Lipschitz, $\Omega(C, C') = \Omega(\hat{C}, \hat{C}')$ so long as $|\phi(C) - \phi(C')| > \epsilon_\Omega + \epsilon$.

Let $\overline{C}$ be the point on the boundary that we are searching for. Since the search interval in θ is narrowed to have measure ϵ, then the set of possible output confusion matrices is contained in a sup-norm ball of radius ϵ. Furthermore, we possibly suffer an extra error

$\epsilon + \epsilon_\Omega$ in either direction from mis-reporting. Therefore, the search-space is narrowed to a sup-norm ball of radius $2\epsilon + \epsilon_\Omega$. Since we are taking a normal vector perpendicular to $\overline{C}$ with respect to the midpoint of $\mathcal{C}$, we get a confidence interval of size $(2\epsilon + \epsilon_\Omega)/w(\mathcal{C})$ in the normal vector, since we must scale up the normal vector. Note that $w(\mathcal{C}) < 1$. *Q.E.D.*

6.5 ELICITING LINEAR FRACTIONAL PERFORMANCE METRICS

We recall, a linear fraction performance metric (LFPM) is one of the form

$$\phi((TP, TN)) = \frac{p_{11}TP + p_{00}TN + p_0}{q_{11}TP + q_{00}TN + q_0}\ , \tag{6.25}$$

as defined in equation (6.5). We assume that p_{11}, p_{00}, q_{11}, and q_{00} are not simultaneously zero and ϕ is bounded over $\mathcal{C}$. As scaling and shifting does not change the linear-fractional form, without loss of generality, we may take $\phi(C) \in [0, 1]\ \forall\ C \in \mathcal{C}$ and both the numerator and the denominator to be positive.

Assumption 6.13. Let $\phi \in \varphi_{LFPM}$ as defined in (6.5). We assume that $p_{11}, p_{00} \geq 0$, $p_{11} \geq q_{11}$, $p_{00} \geq q_{00}$, $p_0 = 0$, $q_0 = (p_{11} - q_{11})\pi + (p_{00} - q_{00})(1 - \pi)$, and $p_{11} + p_{00} = 1$.

Proposition 6.14. If Assumption 6.13 holds, then we may assume without loss of generality that $\phi \in \varphi_{LFPM}$ is bounded in $[0, 1]$ and simultaneously monotonically increasing in TP and TN.

Proof. We want $\phi(C)$ to be monotonic in TP, TN and bounded. By definition, the numerator will be non-negative. To show the denominator is non-negative, we have that, by definition of q_0 and π,

$$q_0 = (p_{11} - q_{11})\pi + (p_{00} - q_{00})(1 - \pi) \geq TP(p_{11} - q_{11}) + TN(p_{00} - q_{00}) \tag{6.26}$$

Which implies that the denominator is lower-bounded by $p_{11}TP + p_{00}TN \geq 0$. Thus, ϕ will be non-negative. Note also that this is exactly the numerator, since $p_0 = 0$, and so the ratio will be upper-bounded by 1. It remains to show that the function is monotone non-decreasing. Differentiating ϕ with respect to TP, we have

$$\begin{aligned}\frac{\partial\phi(C)}{\partial TP} &= \frac{p_{11}}{q_{11}TP + q_{00}TN + q_0} - \frac{q_{11}(p_{11}TP + p_{00}TN + p_0)}{(q_{11}TP + q_{00}TN + q_0)^2} \\ &= \tfrac{1}{(q_{11}TP + q_{00}TN + q_0)^2}\left(p_{11}(q_{11}TP + q_{00}TN + q_0) - q_{11}(p_{11}TP + p_{00}TN + p_0)\right) \\ &= \tfrac{1}{(q_{11}TP + q_{00}TN + q_0)^2}\left(p_{11}q_{00}TN - p_{00}q_{11}TN + p_{11}q_0 - q_{11}p_0\right) \qquad (6.27)\end{aligned}$$

Now, by the assumption, $p_0 = 0$, and $q_0 = \pi(p_{11} - q_{11}) + (1-\pi)(p_{00} - q_{00})$. But by definition, $(1 - \pi) \geq TN$, and therefore

$$\begin{aligned}\frac{\partial \phi(C)}{\partial TP} &\geq \tfrac{1}{(q_{11}TP + q_{00}TN + q_0)^2} \ \left(TN(p_{11}q_{00} - p_{00}q_{11}) + \pi p_{11}(p_{11} - q_{11}) + TNp_{11}(p_{00} - q_{00})\right) \\ &= \tfrac{1}{(q_{11}TP + q_{00}TN + q_0)^2} \ \left(p_{00}TN(p_{11} - q_{11}) + \pi p_{11}(p_{11} - q_{11})\right) \qquad (6.28)\end{aligned}$$

which is a sum of non-negative terms, by the assumption. A similar argument gives the same for the derivative in TN, and therefore, the function is non-decreasing. *Q.E.D.*

We claim Assumption 6.13 is reasonable, as it is satisfied by the examples outlined in (6.6) and Koyejo et al. [157], and furthermore, they allow to reduce the dimension of the parameter space for the elicitation problem.

In the remainder of this chapter, we give our method for eliciting the LFPM, and its optimal classifier. The idea is to first obtain hyperplanes at the maximizer on the upper boundary and at the minimizer on the lower boundary. This gives two systems of nonlinear equations, both of which have one degree of freedom. The choice of the elicited metric is the one where the solutions to the two systems match pointwise on a number of confusion matrices. This is presented formally below.

Suppose that the oracle's metric $\phi^* \in \varphi_{LFPM}$ is defined as:

$$\phi^*(C) = \frac{p^*_{11}TP + p^*_{00}TN}{q^*_{11}TP + q^*_{00}TN + q^*_0}. \qquad (6.29)$$

Let $\overline{\tau}^*$ and $\underline{\tau}^*$ be the maximum and the minimum value of ϕ^* over $\mathcal{C}$ respectively, *i.e.*

$$\underline{\tau}^* \leq \phi^*(C) \leq \overline{\tau}^* \qquad \text{for all } C \in \mathcal{C} \qquad (6.30)$$

Under Assumption 6.5, there exists a hyperplane tangent to $\mathcal{C}$ at $(\overline{TP}^*, \overline{TN}^*)$ on the upper boundary $\partial\mathcal{C}_+$, denoted as follows:

$$\overline{\ell}^*_f := \{(p^*_{11} - \overline{\tau}^* q^*_{11})TP + (p^*_{11} - \overline{\tau}^* q^*_{11})TN + (-\overline{\tau}^* q^*_0) = 0\} \qquad (6.31)$$

Similarly, we denote as follows the tangent hyperplane at $(\underline{TP}^*, \underline{TN}^*)$ on the lower boundary $\partial\mathcal{C}_-$:

$$\tilde{\ell}^*_f := (p^*_{11} - \tilde{\tau}^* q^*_{11})TP + (p^*_{00} - \tilde{\tau}^* q^*_{00})TN + (-\tilde{\tau}^* q^*_0) = 0, \qquad (6.32)$$

See Figure 6.1(c) for an illustration. Since LFPM are quasiconcave, Algorithm 6.1 returns a hyperplane $\bar{\ell} := \overline{m}_{11}TP + \overline{m}_{00}TN - \overline{C}_0 = 0$ (via Theorem 6.8), where $\overline{C}_0 = \overline{m}_{11}\overline{TP} +$

$\overline{m}_{00}\overline{TN}$. This is equivalent to $\overline{\ell}^*_f$ up to a constant multiple; therefore, the elicited metric is the solution to the following non-linear system of equations:

$$p^*_{11} - \overline{\tau}^* q^*_{11} = \alpha \overline{m}_{11}, \quad p^*_{00} - \overline{\tau}^* q^*_{00} = \alpha \overline{m}_{00}, \quad -\overline{\tau}^* q^*_0 = -\alpha \overline{C}_0, \tag{6.33}$$

where $\alpha \geq 0$, because LHS and $\overline{m}$'s are non-negative. Additionally, we ignore the case when $\alpha = 0$, since this would imply a constant ϕ. Next, we may divide the above equations by $\alpha > 0$ on both sides so that all the coefficients $\overline{p}^*$'s and $\overline{q}^*$'s are factored by α. This does not change ϕ^*; therefore, the system of equations becomes:

$$p'_{11} - \overline{\tau}^* q'_{11} = \overline{m}_{11}, \quad p'_{00} - \overline{\tau}^* q'_{00} = \overline{m}_{00}, \quad -\overline{\tau}^* q'_0 = -\overline{C}_0. \tag{6.34}$$

Notice that none of the sufficient conditions in Assumption 6.13 are changed except $p'_{11} + p'_{00} = 1$, which was not used in the proof of monotonicity. Since the LFPM is also quasiconvex, Algorithm 6.1 modified also gives us a hyperplane (via Theorem 6.8) $\tilde{\ell} := \underline{m}_{11} tp + \underline{m}_{00} tn - \underline{C}_0 = 0$, where $\underline{C}_0 = \underline{m}_{11}\underline{TP} + \underline{m}_{00}\underline{TN}$. This is equivalent to $\tilde{\ell}^*_f$ up to a constant multiple; therefore, the eliciated metric is also the solution to the following system of non-linear equations:

$$p^*_{11} - \tilde{\tau}^* q^*_{11} = \kappa \underline{m}_{11}, \quad p^*_{00} - \tilde{\tau}^* q^*_{00} = \kappa \underline{m}_{00}, \quad -\tilde{\tau}^* q^*_0 = -\kappa \underline{C}_0, \tag{6.35}$$

where $\kappa \leq 0$ since LHS is positive, but $\underline{m}$'s are negative. Again, we may assume $\kappa < 0$. By dividing the above equations by $-\kappa$ on both sides, all the coefficients p^*'s and q^*'s are factored by $-\kappa$. This does not change ϕ^*; therefore, the system of equations becomes the following:

$$p''_{11} - \tilde{\tau}^* q''_{11} = \underline{m}_{11}, \quad p''_{00} - \tilde{\tau}^* q''_{00} = \underline{m}_{00}, \quad -\tilde{\tau}^* q''_0 = -\underline{C}_0. \tag{6.36}$$

We can solve the system of equations (6.34) and (6.36) and thus elicit the performance metric, provided we know p'_{11} (p''_{11}) or p'_{00} (p''_{00}). The solution to the elicitation task, given p'_{11}, is discussed below.

Proposition 6.15. Under the sufficient conditions of Assumption 6.13, knowledge of p'_{11} (or, p'_{00}) solves the system of equations in (6.34) and elicits the performance metric as follows:

$$p'_{00} = 1 - p'_{11}, \quad q'_0 = \overline{C}_0 \frac{P'}{Q'}, \quad q'_{11} = (p'_{11} - \overline{m}_{11}) \frac{P'}{Q'}, \quad q'_{00} = (p'_{00} - \overline{m}_{00}) \frac{P'}{Q'}, \tag{6.37}$$

where $P' = p'_{11}\pi + p'_{00}(1-\pi)$ and $Q' = P' + \overline{C}_0 - \overline{m}_{11}\pi - \overline{m}_{00}(1-\pi)$.

Proof. In the following, we denote $TP = C_{11}$ and $TN = C_{00}$. Since the linear fractional matrix is monotonically increasing in C_{11} and C_{00}, it is maximized at the upper boundary $\partial\mathcal{C}_+$. Hence $m_{11} \geq 0$ and $m_{00} \geq 0$. So, after running Algorithm 6.1, we get a hyperplane such that

$$p_{11} - \tau q_{11} = \alpha m_{11}, \quad p_{00} - \tau q_{00} = \alpha m_{00}, \quad p_0 - \tau q_0 = -\alpha \underbrace{(m_{11}C^*_{11} + m_{00}C^*_{00})}_{=:C_0}. \tag{6.38}$$

Since $p_{11} - \overline{\tau} q_{11} \geq 0$ and $m_{11} \geq 0$, $\Rightarrow \alpha \geq 0$. As discussed, we avoid the case when $\alpha = 0$. Hence, we have that $\alpha > 0$, and dividing every one of p_{11}, p_{00}, p_0, q_{11}, q_{00}, and q_0 by α leaves the problem unchanged. Thus, we wish to solve for

$$p_{11} - \overline{\tau} q_{11} = m_{11}, \quad p_{00} - \overline{\tau} q_{00} = m_{00}, \quad p_0 - \overline{\tau} q_0 = -C_0 \tag{6.39}$$

From the rightmost equation, we have that $\overline{\tau} = \frac{C_0 + p_0}{q_0}$. Combining with the rest gives

$$q_0 p_{11} - (C_0 + p_0) q_{11} = m_{11} q_0, \quad q_0 p_{00} - (C_0 + p_0) q_{00} = m_{00} q_0. \tag{6.40}$$

and

$$\begin{aligned} q_0 &= (p_{11} - q_{11})\pi + (p_{00} - q_{00})(1-\pi) + p_0 \\ \Rightarrow q_{11} &= \frac{p_{00}(1-\pi) - q_{00}(1-\pi) + p_{11}\pi - q_0 + p_0}{\pi}, \end{aligned} \tag{6.41}$$

from which we conclude then

$$\begin{aligned} q_0 &= \frac{(C_0 + p_0)[p_{00}(1-\pi) + p_{11}\pi + p_0]}{p_{11}\pi + p_{00}(1-\pi) + p_0 + C_0 - m_{11}\pi - m_{00}(1-\pi)}, \\ q_{00} &= \frac{(p_{00} - m_{00})[p_{00}(1-\pi) + p_{11}\pi + p_0]}{p_{11}\pi + p_{00}(1-\pi) + p_0 + C_0 - m_{11}\pi - m_{00}(1-\pi)}, \\ q_{11} &= \frac{(p_{11} - m_{11})[p_{00}(1-\pi) + p_{11}\pi + p_0]}{p_{11}\pi + p_{00}(1-\pi) + p_0 + C_0 - m_{11}\pi - m_{00}(1-\pi)}. \end{aligned} \tag{6.42}$$

Letting $P := p_{00}(1-\pi) + p_{11}\pi + p_0$, and $Q := P + C_0 - m_{11}\pi - m_{00}(1-\pi)$, we have

$$q_0 = (C_0 + p_0)\frac{P}{Q}, \quad q_{11} = (p_{11} - m_{11})\frac{P}{Q}, \quad q_{00} = (p_{00} - m_{00})\frac{P}{Q}. \tag{6.43}$$

Now using sufficient conditions, we have $p_0 = 0$. The final solution is the following:

```
Algorithm 6.3: Grid Search for Best Ratio
  Data: $k, \Delta$.
  initialize: $\sigma_{opt} \leftarrow \infty, p'_{11,opt} = p'_{11} \leftarrow 0$;
  Generate $C_1, ..., C_k$ on $\partial C_+$ and $\partial C_-$ ;    // Boundaries defined in Section 6.2
  for $p'_{11} = 0,\ \Delta,\ 2\Delta,\ \dots,\ 1$ do
      Compute $\phi'$, $\phi''$ using Proposition 6.15. Compute array $r = [\frac{\phi'(C_1)}{\phi''(C_1)}, ..., \frac{\phi'(C_k)}{\phi''(C_k)}]$;
      Set $\sigma = \mathrm{std}(r)$;
      if $\sigma < \sigma_{opt}$ then $\sigma_{opt} \leftarrow \sigma$ and $p'_{11,opt} \leftarrow p'_{11}$;
  end
  return $p'_{11,opt}$, $\sigma_{opt}$.
```

$$q_0 = C_0 \frac{P}{Q}, \quad q_{11} = (p_{11} - m_{11}) \frac{P}{Q}, \quad q_{00} = (p_{00} - m_{00}) \frac{P}{Q}, \tag{6.44}$$

where $P := p_{11}\pi + p_{00}(1-\pi)$ and $Q := P + C_0 - m_{11}\pi - m_{00}(1-\pi)$. We have taken $p_{11} + p_{00} = 1$, but the original $p'_{11} + p'_{00} = \frac{1}{\alpha}$. Therefore, we learn $\hat{\phi}(C)$ such that such that $\hat{\phi}(C) = \alpha\phi(C)$. *Q.E.D.*

System (6.36) can be solved analogously to Proposition 6.15. We can elicit metrics ϕ' and ϕ'' such that $\phi'(C)/\alpha = \phi^*(C) = -\phi''(C)/\kappa$ provided the true ratio of p'_{11} to p'_{00} and p''_{11} to p''_{00} are known. Since we can generate many confusion matrices on the boundaries $\partial\mathcal{C}_+$ and $\partial\mathcal{C}_-$, we can learn an estimate of the true ratio for p'_{11} to p'_{00} using the grid search based Algorithm 6.3. We start with $p'_{11} = 0$. Then at each iteration, increase it by Δ, solve both the systems (6.34) and (6.36), and compute the ratio of ϕ' to ϕ'' on the set of confusion matrices. We pick the value of p'_{11} for which the standard deviation of the ratio is minimum. These computations are independent of the oracle queries, and thus can be computed offline in many different ways. One of the simplest ways is grid search, which is sufficient for the elicitation task. We set the final elicited metric $\hat{\phi} = \phi'$, which is obtained corresponding to the output of Algorithm 6.3. Thus, $\hat{\phi}$ is a constant multiple of the true metric ϕ^*.

CHAPTER 7: ONLINE REVENUE MAXIMIZATION FOR SERVER PRICING[1]

In this chapter, we solve the problem of maxmizing revenue on a single server, with *i.i.d.* jobs arriving arriving from an unknown distribution. We reduce the problem to solving a Markov Decision Process with an exponentially large action space, and show how to reduce the dimension of the action space to get efficient computation. We restrict ourselves to *posted pricing* mechanisms, which simply post prices, and rely on the agents/buyers to determine what they wish to purchase on the server. This also means that we do not observe the job's parameters directly, but only what resources they chose to purchase. We require that at each time period, the server post prices for each job length starting at the earliest available time, and we find additive approximations to optimal pricing policies within this family.

In our setting, we assume time is allocated non-preemptively, and therefore the ressources have strong complementarities. Furthermore, since the supply (server capacity) is limited, any mechanism trades immediate revenue for future supply. We are considering a "real-time" model, in the sense that time is progressing as the jobs execute while new jobs arrive, and more time becomes available to schedule future jobs.

Model, informally. We assume time is discrete. At every time step, an agent arrives on the server, with a value V, length requirement L, and maximum delay D. For each job, the triple (V, L, D) is drawn *i.i.d.* from an underlying, unknown, distribution. The job wishes to be scheduled for at least L consecutive time slots, no more than D time units after its arrival, and wishes to pay no more than V. Jobs are assumed to have quasi-linear utility in money, and so prefer the least-price interval within their constraints. The mechanism designer never learns the parameters of the job. Instead, a price menu of (length,price) pairs, and the minimum available delay s, are posted. The job accepts to be scheduled so long as $D \geq s$, and there is some (length,price) pair in the menu of length at least L and price at most V. We note that the pricing scheme can be dynamic, changing through time. If, at time epoch t, an agent chooses option (ℓ, π_ℓ), then she pays π_ℓ and her job will be allocated to the interval $[t+s, t+s+\ell]$. She will choose the option which minimizes π_ℓ. Throughout this paper we assume that the random variables L, V, D are discrete, and have finite support, unless specified differently.

[1] This chapter is based on collaboration with Federico Fusco, and Stefano Leonardi, Yishay Mansour, and Ruta Mehta [98].

7.1 OVERVIEW AND MODEL

We solve this problem by modelling the server state as a Markov Chain, and the price menus, as the action space in the associated Markov Decision Process (MDP), with details given in Section 7.1.1. Given a price menu (length,price) and a state (minimum available delay) s at time t, the probability of transition to any other state at time $t+1$ is obtained from the distribution of the job's parameters. The revenue maximizing pricing strategy can be efficiently computed via backwards induction, when we can assume that optimal prices are monotone non-decreasing in length. The details of this are given in Section 7.2. We also extend, in the final subsections of Section 7.2, the result to the infinite-horizon setting, and to continuous random variables.

The result required an assumption that the optimal prices were monotone. We show in Section 7.2.3, that this assumption holds under a distributional assumption, which we show is satisfied when the jobs' valuation follows a log-concave distribution, parametrized by length. Log-concave distributions are also known in some fields as distributions which have a monotone hazard rate. This monotonicity implies that jobs will truthfully choose their desired length, rather than buy more time on the server for less money.

We finally investigate, in Section 7.4, the robustness of the pricing strategy. We first show that a near optimal solution is still obtained when the distribution is known with a certain degree of uncertainty. We complement this result by analyzing the performance of the proposed pricing strategy when the distribution is only known from samples collected through the observations of the agents' decisions. We provide a truthful posted price ε-approximate mechanism if the number of samples is polynomial in $1/\varepsilon$ and the size of the support of the distribution.

7.1.1 Model

Notation. In what follows, the variables t, ℓ or L, v or V, and d or D are reserved for describing the parameters of a job that wishes to be scheduled. Respectively, they represent the arrival time t, required length ℓ, value v, and maximum allowed delay d. The lowercase variables represent fixed values, whereas the uppercase represent random variables. Script-uppercase letters $\mathcal{L}, \mathcal{V}, \mathcal{D}$ represent the supports of the distributions on L, V, and D, respectively; and the bold-uppercase letters $\mathbb{L}, \mathbb{V}, \mathbb{D}$ represent the maximum values in these respective sets. Finally, π is reserved for pricing policy, whereas p is reserved for probabilities.

Single-Machine, Non-Preemptive, Job Scheduling. A sequence of random jobs wish to be scheduled on a server, non-preemptively, for a sufficiently low price, within a time constraint. Formally, at every time step t, a single job with parameters (L, V, D) is drawn from an underlying distribution Q over the space $\mathcal{L} \times \mathcal{V} \times \mathcal{D}$. It wishes to be scheduled for a price $\pi \leq V$ in an interval $[a, b]$ such that $a - t \leq D$ and $b - a \geq L$.

Price Menus. Our goal is to design a take-it-or-leave-it, posted-price mechanism which maximizes expected revenue. At each time period, the mechanism posts a "price menu" and an earliest-available-time s_t, indicating that times t through $t + s_t - 1$ have already been scheduled. (s_t will henceforth be referred to as the *state* of the server.) We let $\mathcal{S} := \{0, \ldots, \mathbb{D} + \mathbb{L}\}$ to be the set of all possible states. The state of the server at a given time t is naturally a random variable which depends on the earlier jobs and on the adopted policy π. As before, we will denote with s or s_t the fixed value, and with S or S_t the corresponding random variable. The price menu will be given by the function $\pi : [T] \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$, i.e., if we are a time t and the server is in state s_t, then the prices are set according to $\pi_t(s_t, \cdot) : \mathcal{L} \to \mathbb{R}$. The reported pair $(\pi_t(s_t, \cdot), s_t)$ is computed by the scheduler's strategy, which we determine in this paper. Once this is posted, a job (L, V, D) is then sampled *i.i.d.* from the underlying distribution Q.

If $V \geq \pi_t(s_t, \ell)$ for some $\ell \geq L$, and $D \geq s_t$, then the job accepts the schedule, and reports the length $\ell \geq L$ which minimize price. Otherwise, the job reports $\ell = 0$ and is not scheduled. To guarantee truthfulness, it suffices to have $\pi_t(s, \cdot)$ be monotonically non-decreasing for every state s: the agent would not want a longer interval since it costs more, and would not want one of the shorter intervals since they cannot run the job. It should be clear that the mechanism's strategy is to always report monotone non-decreasing prices, as a decrease in the price menu will only cause more utilization of the server, without accruing more revenue. The main technical challenge in this paper, then, is to show that under some assumptions, the optimal strategy is monotone non-decreasing, and efficiently computable.

Revenue Objective. Revenue can be measured in either a *finite* or an *infinite discounted* horizon. In the former (finite) case, only T time periods will occur, and we seek to maximize the expected sum of revenue over these periods. In the infinite-horizon setting, future revenue is discounted, at an exponentially decaying rate. Formally, revenue at time t is worth a γ^t fraction of revenue at time 0, for some fixed $\gamma < 1$. See Section 7.3.2. Recall that the job parameters are drawn independently at random from the underlying distribution, so the scheduler can only base their "price menu" on the state of the system and the current time. Thus, the only realistic strategy is to fix a state-and-time-dependent pricing policy

$\pi : [T] \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$, "$\pi_t(s, \ell)$", where $[T] := \{0, 1, \ldots, T\}$.

Let $\mathcal{X} = \{\mathcal{X}_1 := (1, L_1, V_1, D_1),\ \mathcal{X}_2 := (2, L_2, V_2, D_2),\ \mathcal{X}_3, \ldots\}$ be the random sequence of jobs arriving, sampled *i.i.d.* from the underlying distribution. Let $\pi : [T] \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$ be the pricing policy. We denote as $\mathsf{Rev}_t(\mathcal{X}, \pi)$ the revenue earned at time t with policy π and sequence $\mathcal{X}$. If $\mathcal{X}_t$ does not buy, then $\mathsf{Rev}_t(\mathcal{X}, \pi) = 0$, and otherwise, it is equal to $\pi_t(s_t, L_t)$. We denote as CmlRev_T the total (cumulative) revenue earned over the T periods. Thus,

$$\mathsf{CmlRev}_T(\mathcal{X}, \pi) := \textstyle\sum_{t=0}^{T} \mathsf{Rev}_t(\mathcal{X}, \pi). \tag{7.1}$$

We will also need the expected-future-revenue, given a current time and server state, which we will denote as follows:

$$U_t^{\pi}(s) = \mathbb{E}_{\mathcal{X}_{\geq t}} \left[\textstyle\sum_{i=t}^{T} \mathsf{Rev}_i(\pi, \mathcal{X}) \,\middle|\, S_t = s \right], \tag{7.2}$$

The subscript of the expectation $\mathcal{X}_{\geq t}$ denotes that we consider only jobs arriving from time t onward. Our objective is to find the pricing policy π which maximizes $U_0^{\pi}(s = 0)$. Call this π^*, and denote the expected revenue under π^* as $U_t^*(\cdot)$.

7.2 BAYES-OPTIMAL STRATEGIES FOR SEVER PRICING, ASSUMING MONOTONICITY

In this section we seek to compute an optimal monotone pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$ which maximizes revenue in expectation over T jobs sampled *i.i.d.* from an underlying known distribution Q. This is extended to the infinite-horizon, discounted, setting in Section 7.3.2.

We first model the problem of maximizing the revenue in online server pricing as a Markov Decision Process that admits an efficiently-computable, optimal pricing strategy. The main contribution of this section is to show that, for a natural assumption on the distribution Q, the optimal policy is monotone. We recall that this allows us to derive truthful Bayes-optimal mechanisms.

7.2.1 Markov Decision Processes.

We show that the theory of *Markov Decision Processes* is well suited to model our problem. A Markov Decision Process is, in its essence, a Markov Chain whose transition probabilities depend on the *action* chosen at each state, and where to each transition is assigned a reward. A *policy* is then a function π mapping states to actions. In our setting, the states are the states of the system outlined in Section 7.1.1 (i.e., the possible delays before the earliest available time on the server), and the actions are the "price menus." At every state s, a job of a random length arrives, and with some probability, chooses to be scheduled, given the choice of prices. The next state is either $\max\{s-1,0\}$, if the job does not choose to be scheduled (since we have moved forward in time), or $s+\ell-1$, if a job of length ℓ is scheduled, since we have occupied ℓ more units. The transition probabilities depend on the distribution of job lengths, and the probability that a job accepts to be scheduled given the pricing policy (action). Formally,

$$\mathbb{P}[s_{t+1}=s_t+\ell-1]=\begin{cases}\mathbb{P}\left[L_t=\ell, V_t\geq\pi_t(s_t,\ell), D_t\geq s_t+\ell\right] & \text{if } \ell\geq 1\\ 1-\sum_{k\geq 0}\mathbb{P}[s_{t+1}=s_t+k] & \text{if } \ell=0\end{cases}\tag{7.3}$$

(Transitions to state "-1" should be read as transitions to state "0".) Note that a job of length ℓ may choose to purchase an interval of length greater than ℓ, which would render these transition probabilities incorrect. However, this may only happen if the larger interval is more affordable. It is therefore in the scheduler's interest to guarantee that $\pi_t(s,\cdot)$ in monotone non-decreasing in ℓ, which incentivizes truthfulness, since this increases the amount of server-time available, without affecting revenue. Thus we restrict ourselves to this case.

It remains to define the transition rewards. They are simply the revenue earned. Formally, a transition from state s_t to $s_t+\ell-1$ incurs a reward of $\pi_t(s,\ell)$, whereas a transition from state s_t to s_t-1 incurs 0 reward. We wish to compute a policy π in such a way as to maximize the expected cumulative revenue, given as the (possibly discounted) sum of all transition rewards in expectation.

7.2.2 Solving for the Optimal Policy with Distributional Knowledge

In this section, we present a modified MDP whose optimal policies can be efficiently computed, and show that these policies are optimal for the original MDP. In this section, we assume that the mechanism designer is given access to the underlying distribution Q. However, in the following sections, we will show that if the distribution Q is estimated from samples, then solving for the MDP on this estimated distribution is sufficient to ensure sufficiently good revenue guarantees.

Since the problem has been modelled as a Markov Decision Process (MDP), we may rely on the wealth of literature available on MDP solutions, in particular we will leverage the *backwards induction* algorithm (BIA) of [168] Section 4.5, presented here as Algorithm 7.1. We will however need to ensure that this standard algorithm (i) runs efficiently, and (ii) returns a monotone pricing policy.

Algorithm 7.1: Backwards Induction for Finite-Horizon MDP's [168], section 4.5

```
Data: MDP with states $\mathcal{S}$, actions $\mathcal{A}$, and rewards $R$; and a horizon $T$.
Result: Optimal policy $\pi^* : [T] \times \mathcal{S} \to \mathcal{A}$.
begin
    Initialize $U_T^*(s) \leftarrow 0$ for all $s \in \mathcal{S}$.
    for $t$ from $T-1$ to 0, descending do
        for $s \in \mathcal{S}$ do
            $U_t^*(s) \leftarrow \max_{a \in \mathcal{A}} \big\{ \sum_{s' \in \mathcal{S}} \mathbb{P}[s_{t+1} = s' | s, a] \big( \text{Reward}(s \to s' | a) + U_{t+1}^*(s') \big) \big\}$
            $\pi^*(t, s) \leftarrow \arg\max_{a \in \mathcal{A}} \big\{ \sum_{s' \in \mathcal{S}} \mathbb{P}[s_{t+1} = s' | s, a] \big( \text{Reward}(s \to s' | a) + U_{t+1}^*(s') \big) \big\}$
        end
    end
    return $\pi$
end
```

Note that past prices do not contribute to future revenue insofar as the current state remains unchanged. Thus, to compute optimal current prices, we need only know the current state and expected future revenue. This allows us to use the BIA. The idea is to compute the optimal time-dependent policy, and the incurred expected reward, for shorter horizons, then use this to recursively compute the optimal policies for longer horizons.

The total runtime of the BIA is $O(T|\mathcal{S}||\mathcal{A}|)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the action and state spaces, respectively. Note that the dependence on T is unavoidable, since any optimal policy must be time-dependent. Recall that $\mathbb{L}$ and $\mathbb{D}$ denote the maximum values that L and D can take, respectively, and $\mathcal{V}$ is the set of possible values that V can take. Denote $\mathbb{K} := \max\{\mathbb{D} + \mathbb{L}, |\mathcal{V}|\}$. If we define the action space naïvely, we have $|\mathcal{S}| = \mathbb{D} + \mathbb{L} \leq \mathbb{K}$, and $|\mathcal{A}| \leq \mathbb{K}^{\mathbb{L}}$. Thus, a naïve definition of the MDP bounds the runtime at $\mathbb{K}^{O(\mathbb{K})}$, which is far from efficient. Requiring monotonocity only affects lower-order terms.

Modified MDP. To avoid this exponential dependence, we can be a little more clever about the definition of the state space: instead of states being the possible server states, we define our state space as possible (state, length) pairs. Thus, when the MDP is in state (s, ℓ), the server is in state s, and a job of length ℓ has been sampled from the distribution. Our action-space then is simply the possible values of $\pi_t(s, \ell)$, and the rewards become:

$$R((s, \ell) \to (s', \ell') | \pi) = \begin{cases} \pi_t(s, \ell) & \text{if } s' = s + \ell - 1 \\ 0 & \text{otherwise} \end{cases}, \tag{7.4}$$

Algorithm 7.2: Optimal policy in finite horizon

```
Data: Distribution $Q$, $\mathbb{L}$, $\mathbb{V}$, $\mathcal{S}$ and horizon $T$.
Result: Optimal policy $\pi^* : [T] \times \mathcal{S} \times \mathbb{L} \to \mathcal{R}$.
Initialize $U_T^*(s) \leftarrow 0$ for all $s \in \mathcal{S}$, and $u_T^*(s,\ell) \leftarrow 0$ for all $s \in \mathcal{S},\ \ell \in \mathbb{L}$.
for $t$ from $T-1$ to 0, descending do
    for $s \in \mathcal{S}$ do
        for $\ell \in \mathbb{L}$ do
            $u_t^*(s,\ell) \leftarrow \max_{\mu\in\mathbb{V}} \Big\{\mathbb{P}[V \geq \mu, D \geq s | L = \ell] \times \big(\mu + U_{t+1}^*(s+\ell-1) - U_{t+1}^*(s-1)\big) + U_{t+1}^*(s-1)\Big\}$
            $\pi_t^*(s,\ell) \leftarrow \arg\max_{\mu\in\mathbb{V}} \Big\{\mathbb{P}[V \geq \mu, D \geq s | L = \ell] \times \big(\mu + U_{t+1}^*(s+\ell-1) - U_{t+1}^*(s-1)\big) + U_{t+1}^*(s-1)\Big\}$
        end
        $U_t^*(s) \leftarrow \sum_{\ell\in\mathbb{L}} \mathbb{P}[L = \ell] u_t^*(s,\ell)$.
    end
end
return $\pi^*$
```

and the transition probabilities become

$$\mathbb{P}[(s,\ell) \to (s',\ell')|\pi] = \begin{cases} \mathbb{P}[V \geq \pi_t(s,\ell), D \geq s | L = \ell]\mathbb{P}[L' = \ell'] & \text{if } s' = s + \ell - 1 \\ \mathbb{P}[V < \pi_t(s,\ell) \text{ or } D < s | L = \ell]\mathbb{P}[L' = \ell'] & \text{if } s' = s - 1 \\ 0 & \text{otherwise} \end{cases}. \tag{7.5}$$

Therefore, we get $|\mathcal{S}| = (\mathbb{D}+\mathbb{L}) \cdot \mathbb{L} \leq \mathbb{K}^2$, and $|\mathcal{A}| \leq \mathbb{K}$. Thus, the runtime of the algorithm becomes $O(T\mathbb{K}^3)$. A full description of the procedure is given as Algorithm 7.2. It remains to prove that it is correct. We begin by claiming that these two MDPs are equivalent in the following sense:

Lemma 7.1. For any fixed pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$,

$$U_t^\pi(s) = \mathbb{E}_L\left[u_t^\pi(s,L)\right], \forall t \in T,\ s \in \mathcal{S}, \tag{7.6}$$

where the $U_t^\pi(\cdot)$'s are as in (7.2), and the $u_t^\pi(\cdot,\cdot)$'s are from the modified MDP.

Proof. The statement is true for $t = T$ since in that case everything is zero. Suppose $\mathbb{E}_{L'}\left[u_{t+1}^\pi(s,L')\right] = U_{t+1}^\pi(s)$ for all s. For the fixed policy π, we define $\boldsymbol{p}_{t,s}^\ell := \mathbb{P}[V \geq$

$\pi_t(s, \ell), D \geq s | L = \ell]$. Then,

$$\begin{aligned}\mathbb{E}_L \left[u_t^\pi(s, L)\right] &= \sum_{\ell \in \mathcal{L}} \mathbb{P}[L = \ell] u_t^\pi(s, \ell) \\ &= \sum_{\ell \in \mathcal{L}} \mathbb{P}[L = \ell] \Big(\pi_t(s, \ell) \boldsymbol{p}_{t,s}^\ell + \boldsymbol{p}_{t,s}^\ell \, \mathbb{E}_{L'} \left[u_{t+1}^\pi(s + \ell - 1, L')\right] \\ &\qquad\qquad + (1 - \boldsymbol{p}_{t,s}^\ell) \, \mathbb{E}_{L'} \left[u_{t+1}^\pi(s - 1, L')\right] \Big) \\ &= \sum_{\ell \in \mathcal{L}} \mathbb{P}[L = \ell] \Big(\pi_t(s, \ell) \boldsymbol{p}_{t,s}^\ell + \boldsymbol{p}_{t,s}^\ell u_{t+1}^\pi(s + \ell - 1, L') \\ &\qquad\qquad + (1 - \boldsymbol{p}_{t,s}^\ell) u_{t+1}^\pi(s - 1, L') \Big) \\ &= \mathbb{E}_{\mathcal{X}} \left[\mathsf{Rev}_t(\pi, \mathcal{X}) + U_{t+1}^\pi(S_{t+1}(S_t, \mathcal{X})) \,|\, S_t = s \right] \\ &=: U_t^\pi(s) \end{aligned} \tag{7.7}$$

Q.E.D.

This lemma, however, does not suffice on its own, as agents may behave strategically by over-reporting their length, if the prices are not increasing. This would alter the transition probabilities, breaking the analysis. We will see that under a mild assumption, this can not happen, as the optimal policy for non-strategic agents will be monotone, and therefore truthful.

7.2.3 Monotonicity of the Optimal Pricing Policies

Recall that the solution of the more efficient MDP formulation is only correct if we can show that it is always monotone without considering the strategic behaviour of agents, ensuring incentive-compatibility of the optimum.

An optimal monotone strategy cannot be obtained for all the distributions on L, V, and D. As an example, for any distribution where a job's value is a deterministic function of their length, the optimal policy is to price-discriminate by length. If this function is not monotone, the optimum won't be either. To this end, we introduce the following assumption, which we will discuss below, and which will imply monotonicity of the pricing policy.

Assumption 7.1. The quantity $\frac{\mathbb{P}[V \geq \mu', D \geq s | L = \ell]}{\mathbb{P}[V \geq \mu, D \geq s | L = \ell]}$ is monotone non-decreasing as ℓ grows, for any state s and $0 \leq \mu < \mu'$ fixed.

This is not a natural, or immediately intuitive assumption. However, we will show that it is satisfied if the valuation of jobs follows a log-concave distribution which is parametrized by the job's length, and where the valuation is (informally) positively correlated with this length. Log-concave distributions are also commonly referred to as distributions possessing a *monotone hazard rate*, and it is common practice in economic settings to require this property of the agent valuations.

Lemma 7.2. *Let, V_ℓ^s denote the marginal r.v. V conditioned on $L = \ell$ and $D \geq s$. Let Z be a continuously-supported random variable, and $\gamma_1^s \leq \gamma_2^s \leq \cdots \in \mathbb{R}$. If V_ℓ^s is distributed like $\gamma_\ell^s \cdot Z$, $\lfloor \gamma_\ell^s \cdot Z \rfloor$, $Z + \gamma_\ell^s$, or $\lfloor Z + \gamma_\ell^s \rfloor$, then Assumption 7.1 is satisfied if Z is log-concave, or if the γ's are independent of ℓ.*

A discussion of log-concave random variables and a proof of this fact is given in Section 7.5. Many standard (discrete) distributions are (discrete) log-concave random variables, including the uniform, Gaussian, logistic, exponential, Poisson, binomial, etc. These can be proved to be log-concave from the discussion in Section 7.5. In the above, the γ terms represent a notion of spread or shifting, parametrized by the length, indicating some amount of positive correlation.

It remains to show price monotonicity under the above assumption. First, we begin with the following, which holds for arbitrary distributions. This ensures that over-selling time on the server can only hurt the mechanism.

Lemma 7.3. *Let $U_t^*(s)$ be the expected future revenue earned starting at time t in state s, for the optimal policy computed by Algorithm 7.2. Then the function $s \mapsto U_t^*(s)$ is monotone non-increasing in s for any t fixed.*

Proof. The proof is by induction on the time, decreasing. At time $t = T$, there is no future revenue and $U_T^*(s) = 0$, so the inductive claim follows trivially. Suppose, now, that the inductive claim holds at time $t + 1$. It suffices to show that this holds for each $u_t^*(s, \ell)$, since $U_t^*(s)$ is simply their expectation. Let π_t^* be the optimal pricing policy computed for the time t by the Algorithm 7.2. Since the function $\mu \mapsto \mathbb{P}[V \geq \mu \text{ and } \mathcal{E}]$, for any event $\mathcal{E}$, is left-continuous in the variable μ, we may define, for every $\ell \in \mathcal{L}$ and $s \in \mathcal{S}$,

$$\mu_s' := \max\left\{\mu : \mathbb{P}[V \geq \mu, D \geq s | L = \ell] \geq \mathbb{P}[V \geq \pi_t^*(s+1, \ell), D \geq s+1 | L = \ell]\right\} \tag{7.8}$$

We must have $\mu' \geq \pi_t^*(s+1, \ell)$, as $\mu = \pi_t^*(s+1, \ell)$ is in the set. Now, letting $\boldsymbol{p} := \mathbb{P}[V \geq$

$\pi^*_t(s+1,\ell), D \geq s+1 | L = \ell]$, we have

$$
\begin{aligned}
&u^*_t(s+1,\ell) && (7.9)\\
&= \boldsymbol{p}\cdot\pi^*_t(s+1,\ell) + \boldsymbol{p}\cdot U^*_{t+1}(s+\ell) + (1-\boldsymbol{p})U^*_{t+1}(s)\\
&\leq \boldsymbol{p}\cdot\pi^*_t(s+1,\ell) + \boldsymbol{p}\cdot U^*_{t+1}(s+\ell-1) + (1-\boldsymbol{p})U^*_{t+1}(s-1) && \text{(by induction)}\\
&\leq \boldsymbol{p}\cdot\Big(\pi^*_t(s+1,\ell) + U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)\Big)_+ + U^*_{t+1}(s-1)\\
&\leq \mathbb{P}[V \geq \mu'_s, D \geq s]\cdot\Big(\mu'_s + U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)\Big)_+ + U^*_{t+1}(s-1)\\
&\leq u^*_t(s,\ell) && \text{(subopt. price)},
\end{aligned}
$$

where $(x)_+ := \max\{x, 0\}$. The first inequality holds by the induction hypothesis, the second is by definition of $(\cdot)_+$, the third by the definition of μ'_s, and in the last, from the fact that μ'_s is a (possibly) suboptimal pricing policy for the state s at time t. Note that this last inequality requires that the 0 value be feasible in the max, which it is, by setting μ' arbitrarily large. *Q.E.D.*

We can now conclude

Lemma 7.4. If the distribution on job parameters satisfies the above assumption, then for all ℓ, s, t, we have $\pi^*_t(s,\ell) \leq \pi^*_t(s,\ell+1)$.

Proof. The idea is to show that, for any price μ less than the optimum $\pi^*_t(s,\ell)$, the difference in revenue between charging μ and $\pi^*_t(s,\ell)$ to jobs of length ℓ is less than the difference in revenue between the same prices for jobs of length $\ell+1$. This is achieved by applying the assumption to recursive definition of future revenue, along with the previous lemma.

Let $\boldsymbol{p}^\ell_s(\mu) := \mathbb{P}[V \geq \mu, D \geq s | L = \ell]$, fix s, t, and ℓ, and let μ_0 be equal to the optimal price $\pi^*_t(s,\ell)$. Observe that μ_0 maximizes the expression

$$\boldsymbol{p}^\ell_s(\mu)\big(\mu + U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)\big) + U^*_{t+1}(s-1) \tag{7.10}$$

For simplicity, let $\Delta_\ell := U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)$, and so for any $\mu \neq \mu_0$,

$$
\begin{aligned}
0 &\leq \boldsymbol{p}^\ell_s(\mu_0)\big(\mu_0 + \Delta_\ell\big) - \boldsymbol{p}^\ell_s(\mu)\big(\mu + \Delta_\ell\big)\\
&= \Big(\boldsymbol{p}^\ell_s(\mu_0) - \boldsymbol{p}^\ell_s(\mu)\Big)\big(\mu_0 + \Delta_\ell\big) + \boldsymbol{p}^\ell_s(\mu)\big(\mu_0 - \mu\big) \qquad (7.11)
\end{aligned}
$$

Note that, as discussed in the proof of the previous lemma, $\mu_0 + \Delta_\ell \geq 0$, as otherwise it would be beneficial to set $\pi_t^*(s,\ell) \leftarrow \infty$. The above inequality is then equivalent to

$$\frac{\boldsymbol{p}_s^\ell(\mu) - \boldsymbol{p}_s^\ell(\mu_0)}{\boldsymbol{p}_s^\ell(\mu)} \leq \frac{\mu_0 - \mu}{\mu_0 + \Delta_\ell} \quad \iff \quad \frac{\boldsymbol{p}_s^\ell(\mu_0)}{\boldsymbol{p}_s^\ell(\mu)} \geq 1 - \frac{\mu_0 - \mu}{\mu_0 + \Delta_\ell} \tag{7.12}$$

We wish to show that, if $\mu \leq \mu_0$, then as ℓ increases, the above inequality still holds. This would imply that the price $\mu_0 =: \pi_9^*(s,\ell)$ gives better return than μ for jobs of length $\ell + 1$, implying that the optimal price must be at least $\pi_t^*(s,\ell)$, which is our desired goal.

Now, by assumption 7.1, the left-hand-side is non-decreasing in ℓ, so it remains to show that the right-hand-side is non-increasing in ℓ. The only changing term is Δ_ℓ, which by Lemma 7.3, is non-increasing in ℓ. Since it is in the denominator of a subtracted, non-negative term, we have our desired result. *Q.E.D.*

With Lemma 7.4 and the results of Section 7.3, we finally have:

Theorem 7.2. The online server pricing problem admits an optimal monotone pricing strategy when the variables L, V, and D satisfy Assumption 7.1. Also,

1. In the finite horizon setting, when $\mathcal{V}$ is finitely supported, an exact optimum can be computed in time $O(T\mathbb{K}^3)$.

2. In the infinite horizon setting, when $\mathcal{V}$ is finitely supported, for all $\varepsilon > 0$, an ε-additive-approximate policy can be computed in time

$$O\left(\mathbb{K}^3 \log_\gamma\left(\tfrac{\varepsilon(1-\gamma)}{\mathbb{V}}\right)\right) \leq O\left(\tfrac{\mathbb{K}^3}{1-\gamma} \ln\left(\tfrac{\mathbb{V}}{\varepsilon(1-\gamma)}\right)\right) \tag{7.13}$$

3. In the finite horizon setting, when V is continuously supported, for all $\eta > 0$, an ηT-additive-approximate policy can be computed in time $O(T\mathbb{K}^2\mathbb{V}/\eta)$.

7.3 CONCENTRATION BOUNDS, AND EXTENSIONS

We show in this section that the performance of optimal policies concentrates, and we show how to extend the results of Section 7.2 to the infinite horizon, and how to approximate the optimal policy when parameter distributions are continuous.

7.3.1 Concentration Bounds on Revenue for Online Scheduling

In this section, we show that the revenue of arbitrary policies concentrates around their mean. In particular it holds true for the optimal or approximately optimal strategies described above. This will also allow us to argue later that, if we have an estimate $\hat{Q}$ of Q, then execute Algorithm 7.2 given the distribution $\hat{Q}$, then the output policy will perform well with respect to Q, both in expectation, and with high probability. To show this concentration, we will consider the *Doob* or *exposure* martingale of the cumulative revenue function, introduced in Section 7.1.1. Define

$$R_i^\pi := \mathbb{E}\left[\mathsf{CmlRev}_T(\pi, \mathcal{X}) | \mathcal{X}_1, \ldots, \mathcal{X}_i\right] \tag{7.14}$$

where the $\mathcal{X}_i$'s are jobs in the sequence $\mathcal{X}$ and the expected value is taken with respect to $\mathcal{X}_{i+1}, \ldots \mathcal{X}_T$. Thus, R_0^π is the expected cumulative revenue, and R_T^π is the random cumulative revenue. To formally describe this martingale sequence, we will introduce some notation, and formalize some previous notation. Recall that $\mathcal{X}_1, \mathcal{X}_2, \ldots$ is a sequence of jobs sampled *i.i.d.* from an underlying distribution Q. Fix a pricing policy $\pi : [T] \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$. Note that the state at time t is a random variable depending on both the (deterministic) pricing policy and the (random) $\mathcal{X}_1, \ldots, \mathcal{X}_{t-1}$. We denote it $S_t(\pi, \mathcal{X})$, or S_t for short. Formally, suppose $\mathcal{X}_t = (V_t, L_t, D_t)$, then $S_{t+1}(\pi, \mathcal{X}) = S_t(\pi, \mathcal{X}) - 1$ if either $V_t < \pi_t(S_t, L_t)$ or $D_t < S_t$, and otherwise $S_{t+1}(\pi, \mathcal{X}) = S_t(\pi, \mathcal{X}) + L_t - 1$. Furthermore, let $\mathsf{Rev}_t(\pi, \mathcal{X})$ be equal to 0 in the first case above (the t-th job is not scheduled), and $\pi_t(S_t, L_t)$ otherwise. Thus, $S_t(\pi, \mathcal{X})$ and $\mathsf{Rev}_t(\pi, \mathcal{X})$ are functions of the random values $\mathcal{X}_1, \ldots, X_t$ for π fixed. Note that Rev_t implicitly depends on S_t. Let $\mathcal{X}_{>i} := (\mathcal{X}_{i+1}, \mathcal{X}_{i+2}, \ldots)$ and $\mathcal{X}_{\le i} := (\mathcal{X}_1, \ldots \mathcal{X}_i)$. Recalling that $\mathsf{CmlRev}_T(\mathcal{X}, \pi) = \sum_{t=1}^T \mathsf{Rev}_t(\mathcal{X}, \pi)$, we have

$$\begin{aligned} R_i^\pi &= \sum_{t=0}^{i} \mathsf{Rev}_t(\pi, \mathcal{X}) + \mathbb{E}_{\mathcal{X}_{>i}}\left[\textstyle\sum_{t=i+1}^T \mathsf{Rev}_t(\pi, \mathcal{X}) \;\middle|\; S_{i+1}(\pi, \mathcal{X}_{\le i})\right] && \text{(7.15a)} \\ &= \left(\textstyle\sum_{t=0}^i \mathsf{Rev}_t(\pi, \mathcal{X}_{\le t})\right) + U_{i+1}^\pi(S_{i+1}(\pi, \mathcal{X}_{\le i})) && \text{(7.15b)} \end{aligned}$$

We wish to show that $\mathsf{CmlRev}(\mathcal{X}, \pi)$ concentrates around its mean. Since R_0^π is the expected revenue due to π, and R_T^π is the (random) revenue observed, it suffices to show $|R_0^\pi - R_T^\pi|$ is small, which we will do by applying Azuma's inequality, after showing the bounded-differences property. This gives the following.

Theorem 7.3. Let $\mathcal{X}$ be a finite sequence of T jobs sampled *i.i.d.* from Q, and let π be any monotone policy. Then, with probability $1 - \delta$,

$$|\mathsf{CmlRev}_T(\mathcal{X}, \pi) - \mathbb{E}_{\mathcal{X}'}\left[\mathsf{CmlRev}_T(\mathcal{X}', \pi)\right]| \le \mathbb{V} \cdot \sqrt{2 \log(2/\delta) T}. \tag{7.16}$$

in the finite horizon, and in the infinite-horizon-discounted,

$$|\mathsf{CmlRev}_\infty(\mathcal{X},\pi) - \mathbb{E}_{\mathcal{X}'}\left[\mathsf{CmlRev}_T(\mathcal{X}',\pi)\right]| \leq \mathbb{V}\cdot\sqrt{2\log(2/\delta)/(1-\gamma^2)}. \tag{7.17}$$

In particular these results hold true for the (approximately) optimal pricing strategies of Theorem 7.2.

Proof. For the finite horizon, we apply Azuma's inequality to the martingale R^π_t. We being by showing the bounded-differences property. Note that we do not require truthful behaviour from the jobs, since taking strategic behaviour into account for a non-monotone policy is equivalent to modifying the distribution over the jobs, and making the distribution state-dependent, by increasing the length of those jobs who would rather buy a longer interval. Thus,

$$\begin{aligned}
&\left|R^\pi_{t+1} - R^\pi_t\right| && (7.18)\\
&= \left| \textstyle\sum_{\tau=0}^{t+1} \mathsf{Rev}_\tau(\pi,\mathcal{X}) + \mathbb{E}_{\mathcal{X}_{>t+1}}\left[\sum_{\tau=t+2}^{T}\mathsf{Rev}_\tau(\pi,\mathcal{X}) \,\middle|\, S_{t+2}(\pi,\mathcal{X}_{\leq t+1})\right]\right. && (7.19)\\
&\qquad \left. - \textstyle\sum_{\tau=0}^{t}\mathsf{Rev}_\tau(\pi,\mathcal{X}) - \mathbb{E}_{\mathcal{X}_{>t}}\left[\sum_{\tau=t+1}^{T}\mathsf{Rev}_\tau(\pi,\mathcal{X}) \,\middle|\, S_{t+1}(\pi,\mathcal{X}_{\leq t})\right]\right| && (7.20)\\
&= \left|\mathsf{Rev}_{t+1}(\pi,\mathcal{X}) - \mathbb{E}_{\mathcal{X}_{t+1}}[\mathsf{Rev}_{t+1}(\pi,\mathcal{X})|S_{t+1}(\pi,\mathcal{X}_{\leq t})]\right| \leq \mathbb{V}
\end{aligned}$$

where the last inequality follows from properties of conditional expectation. With this property, we can apply Azuma's, and get

$$|\mathsf{CmlRev}_T(\mathcal{X},\pi) - \mathbb{E}_{\mathcal{X}'}\left[\mathsf{CmlRev}_T(\mathcal{X}',\pi)\right]| \leq \sqrt{2\log(2/\delta)(T+1)\mathbb{V}^2}\ . \tag{7.21}$$

For the infinite-horizon-discounted, we observe that equation (7.15) becomes

$$R^\pi_i = \sum_{t=0}^{i}\gamma^t\mathsf{Rev}_t(\pi,\mathcal{X}) + \mathbb{E}_{\mathcal{X}_{>i}}\left[\textstyle\sum_{t=i+1}^{T}\gamma^t\mathsf{Rev}_t(\pi,\mathcal{X}) \,\middle|\, S_{i+1}(\pi,\mathcal{X}_{\leq i})\right] \tag{7.22}$$

and thus we get that $|R^\pi_t - R^\pi_{t-1}| \leq \gamma^t\mathbb{V}$. Therefore with probability $1-\delta$,

$$|R^\pi_T - R^\pi_0| \leq \sqrt{2\log(2/\delta)\textstyle\sum_{t=0}^{T}(\gamma^t\mathbb{V})^2}\ =\ \mathbb{V}\cdot\sqrt{2\log(2/\delta)\textstyle\sum_{t=0}^{T}(\gamma^2)^t} \tag{7.23}$$

Thus, taking the limit as $T\to\infty$, we get that with probability $1-\delta$,

$$|\mathsf{CmlRev}_\infty(\mathcal{X},\pi) - \mathbb{E}_{\mathcal{X}'}\left[\mathsf{CmlRev}_T(\mathcal{X}',\pi)\right]| \leq \mathbb{V}\cdot\sqrt{2\log(2/\delta)/(1-\gamma^2)}. \tag{7.24}$$

Q.E.D.

7.3.2 Infinite Discounted Horizon

In this section, we extend the finite-horizon results to compute the optimal policies in the infinite-horizon-discounted setting. Recall, in this infinite horizon discounted setting, we seek to maximize the γ-discounted future revenue,

$$\mathsf{CmlRev}_{\infty}(\mathcal{X}, \pi) := \sum_{t=0}^{\infty} \gamma^t \mathsf{Rev}_t(\mathcal{X}, \pi) \tag{7.25}$$

over the choice of $\pi : \mathbb{N} \times \mathcal{S} \times \mathcal{L} \to \mathbb{R}$. Algorithm 7.2 does not allow us to immediately compute a solution for the infinite discounted horizon case. However we can exploit the discounting factor on the revenues to obtain an approximation of the infinite optimum: it suffices to consider the truncated problem up to a certain sufficiently large T and solve it optimally using the algorithm presented above. In fact we have the following Lemma.

Lemma 7.5. For any $\varepsilon > 0$ and $T \geq \log_{\gamma} (\varepsilon(1-\gamma)/\mathbb{V})$, let π be the pricing policy computed by the finite-horizon algorithm up to time T. Let $\bar{\pi}$ be the time-independent pricing policy such that $\bar{\pi}(\cdot, \cdot) := \pi_0(\cdot, \cdot)$. Then the expected performance of the optimal policy in the infinite horizon is within an additive ε of expected performance of $\bar{\pi}$.

Proof. Note that in order to compute policy π it is necessary to add the discount factor to Algorithm 7.2, and to all of the proofs of previous sections. One can verify that all proofs go through. Let π^* be the Bayes-optimal infinite-horizon strategy — which is known to be time-independent — and let π be as in the statement (where we set $\pi_t(s, \ell) = \infty$ for all $t > T$.) Then, in expectation over times 0 through T, pricing as π yields greater revenue than following π^*. Conversely, in expectation over all time, pricing as π^* yields greater revenue than π. However, after time T, the maximum possible revenue due to any policy is

$$\textstyle\sum_{t=T}^{\infty} \gamma^t \cdot \mathbb{V} = \gamma^T \cdot \mathbb{V} \cdot \left(\frac{1}{1-\gamma}\right) \ \leq \ \varepsilon \tag{7.26}$$

And so the difference in revenue due to following π or π^* is at most ε, since T is sufficiently large.

It remains to show that $\bar{\pi}$ performs better than π overall. Let π^i be the policy which agrees with π_0 for all $t \leq i$, then equals π_{t-i} for $t > i$. Observe that, π^1 is optimal in expectation over the interval $[1, T+1]$, and is equivalent to $\pi = \pi^0$ for the first step. Therefore, π^1 performs better than π. Similarly, we can argue π^{i+1} performs better than π^i over the interval $[i, T+i]$ and equally before, hence performs better overall.

Thus, we have a sequence of policies $\pi = \pi^0, \pi^1, \pi^2, \ldots$ converging to $\bar{\pi}$, and whose expected revenue is monotone non-decreasing along the sequence. Therefore, the expected revenue due to $\bar{\pi}$ is greater than that of π, which is an ε additive-approximation to the optimal policy. *Q.E.D.*

The approach above is analogous to the classical value iteration technique [168].

(Lemma 7.7, p. 145) *Let Q, and $\hat{Q}$ such that $|Q - \hat{Q}| < \varepsilon$. In the infinite horizon, $|U^*(s) - \hat{U}^*(s)| < 2\mathbb{V}\mathbb{L}\varepsilon/(1-\gamma)$ for all s.*

Proof. As in the proof of Lemma 7.5, if T is sufficiently large, we may analyze the first T time steps as a finite-horizon problem, and the remaining revenue will be negligibly small. Now, the calculation above can be reproduced with discount terms, to show

$$\left|U_t^*(s) - \hat{U}_t^*(s)\right| \leq \sup_{\sigma'} \left|\gamma U_{t+1}^*(\sigma') - \gamma \hat{U}_{t+1}^*(\sigma')\right| + \textstyle\sum_{\ell \in \mathcal{L}} 2\underline{\varepsilon}\mathbb{V} \qquad (7.27)$$

Then, inductively applying this and taking $T \to \infty$, we have $|U_0^*(s) - \hat{U}_0^*(s)| \leq 2\mathbb{V}\mathbb{L}\underline{\varepsilon}/(1-\gamma)$. *Q.E.D.*

These results are used to prove the infinite-horizon versions of the various results throughout the paper, specifically the Theorems 7.2–7.3, and 7.5.

7.3.3 Approximation Algorithm for Continuously Supported Values

In this section, we argue that the optimal policy may be computed within some error when the distribution over values is continuous, rather than discrete. Note that the algorithms above assume that the *value* of the jobs (V) is discretely supported, and the running time depends on $|\mathcal{V}|$. In this section, we analyze the error incurred by discretizing the space of possible values, and then computing the optimal policy.

Let $\eta > 0$ be some desired small grid size, and suppose we only allow ourselves to set prices which are multiples of η. We claim that this incurs a small loss on the total revenue.

Define, as in the previous subsections, $\boldsymbol{p}_s^\ell(\mu) := \mathbb{P}[V \geq \mu, D \geq s | L = \ell]$. Further, define as previously $U_t^*(s) = \mathbb{E}_L\left[u_t^*(s, L)\right]$, and

$$u_t^*(s, \ell) := \max_{\mu \in \mathbb{R}} \left[\boldsymbol{p}_s^\ell(\mu)\big(\mu + U_{t+1}^*(s + \ell - 1)\big) + (1 - \boldsymbol{p}_s^\ell(\mu))U_{t+1}^*(s-1)\right] \qquad (7.28)$$

Define $U_{t,\eta}^*(s)$ and $u_{t,\eta}^*(s,\ell)$ similarly, restricting the maximum to choosing μ from multiples of η.

Lemma 7.6. Let $U_t^*(\cdot)$ and $U_{t,\eta}^*(\cdot)$ be defined as above, then $|U_t^*(s) - U_{t,\eta}^*(s)| \leq (T-t)\eta \; \forall s, t$.

Proof. We will show this by induction on the value of t, decreasing. Assume that $|U^*_t(s) - U^*_{t,\eta}(s)| < \Delta_t$ for all t, s, and set $\Delta_{T+1} = 0$. We wish to inductively bound the value of Δ_t. Now,

$$\begin{aligned} u^*_{t,\eta}(s,\ell) &= \max_{\mu\in\eta\cdot\mathbb{Z}} \left[\boldsymbol{p}^\ell_s(\mu)\big(\mu + U^*_{t+1,\eta}(s+\ell-1)\big) + (1-\boldsymbol{p}^\ell_s(\mu))U^*_{t+1,\eta}(s-1)\right] \\ &\geq \max_{\mu\in\eta\cdot\mathbb{Z}} \Big[\boldsymbol{p}^\ell_s(\mu)\big(\mu + U^*_{t+1}(s+\ell-1) - \Delta_{t+1}\big) + \\ &\qquad\qquad\qquad + (1-\boldsymbol{p}^\ell_s(\mu))U^*_{t+1}(s-1) - \Delta_{t+1}\Big] \\ &= -\Delta_{t+1} + \max_{\mu\in\eta\cdot\mathbb{Z}} \left[\boldsymbol{p}^\ell_s(\mu)\big(\mu + U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)\big) + U^*_{t+1}(s-1)\right] \quad (7.29) \end{aligned}$$

Now, let μ^* be the optimizer of this right hand side over $\mathbb{R}$ (where the value would attain $u^*_t(s,\ell)$), and $\hat{\mu}$ be μ^* rounded *down* to the nearest multiple of η. Then, since $\boldsymbol{p}^\ell_s(\cdot)$ is non-increasing,

$$\begin{aligned} &\boldsymbol{p}^\ell_s(\hat{\mu})\big(\hat{\mu} + U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)\big) + U^*_{t+1}(s-1) \\ &\quad\geq \boldsymbol{p}^\ell_s(\mu^*)\big(\mu^* - \eta + U^*_{t+1}(s+\ell-1) - U^*_{t+1}(s-1)\big) + U^*_{t+1}(s-1) \\ &\quad= u^*_t(s,\ell) - \eta\cdot\boldsymbol{p}^\ell_s(\mu^*) \end{aligned} \quad (7.30)$$

Thus combining both equations, we get

$$u^*_{t,\eta}(s,\ell) \;\leq\; u^*_t(s,\ell) \;\leq\; u^*_{t,\eta}(s,\ell) + \eta + \Delta_{t+1} \quad (7.31)$$

From which we conclude, by averaging over ℓ, that $\Delta_t \leq (T-t)\eta$, as desired. *Q.E.D.*

Corollary 7.1. *Let $U^*(\cdot)$ and $U^*_{\infty,\eta}(\cdot)$ be defined as above, but for the infinite horizon discounted, then $|U^*(s) - U^*_{\infty,\eta}(s)| \leq \eta/(1-\gamma)$ $\forall s$.*

Proof. As shown in the previous subsection, it suffices to perform the analysis in the finite horizon, while taking the discount factor into account, then take the limit as $T \to \infty$. The same calculations as above gives

$$\begin{aligned} u^*_{t,\eta}(s,\ell) &\geq -\Delta_{t+1} + \max_{\mu\in\eta\mathbb{Z}} \left[\boldsymbol{p}^\ell_s(\mu)\left(\mu + \gamma U^*_{t+1}(s+\ell-1) - \gamma U^*_{t+1}(s-1)\right) + \gamma U^*_{t+1}(s-1)\right] \\ &\geq u^*_{t+1}(s,\ell) - \eta - \gamma\Delta_{t+1} \end{aligned} \quad (7.32)$$

Summing the Δ's and taking $T \to \infty$, we get $u^*_{\infty,\eta}(s,\ell) \geq u^*(s,\ell) - \eta/(1-\gamma)$ as desired. *Q.E.D.*

7.4 ROBUSTNESS OF PRICING WITH APPROXIMATE DISTRIBUTIONAL KNOWLEDGE

In this section, we show that results analogous to Theorems 7.2 and 7.3 may be obtained even in the case in which we do not have full knowledge of the distribution Q, but only an estimate $\hat{Q}$. We then show how to obtain a valid $\hat{Q}$ from samples.

7.4.1 Robustness of the pricing strategy

Let's suppose that instead of knowing the exact distribution $Q = (D, L, V)$ of the jobs, we have only access to some estimate $\hat{Q} = (\hat{D}, \hat{L}, \hat{V})$ with the following property, for some $\varepsilon > 0$:

$$\left|\mathbb{P}(\hat{L} = \ell, \hat{V} \geq v, \hat{D} \geq s) - \mathbb{P}(L = \ell, V \geq v, D \geq s)\right| < \varepsilon \quad \forall s \in S, \ell \in \mathcal{L} \text{ and } v \in \mathcal{V}. \tag{7.33}$$

For the sake of brevity, we abuse notation and denote the condition in (7.33) as $|Q - \hat{Q}| < \varepsilon$. Later, we will need to estimate the value $\mathbb{P}[L = \ell, \neg(V \geq v, D \geq s)]$, given $\hat{Q}$, that is the probability that the job has length ℓ, but either cannot afford price v, or cannot be scheduled s slots in the future. This is equal to $\mathbb{P}[L = \ell] - \mathbb{P}[L = \ell, V \geq v, D \geq s]$.

The left-hand term is equal to $\mathbb{P}[L = \ell, V \geq 0, D \geq 0]$, and so we have access to both terms. The estimation error is additive, so the deviation is at most 2ε.
Denote $\boldsymbol{p}^{\ell}_{t,s} := \mathbb{P}[V \geq \pi^t(s, \ell), D \geq s | L = \ell]$, and recall

$$U^{\pi}_t(s) := \sum_{\ell \in \mathcal{L}} \mathbb{P}[L = \ell]\Big(\boldsymbol{p}^{\ell}_{t,s}\big(\pi_t(s, \ell) + U^{\pi}_{t+1}(s + \ell - 1)\big) + (1 - \boldsymbol{p}^{\ell}_{t,s})U^{\pi}_{t+1}(s - 1)\Big), \tag{7.34}$$

the expected revenue from time t onwards, conditioning on $S_t = s$. Let $\hat{U}^{\pi}_t(\cdot)$ be the same as $U^{\pi}_t(\cdot)$, but where the variables are distributed as $\hat{Q}$. As before, let $U^*_t(\cdot)$ be $U^{\pi}_t(\cdot)$ for $\pi = \pi^*$, the Bayes-optimal policy returned by Algorithm 7.2, and $\hat{U}^*_t(\cdot)$ defined similarly but with respect to $\hat{Q}$. We will show that $\hat{U}^*_t(\cdot)$ is a good estimate for $U^*_t(\cdot)$.

Lemma 7.7. Let Q, and $\hat{Q}$ such that $|Q - \hat{Q}| < \varepsilon$.

1. In the finite horizon, $|U^*_t(s) - \hat{U}^*_t(s)| < 2(T - t)\mathbb{V}\mathbb{L}\varepsilon$ for all t, s;
2. In the infinite horizon, $|U^*(s) - \hat{U}^*(s)| < 2\mathbb{V}\mathbb{L}\varepsilon/(1 - \gamma)$ for all s, where U^* is the optimal time independent strategy.

We present here the proof for the finite horizon setting. The result in the infinite horizon is then concluded from Section 7.3.2.

Proof, finite horizon. Let π^* be the policy computed by Algorithm 7.2 with access to Q. As in Section 7.2, we denote $\boldsymbol{p}^\ell_{t,s} := \mathbb{P}[V \geq \pi^*_t(s,\ell), D \geq s | L = \ell]$, and $\boldsymbol{P}(\ell) := \mathbb{P}_{\mathcal{X}}[L = \ell]$. In an abuse of notation, denote $\hat{\boldsymbol{p}}^\ell_{t,s}$ and $\hat{\boldsymbol{P}}(\ell)$ the estimated values of $\boldsymbol{p}^\ell_{t,s}$ and $\boldsymbol{P}(\ell)$, respectively. We cannot estimate $\boldsymbol{p}^\ell_{t,s}$ directly with good error bounds, but we will only need the values $\hat{\boldsymbol{P}}(\ell)\hat{\boldsymbol{p}}^\ell_{t,s}$ and $\hat{\boldsymbol{P}}(\ell)(1 - \hat{\boldsymbol{p}}^\ell_{t,s})$. Now, substituting these estimates into (7.34), we get:

$$\begin{aligned}
&\left| U^*_t(s) - \hat{U}^*_t(s) \right| \\
&\quad = \Bigg| \sum_{\ell \in \mathcal{L}} \boldsymbol{P}(\ell) \Big(\boldsymbol{p}^\ell_{t,s} \pi^*_t(s,\ell) + \boldsymbol{p}^\ell_{t,s} U^*_{t+1}(s + \ell - 1) + (1 - \boldsymbol{p}^\ell_{t,s}) U^*_{t+1}(s-1) \Big) \\
&\qquad - \sum_{\ell \in \mathcal{L}} \hat{\boldsymbol{P}}(\ell) \Big(\hat{\boldsymbol{p}}^\ell_{t,s} \pi^*_t(s,\ell) + \hat{\boldsymbol{p}}^\ell_{t,s} \hat{U}^*_{t+1}(s + \ell - 1) + (1 - \hat{\boldsymbol{p}}^\ell_{t,s}) \hat{U}^*_{t+1}(s-1) \Big) \Bigg| \qquad (7.35)
\end{aligned}$$

To simplify this expression, we begin by showing a simple claim: Let x, y, $\hat{x}$, $\hat{y} \in \mathbb{R}$, and let $\lambda, \hat{\lambda} \in [0,1]$, such that $|x - \hat{x}| < \delta$, $|y - \hat{y}| < \delta$, and $|\lambda - \hat{\lambda}| < \varepsilon$. Then

$$\begin{aligned}
\big| \big(\lambda x + &(1-\lambda) y\big) - \big(\hat{\lambda}\hat{x} + (1-\hat{\lambda})\hat{y}\big) \big| \\
&\leq \big| \big(\lambda x + (1-\lambda) y\big) - \big(\hat{\lambda} x + (1-\hat{\lambda}) y\big) \big| + \\
&\qquad \big| \big(\hat{\lambda} x + (1-\hat{\lambda}) y\big) - \big(\hat{\lambda}\hat{x} + (1-\hat{\lambda})\hat{y}\big) \big| \\
&\leq |\lambda - \hat{\lambda}| \cdot |x - y| + \hat{\lambda}|x - \hat{x}| + (1 - \hat{\lambda})|y - \hat{y}| \\
&\leq \varepsilon |x - y| + \delta \qquad (7.36)
\end{aligned}$$

Now, replacing x and y with $\big(\pi^*_t(s,\ell) + U^*_{t+1}(s + \ell - 1)\big)$ and $U^*_{t+1}(s-1)$, respectively, and replacing λ with $\boldsymbol{P}(\ell)\boldsymbol{p}^\ell_{t,s}$, we have

$$\begin{aligned}
\left| U^*_t(s) - \hat{U}^*_t(s) \right| \leq \sum_{\ell \in \mathcal{L}} \Bigg(& 2\underline{\varepsilon} \cdot \sup_{\sigma} \left| \pi^*_t(\sigma,\ell) + U^*_{t+1}(\sigma + \ell - 1) - U^*_{t+1}(\sigma - 1) \right| + \\
& + \hat{\boldsymbol{P}}(\ell) \cdot \sup_{\sigma'} \left| U^*_{t+1}(\sigma') - \hat{U}^*_{t+1}(\sigma') \right| \Bigg) \qquad (7.37)
\end{aligned}$$

However, the argument of the supremum in left-hand term in the summand must be at most $\mathbb{V}$, since if $U^*_{t+1}(\sigma + \ell - 1) \leq U^*_{t+1}(s-1)$, it is best to $\pi^*_t(\sigma) = \infty$, which makes $\boldsymbol{p}^\ell_{t,s} = 0$, putting all the weight on $U^*_{t+1}(s-1)$. Furthermore, we have shown in Lemma 7.3 that $U^*_{t+1}(s + \ell - 1) \leq U^*_{t+1}(s-1)$. Thus, we get

$$\left| U^*_t(s) - \hat{U}^*_t(s) \right| \leq \sup\nolimits_{\sigma'} \left| U^*_{t+1}(\sigma') - \hat{U}^*_{t+1}(\sigma') \right| + \textstyle\sum_{\ell \in \mathcal{L}} 2\underline{\varepsilon}\mathbb{V} \qquad (7.38)$$

Inductively applying this gives $\left|U_t^*(s) - \hat{U}_t^*(s)\right| \leq 2(T-t)\mathbb{L}\mathbb{V}\underline{\varepsilon}$ as desired. *Q.E.D.*

7.4.2 Learning the Underlying Distribution from Samples

As discussed above, we show here how to compute a $\hat{Q}$ from samples of Q, such that $|Q - \hat{Q}|$ is small with high probability. In particular we present a sampling procedure which respects the rules of the pricing server mechanism. When a job arrives, we only learn its length, and only if it agrees to be scheduled. Thus, we are not given full samples of Q, complicating the learning procedure. Thanks to the previous section, we know that a policy which is optimal with respect to $\hat{Q}$ will be close-to-optimal with respect to Q.
We remark, however, that the power of the results of the previous section is not exhausted by this application: one may apply directly the robustness results to specific problems in which the $\hat{Q}$ is subject to (small) noise or an approximate distribution is already known from other sources.
Let $\mathcal{X} = \{(L_1, V_1, D_1),\ \ldots,\ (L_n, V_n, D_n),\}$ be an *i.i.d.* sample of n jobs from the underlying distribution Q. Note that the expectation of an indicator is the probability of the indicated event. Fix a length ℓ, a state s, and a value v. As a consequence of Höffding's inequality, with probability $1-\delta$,

$$\left|\tfrac{1}{n}\textstyle\sum_{k=1}^{n}\mathbb{I}[L_k = \ell, V_k \geq v, D_k \geq s] - \mathbb{P}[L = \ell, V \geq v, D \geq s]\right| \leq \sqrt{\tfrac{\log(2/\delta)}{2n}} \tag{7.39}$$

Sampling Procedure. We wish to estimate the value $\mathbb{P}[L = \ell, V \geq v, D \geq s]$ for all choices of ℓ, v, and s. Fixing v and s, we may repeatedly post prices $\pi_t(s,\ell) = v$ and declare that the earliest available time is s, then record (i) which job accepts to be scheduled, and (ii) the length of each scheduled job. Let $\underline{\varepsilon} > 0$ and $n \geq \log(2/\delta)/(2\underline{\varepsilon}^2)$, then by (7.39), the sample-average of each value will have error at most $\underline{\varepsilon}$ with probability $1-\delta$, for any one choice of (ℓ, v, s).

Repeating this process for all $\leq \mathbb{K}^2$ choices of $v \in \mathcal{V}$ and $s \in \mathcal{S}$ gives us estimates for each. Now, if we want to have the estimate hold over all choices of ℓ, v, s, it suffices to take the union bound over all $\leq \mathbb{K}^3$ values (incl. ℓ), and scaling accordingly. If we take $n \geq 3\log(2\mathbb{K}/\delta)/(2\underline{\varepsilon}^2)$ samples for each of the $\leq \mathbb{K}^2$ choices of v and s, then simultaneously for all ℓ, v, and s, the quantity in (7.39) is at most $\underline{\varepsilon}$. So we have obtained the "$|Q - \hat{Q}| < \underline{\varepsilon}$" condition. It should be noted that, for this sampling procedure, if a job of length ℓ is scheduled, we must possibly wait at most ℓ times units before taking the next sample to clear the buffer. This blows up the sampling *time* by a factor of $O(\mathbb{L})$. The following result follows immediately from Lemma 7.7 and Höffding's inequality for the right choice of n.

Lemma 7.8. Let n, Q, and $\hat{Q}$, be as above. In the finite horizon, for all $\varepsilon > 0$, if $n \geq 6T\mathbb{K}^4 \log(2\mathbb{K}/\delta)/\varepsilon^2$, we have that with probability $1-\delta$, $|U_t^*(s) - \hat{U}_t^*(s)| < \varepsilon$ for all t, s. In the infinite horizon, if $n \geq 6\mathbb{K}^4 \log(2\mathbb{K}/\delta)/((1-\gamma)\varepsilon^2)$, we have that with probability $1-\delta$, $|U^*(s) - \hat{U}^*(s)| < \varepsilon$ for all s.

7.4.3 Performance of the Computed Policy

We use here the result of the previous sections to analyze the performance of the policy output by Algorithm 7.2 after the learning procedure. By the estimation of revenue, the best policy in estimated-expectation is near-optimal in expectation. Since revenues from arbitrary policies concentrate, we get near-optimal revenue in hindsight.
Formally, for $\varepsilon > 0$, Lemma 7.8 gives us that if the sample-distribution $\hat{Q}$ is computed on $n \geq 6T\mathbb{K}^4 \log(2\mathbb{K}/\delta)/\varepsilon^2$ samples, then with probability $1-\delta$ over the samples, $|U_t^*(s) - \hat{U}_t^*(s)| \leq \varepsilon$. Note that $U_{t=0}^*(s=0)$ is exactly the expected cumulative revenue of the optimal policy. For clarity of notation, denote

$$\mathsf{ECRev}_T(\pi|Q) := \mathbb{E}_{\mathcal{X}\sim Q}\left[\mathsf{CmlRev}_T(\mathcal{X}, \pi)\right] \tag{7.40}$$

We have shown that for sufficient samples, $|\mathsf{ECRev}_T(\pi^*|Q) - \mathsf{ECRev}_T(\pi^*|\hat{Q})| < \varepsilon$, with probability $1-\delta$. This observation allows us to then conclude

Theorem 7.4 (Finite Horizon)**.** Let Q be the underlying distribution over jobs. Let $\varepsilon > 0$, and $n \geq 24T\mathbb{K}^4 \log(8\mathbb{K}/\delta)/\varepsilon^2$. Then in time $O(T\mathbb{K}^3 + n\mathbb{L})$, we may compute a policy $\hat{\pi}$ which is monotone in length, and therefore incentive compatible, such that for any policy π, with probability $(1-\delta)$,

$$\mathsf{CmlRev}_T(\mathcal{X}, \hat{\pi}) \geq \mathsf{CmlRev}_T(\mathcal{X}, \pi) - 2\mathbb{V}\sqrt{2\log({}^8\!/\!_\delta)(T+1)} - \varepsilon \tag{7.41}$$

Furthermore, if the distribution over values V is continuous rather than discrete, we may compute in time $O(T\mathbb{K}^2\mathbb{V}/\eta + n\mathbb{L})$ a monotone policy $\hat{\pi}$ such that for any policy π, with probability $1-\delta$,

$$\mathsf{CmlRev}_T(\mathcal{X}, \hat{\pi}) \geq \mathsf{CmlRev}_T(\mathcal{X}, \pi) - 2\mathbb{V}\sqrt{2\log({}^8\!/\!_\delta)(T+1)} - \varepsilon - \eta T \tag{7.42}$$

Proof. We have chosen $n \geq 6T\mathbb{K}^4 \log(2\mathbb{K}/(\delta/4))/(\varepsilon/2)^2$. Let π^* be the optimal policy for the true distribution Q. By Theorem 7.3, we have $|\mathsf{CmlRev}_T(\mathcal{X}, \pi) - \mathsf{ECRev}_T(\pi|Q)| < \mathbb{V}\sqrt{2\log(8/\delta)(T+1)}$ with probability $1-\delta/4$ for both π and $\hat{\pi}$. Furthermore, by Lemma 7.8,

$|\mathsf{ECRev}_T(\pi|Q) - \mathsf{ECRev}_T(\pi|\hat{Q})| < \varepsilon/2$ with probability $1-\delta/4$, for both $\pi = \hat{\pi}$ and π^*. This is because from the point of view of $\hat{\pi}$, $\hat{Q}$ is the true distribution, and Q is the estimate. Taking the union bound over all four events above, and recalling that $\hat{\pi}$ maximizes $\mathsf{ECRev}_T(\pi|\hat{Q})$, and π^* maximizes $\mathsf{ECRev}_T(\pi|Q)$, we get the following with probability $1-\delta$:

$$\begin{aligned}
\mathsf{CmlRev}_T(\mathcal{X}, \hat{\pi}) &\geq \mathsf{ECRev}_T(\hat{\pi}|Q) - \mathbb{V}\sqrt{2\log(8/\delta)(T+1)} && \text{(concentration)} \\
&\geq \mathsf{ECRev}_T(\hat{\pi}|\hat{Q}) - \mathbb{V}\sqrt{2\log(8/\delta)(T+1)} - \varepsilon/2 && \text{(sample error)} \\
&\geq \mathsf{ECRev}_T(\pi^*|\hat{Q}) - \mathbb{V}\sqrt{2\log(8/\delta)(T+1)} - \varepsilon/2 && \text{(optimality)} \\
&\geq \mathsf{ECRev}_T(\pi^*|Q) - \mathbb{V}\sqrt{2\log(8/\delta)(T+1)} - \varepsilon && \text{(sample error)} \\
&\geq \mathsf{ECRev}_T(\pi|Q) - \mathbb{V}\sqrt{2\log(8/\delta)(T+1)} - \varepsilon && \text{(optimality)} \\
&\geq \mathsf{CmlRev}_T(\mathcal{X}, \pi) - 2\mathbb{V}\sqrt{2\log(8/\delta)(T+1)} - \varepsilon && \text{(concentration)} \qquad (7.43)
\end{aligned}$$

as desired.

When V is continuously distributed, choose prices which are multiples of η between 0 and $\mathbb{V}$, as is outlined in Section 7.3.3. *Q.E.D.*

For what concerns the γ-discounted infinite horizon case, we have the following

Theorem 7.5 (Infinite Horizon, Discounted)**.** Let Q be the underlying distribution over jobs. Let $\varepsilon > 0$, and $n \geq 24\mathbb{K}^4 \frac{\log(8\mathbb{K}/\delta)}{\varepsilon^2(1-\gamma)}$. Then we may compute a policy $\hat{\pi}$ in time $O\left(\frac{\mathbb{K}^3}{1-\gamma} \ln\left(\frac{\mathbb{V}}{\varepsilon(1-\gamma)}\right) + n\mathbb{L}\right)$, which is monotone, and thus incentive compatible, such that for any policy π, with probability $(1-\delta)$,

$$\mathsf{CmlRev}_\infty(\mathcal{X}, \hat{\pi}) \geq \mathsf{CmlRev}_\infty(\mathcal{X}, \pi) - 2\mathbb{V}\sqrt{2\log(^8\!/_\delta)/(1-\gamma^2)} - 2\varepsilon \qquad (7.44)$$

Furthermore, if the distribution over values V is continuous rather than discrete, we may compute in time $O\left(\frac{\mathbb{K}^2\mathbb{V}/\eta}{1-\gamma} \ln\left(\frac{\mathbb{V}}{\varepsilon(1-\gamma)}\right) + n\mathbb{L}\right)$ a monotone policy $\hat{\pi}$ such that for any π, with probability $1-\delta$,

$$\mathsf{CmlRev}_\infty(\mathcal{X}, \hat{\pi}) \geq \mathsf{CmlRev}_\infty(\mathcal{X}, \pi) - 2\mathbb{V}\sqrt{2\log(^8\!/_\delta)/(1-\gamma^2)} - 2\varepsilon - \eta/(1-\gamma) \qquad (7.45)$$

As above, this policy $\hat{\pi}$ is computed by learning $\hat{Q}$ from n samples as in Section 7.4.2, then running the modified Algorithm 7.2 for the estimated distribution as in Section 7.3.2. In case V is continuously distributed, we restrict ourselves to prices which are multiples of η between 0 and $\mathbb{V}$. The details of the proof are in Section 7.3.

We recall that all these results need the distribution assumption from Section 7.2.3.

7.5 LOG-CONCAVE DISTRIBUTIONS

In Section 7.2.3, we sought to show that if the value of a random job has a log-concave distribution, then the optimal policy will be monotone. We present here a discussion of log-concavity, both for continuous and discrete random variables, and give the proof of the monotonicity of the prices.

Formally, a function $f : \mathbb{R} \to \mathbb{R}$ is log-concave if for any x and y, and for any $0 \le \theta \le 1$, $\lg f(\theta x + (1-\theta)y) \ge \theta \lg f(x) + (1-\theta) \lg f(y)$. Equivalently, $f(\theta x + (1-\theta)y) \ge f(x)^{\theta} f(y)^{1-\theta}$. For a discretely supported $f : \mathbb{Z} \to \mathbb{R}$, we replace this condition with $f(x)^2 \ge f(x-1)f(x+1)$, emulating the continuous definition with $\theta = \frac{1}{2}$. We further require that the support of f be "connected".

Definition 7.1. A continuous random variable X with density function f is said to be log-concave if f is log-concave. A discrete random variable Y with probability mass function p is said to be log-concave if p is discretely log-concave.

A well-known fact is that log-concave random variables also have log-concave cumulative density/mass functions. We present here a quick proof of this fact, for completeness.

Claim 7.6. *If X is a log-concave continuous r.v., then $\mathbb{P}[X \le x]$, and $\mathbb{P}[X \ge x]$ are log-concave functions of x. If Y is a log-concave discrete r.v. supported on $\mathbb{N}$, then $\mathbb{P}[Y \le y]$ and $\mathbb{P}[Y \ge y]$ are discretely log-concave functions of y.*

Proof. The continuous case is well-documented in the literature. See for example [169]. For the discrete case, observe first that since a mass function is non-negative, and we have assumed contiguous support, the function must be single-peaked, *i.e.* quasi-concave, as any local minimum would contradict the definition. Furthermore, the definition of log-concavity is equivalent to $\frac{p_y}{p_{y-1}} \ge \frac{p_{y+1}}{p_y}$. Repeatedly applying this, and rearranging, we get

$$p_y p_{y+k} \ge p_{y-1} p_{y+k+1} \quad \forall y, k \in \mathbb{Z},\, k \ge 0 \;. \tag{7.46}$$

It remains to show that $P(y) := \sum_{-\infty}^{y} p_k$ is log-concave. We have

$$\begin{aligned} P(y)P(y) &= P(y-1)P(y) + \sum_{-\infty}^{y} p_k p_y \\ &\ge P(y-1)P(y) + \sum_{-\infty}^{y} p_{k-1} p_{y+1} = P(y-1)P(y+1) \end{aligned} \tag{7.47}$$

as desired. The same technique applies for the upper-sum. *Q.E.D.*

This will allow us to then conclude:

(Lemma 7.2, p.137) *Let, V_ℓ^s denote the marginal r.v. V conditioned on $L = \ell$ and $D \geq s$. Let Z be a continuously-supported random variable, and $\gamma_1^s \leq \gamma_2^s \leq \cdots \in \mathbb{R}$. If V_ℓ^s is distributed like $\gamma_\ell^s \cdot Z$, $\lfloor \gamma_\ell^s \cdot Z \rfloor$, $Z + \gamma_\ell^s$, or $\lfloor Z + \gamma_\ell^s \rfloor$, then Assumption 7.1 is satisfied if Z is log-concave, or if the γ's are independent of ℓ.*

Proof. First, observe that

$$\mathbb{P}[V \geq \mu, D \geq s | L = \ell] = \mathbb{P}[V \geq \mu | D \geq s, L = \ell] \cdot \mathbb{P}[D \geq s | L = \ell] \ . \tag{7.48}$$

and since we are taking ratios for s fixed, we can replace the joint cumulatives on V and D in the assumption, with the marginals on just V.

Now, if the γ's are independent of ℓ, then the ratio remains unchanged as ℓ changes, satisfying Assumption 7.1. Otherwise, we begin by analyzing the distributions given by $\gamma_\ell^s Z$ and $Z + \gamma_\ell^s$. Let $\bar{F}(x) := \mathbb{P}[Z \geq x]$, noting that $\mathbb{P}[V_\ell^s \geq \mu] = \bar{F}(\mu/\gamma_\ell^s)$ and $\bar{F}(\mu - \gamma_\ell^s)$, for the two cases, respectively. Note that we wish to show $\mathbb{P}[V_\ell^s \geq \mu']/\mathbb{P}[V_\ell^s \geq \mu]$ is increasing, which is equivalent to $\log(\mathbb{P}[V_\ell^s \geq \mu']) - \log(\mathbb{P}[V_\ell^s \geq \mu])$ increasing.

For $V_\ell^s \sim Z + \gamma_\ell^s$, observe that for $x' > x$ and $\gamma' > \gamma$, we have

$$\log \bar{F}(x - \gamma) - \log \bar{F}(x' - \gamma) \geq \log \bar{F}(x - \gamma') - \log \bar{F}(x' - \gamma') \tag{7.49}$$

since $\log \bar{F}$ is a non-increasing and concave function, by assumption. Also

$$\begin{aligned} \log \bar{F}(x/\gamma) - \log \bar{F}(x'/\gamma) &\geq \log \bar{F}(x/\gamma') - \log \bar{F}(x/\gamma' + (x' - x)/\gamma) \\ &\geq \log \bar{F}(x/\gamma') - \log \bar{F}(x'/\gamma') \end{aligned} \tag{7.50}$$

where the first inequality is the same as the previous equation, as the second is by monotonicity. Thus we have done the continuous case.

For $V_\ell^s \sim \lfloor Z + \gamma_\ell^s \rfloor$, we note that $\lfloor Z + \gamma \rfloor \geq x$ if $Z + \gamma \geq \lceil x \rceil$. So the probability is $\bar{F}(\lceil x \rceil - \gamma)$. Similarly, for $V_\ell^s \sim \lfloor \gamma_\ell^s \cdot Z \rfloor$, $\mathbb{P} \lfloor \gamma Z \rfloor \geq x$ is $\bar{F}(\lceil x \rceil / \gamma)$. Thus, if we assume that x and x' are integers, the calculations above go through, as desired. *Q.E.D.*

We present a final fact that justifies the use of $\lfloor Z \rfloor$-type random variables:

Lemma 7.9. If Y is a discrete log-concave random variable, then there exists a continuous log-concave Z such that $Y \sim \lfloor Z \rfloor$.

Proof. Let $P : \mathbb{Z} \to [0, 1]$ be the right-hand cumulative mass function for Y. Then, it suffices to have $\mathbb{P}[Z \geq n] = P(n)$ for all integers n. Let $\phi : \mathbb{R} \to \mathbb{R}$ be the piecewise-linear function such that $\phi(-\infty) \to 0$, $\phi(\infty) \to -\infty$, and $\phi(n) = \log(P(n))$ for all n. Since $\log(P)$ is a discretely concave and non-increasing function, ϕ must be concave and non-increasing. We can then set Z to be the random variable whose density is given by $-\frac{\mathrm{d}}{\mathrm{d}x} \exp(\phi(x))$. *Q.E.D.*

CHAPTER 8: EXTERIOR-POINT METHODS FOR FAIRLY ALLOCATING CHORES[1]

In this chapter, we extend the recent characterization of competitive equilibria of [87] in the setting of mixed manna to give efficient algorithms for finding approximately-competitive equilibria in the setting of bads, *i.e.* when all agents have negative utility for all bundles.

8.1 PRELIMINARIES

In this model, n agents are being allocated bundles from a collection of m chores/bads. There exists 1 unit of each chore, and we wish to distribute them among the agents. Let x_{ij} denote the amount of chore j that agent i is allocated, we want

$$\sum_i x_{ij} = 1 \text{ for all } j \ , \qquad x_{ij} \geq 0 \text{ for all } i, j \ . \tag{8.1}$$

The agents have dis-utilities — *i.e.* negative utilities — for the chores. The dis-utility for agent i is given by the function $d_i(\boldsymbol{x}_i)$, where $\boldsymbol{x}_i := (x_{i1}, \ldots, x_{im})$. We assume the dis-utilities are non-negative, *i.e.* $d_i(\boldsymbol{x}_i) \geq 0$ for all feasible $\boldsymbol{x}_i$. We further assume that they are 1-homogeneous and sub-additive, *i.e.*

$$d_i(a \cdot \boldsymbol{x}_i) = a \cdot d_i(\boldsymbol{x}_i) \ , \qquad d_i(\boldsymbol{x}_i + \widehat{\boldsymbol{x}}_i) \leq d_i(\boldsymbol{x}_i) + d_i(\widehat{\boldsymbol{x}}_i) \tag{8.2}$$

Note that these two conditions together imply convexity as well, which is natural, as positive utility is often assumed to be concave.

If d_i is a linear function, we denote the vector $\boldsymbol{d}_i$ such that $d_i(\boldsymbol{x}_i) = \langle \boldsymbol{d}_i, \boldsymbol{x}_i \rangle$. We denote as d_{ij} the dis-utility to agent i for each unit of chore j. Furthermore, let D be the $n \times nm$ block-diagonal matrix whose blocks are given by the $\boldsymbol{d}_i$'s. If $\boldsymbol{x} = \boldsymbol{x}_1 \cdots \boldsymbol{x}_n$ denotes the total allocation vector, then $D\boldsymbol{x}$ is the n-dimensional vector of dis-utility values for the n players.

In the following, we define a notion of multiplicatively-approximate equilibrium, which agrees with the definition of competitive equilibrium presented in [87] when $\gamma = 1$.

Definition 8.1. An allocation $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is γ-*approximately competitive* if there exists linear rewards for each unit chore $\boldsymbol{p} = (p_1, \ldots, p_m)$, and payouts $\pi_1, \ldots, \pi_n$ such that

$\langle i \rangle$ Every chore is exactly fully allocated, *i.e.* (8.1) holds,

[1] This chapter is the result of as-yet unpublished collaboration with Ruta Mehta. The results of this chapter are being extended in collaboration with Bhaskar Ray Chaudhury and Ruta Mehta, but these extensions will not be included in this thesis, as they are not completed at the time of writing.

$\langle ii \rangle$ All payouts are approximately equal, *i.e.* $\langle \boldsymbol{p}, \boldsymbol{x}_i \rangle = \pi_i$ for all i, and $\pi_i \leq \gamma \pi_j$ for all i and j,

$\langle iii \rangle$ Every agent receives an optimal allocation subject to payment, *i.e.* $d_i(\boldsymbol{x}_i) \leq d_i(\widehat{\boldsymbol{x}}_i)$ for all $\widehat{\boldsymbol{x}}_i \in \mathbb{R}^m_{\geq 0}$ such that $\langle \boldsymbol{p}, \widehat{\boldsymbol{x}}_i \rangle \geq \pi_i$,

$\langle iv \rangle$ The allocation is Pareto-optimal.

$\langle v \rangle$ Any $\widehat{\boldsymbol{x}}_i \in \mathbb{R}^m_{\geq 0}$ such that $\langle \boldsymbol{p}, \widehat{\boldsymbol{x}}_i \rangle \geq \pi_i$ with $d_i(\widehat{\boldsymbol{x}}_i) \leq d_i(\boldsymbol{x})$ has $\langle \boldsymbol{p}, \widehat{\boldsymbol{x}}_i \rangle = \pi_i$, *i.e.* $\boldsymbol{x}_i$ maximizes payout subject to being a min-disutility allocation within the payment constraint,

$\langle vi \rangle$ No agent envies another agent's γ-rescaled allocation, *i.e.* $d_i(\boldsymbol{x}_i) \leq \gamma d_i(\boldsymbol{x}_{i'})$, for every i, i'.

$\langle vii \rangle$ For every agent i, $d_i(\boldsymbol{x}_i) \leq \gamma d_i(\frac{1}{n}\mathbf{1})$.

It is a folklore result that the last conditions in the above definition can be concluded by combining conditions $\langle i \rangle$, $\langle ii \rangle$, and $\langle iii \rangle$, when $\gamma = 1$. We give a brief proof here for completeness, since the definition of approximate equilibria is novel.

Claim 8.1. If conditions $\langle i \rangle$, $\langle ii \rangle$, and $\langle iii \rangle$ hold, then we can conclude the remaining statements above.

Proof. $\langle iv \rangle$: Suppose the allocation is not Pareto-optimal, let $\boldsymbol{x}$ be the allocation, and $\boldsymbol{x}'$ be a Pareto-improvement where some agent, say i, strictly decreases their disutility. Then $\boldsymbol{x}'_i$ must pay strictly less than $\boldsymbol{x}_i$, as otherwise, we would contradict $\langle iii \rangle$. However, taking the sum of this inequality over every agent, we see that the sum of prices paid by all agents must strictly be higher in $\boldsymbol{x}'$ than in $\boldsymbol{x}$, but prices are linear, and in both cases, the sum of prices must be $\langle \boldsymbol{p}, \mathbf{1} \rangle$, a contradiction.

$\langle v \rangle$: Suppose not, and there is an allocation $\widehat{\boldsymbol{x}}_i$ with pay-out greater than π_i, which has disutility at most $d_i(\boldsymbol{x}_i)$, the reference allocation. Then, rescaling $\widehat{\boldsymbol{x}}_i$ to pay exactly π_i contradicts $\langle iii \rangle$.

$\langle vi \rangle$: Fix an allocation $\boldsymbol{x}$ which satisfies the conditions, and fix players i and i'. By $\langle ii \rangle$, $\langle \boldsymbol{p}, \boldsymbol{x}_i \rangle \leq \langle \boldsymbol{p}, \gamma \boldsymbol{x}_{i'} \rangle$. Thus, by $\langle iii \rangle$, $d_i(\boldsymbol{x}_i) \leq d_i(\gamma \boldsymbol{x}_{i'})$, and we conclude the statement by homogeneity.

$\langle vii \rangle$: Payouts are additive, and so the sum of the allocations to each player will be $\mathbf{1}$ and has pay-out $\sum_{k=1}^{n} \pi_k$. Since for any agents i, k, $\pi_i \leq \gamma \pi_i$, then this sum is at most $\gamma \cdot n \cdot \pi_i$, and therefore by $\langle iii \rangle$, agent i must prefer their allocation to $(\gamma/n) \cdot \mathbf{1}$. *Q.E.D.*

[87] show that these conditions hold if and only if the utility profile is a critical point for the Nash welfare on the boundary of the feasible region. Formally:

Theorem 8.2 ([87])**.** Let $\mathcal{F}$ be the feasible space of allocations, *i.e.* the set of points such that (8.1) holds, and let $\mathcal{D}$ be the set of feasible disutility profiles, *i.e.* $\big\{ (d_1(\boldsymbol{x}_1), \ldots, d_n(\boldsymbol{x}_n)) \bigm| \boldsymbol{x} \in \mathcal{F} \big\}$. For some $\boldsymbol{d} \in \mathbb{R}^n$, denote the Nash social welfare as $\operatorname{NSW}(\boldsymbol{d}) := \prod_{i=1}^n \boldsymbol{d}_i$. Then $\boldsymbol{d}$ can be achieved by a competitive equilibrium if and only if the following conditions all hold: a) $\boldsymbol{d} \in \mathcal{D}$, b) $\operatorname{NSW}(\boldsymbol{d}) > 0$, and c) $\boldsymbol{d}$ satisfies the KKT conditions for the problem of minimizing NSW on $\mathcal{D}$. Equivalently, $\boldsymbol{d}$ is on the lower-boundary of $\mathcal{D}$, but not on the boundary of $\mathbb{R}^n_{\geq 0}$, and the gradient $\nabla \operatorname{NSW}(\boldsymbol{d})$ is parallel to some supporting hyperplane for $\mathcal{D}$ at the point $\boldsymbol{d}$.

We extend their theorem to show that if we are approximately KKT, then we satisfy the conditions of Definition 8.1.

8.2 APPROXIMATE KKT SUFFICES

In this section, we define the notion of approximate KKT, and prove that it gives approximately-competitive equilibria. We will repeatedly use the element-wise inverse of a vector, and so we define the following notation: For any two n-dimensional vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$, we denote

$$\boldsymbol{x}/\boldsymbol{y} := (x_1/y_1, \ldots, x_n/y_n) \ . \tag{8.3}$$

We begin by making the simplifying step of considering the logarithm of the Nash social welfare. Note that the gradient for the NSW and its logarithm are proportional, *i.e.*

$$\nabla \operatorname{NSW}(\boldsymbol{d}) = (\boldsymbol{1}/\boldsymbol{d}) \cdot \operatorname{NSW}(\boldsymbol{d}) \propto (\boldsymbol{1}/\boldsymbol{d}) = \nabla(\log \operatorname{NSW})(\boldsymbol{d}) \tag{8.4}$$

where $\boldsymbol{1}/\boldsymbol{d}$ is the component-wise inverse of $\boldsymbol{d}$. Thus, the above theorem holds if we replace the Nash social welfare function with its logarithm. To this end, denote

$$\mathcal{L}(\boldsymbol{d}) := \log(\operatorname{NSW}(\boldsymbol{d})) = \sum_{i=1}^{n} \log(d_i) \ . \tag{8.5}$$

And observe that $\nabla \mathcal{L}(\boldsymbol{d}) = \boldsymbol{1}/\boldsymbol{d}$. Thus, a point satisfies the KKT conditions for minimizing $\mathcal{L}$ on $\mathcal{D}$ if it is on $\mathcal{D}$'s boundary, and $\{\boldsymbol{y} \mid (\boldsymbol{1}/\boldsymbol{d})^\top \boldsymbol{y} \geq n\}$ is a supporting hyperplane for $\mathcal{D}$.

Definition 8.2 (Approximate KKT)**.** We say a point $\boldsymbol{d}$ is a γ-approximate KKT point for the minimization problem on $\mathcal{D}$ if $\boldsymbol{d} \in \mathcal{D}$, there exists a vector $\boldsymbol{a} \in \mathbb{R}^n$ such that $\gamma^{-1} \leq a_i d_i \leq \gamma$ for all i, and $\mathcal{D} \subseteq \{\boldsymbol{y} \mid \boldsymbol{a}^\top \boldsymbol{y} \geq \langle \boldsymbol{a}, \boldsymbol{d} \rangle\}$.

Informally, each entry of $\boldsymbol{a}$ is a γ-approximation of $\mathbf{1}/\boldsymbol{d}$, the gradient of $\mathcal{L}$, and $\boldsymbol{a}$ is a supporting hyperplane for $\mathrm{conv}(\mathcal{D})$. (We will show that there is a convex set C such that $\partial C \subseteq \mathcal{D} \subseteq C$, and so supporting hyperplane is well-defined.)

We show the following, which directly emulates the proof of [87], with this notion of approximate KKT.

Corollary 8.1. Let $\boldsymbol{d}$ be a γ-approximate KKT point for the problem of minimizing $\mathcal{L}$ on $\mathcal{D}$, and $d_i > 0$ for all i. Then any allocation which has dis-utility profile $\boldsymbol{d}$ is a γ^2-approximate competitive equilibrium, as in Definition 8.1.

Proof. As mentioned, this proof follows similarly the arguments presented in [87]. Denote $\boldsymbol{d}(\boldsymbol{x})$ to be the vector of $d_i(\boldsymbol{x}_i)$'s. Recall, we have let $\mathcal{F}$ be the feasible space of allocations, *i.e.* the set of points such that (8.1) holds, and let $\mathcal{D}$ be the set of feasible dis-utility profiles, *i.e.* $\{(d_1(\boldsymbol{x}_1), \ldots, d_n(\boldsymbol{x}_n)) \mid \boldsymbol{x} \in \mathcal{F}\}$.

Let $\boldsymbol{z}$ be any allocation in $\mathcal{F}$ such that $\boldsymbol{d}(\boldsymbol{z})$ is the disutility profile of the approximate KKT point. Now, the set $S_\lambda := \{\boldsymbol{x} \in \mathbb{R}^{nm} \mid \langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{x}) \rangle \leq \lambda\}$ is closed, convex, and non-empty for all $\lambda > 0$, and it does not intersect $\mathcal{F}$ for any $\lambda < \langle \boldsymbol{a}, \boldsymbol{d} \rangle$. Denote $S^* := S_{\langle \boldsymbol{a}, \boldsymbol{d} \rangle}$. We can conclude then that S^* is only tangent to $\mathcal{F}$, since the d_i's are continuous, but $\boldsymbol{z} \in \mathcal{F} \cap S^*$.

Thus, there exists a half-space $H_c := \{\boldsymbol{x} \mid \langle \boldsymbol{c}, \boldsymbol{x} \rangle \geq b\}$ which separates the two sets, *i.e.* $\mathcal{F} \subseteq H_c$, and $S^* \subseteq \mathrm{cl}(H_c^{\complement})$. Note that we must have $\langle \boldsymbol{c}, \boldsymbol{z} \rangle = b$. Define $p_j := \min_i c_{ij}$, and let $\boldsymbol{p} := (p_1, \ldots, p_m)$. We can now show that we satisfy the necessary conditions of Claim 8.1, *i.e.* the first conditions of Definition 8.1.

Condition $\langle \boldsymbol{i} \rangle$ of Definition 8.1. This holds by definition, since $\boldsymbol{z} \in \mathcal{F}$.

Condition $\langle \boldsymbol{ii} \rangle$ of Definition 8.1. We wish to show that all pay-outs are *approximately* equal to each other. Since $p_j \leq c_{ij}$ for all i, j, we have that $H_p := \{\boldsymbol{x} \mid \sum_{ij} p_j x_{ij} \geq b\}$ is a subset of H_c, and so we can conclude that $\langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{z}) \rangle \leq \langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{x}) \rangle$ for all $\boldsymbol{x} \in H_p$, since $\boldsymbol{c}$ separates S from H_p, by definition of $\boldsymbol{c}$ and S^*. Since $\boldsymbol{a}$ is a non-negative vector, this implies that $\boldsymbol{z}$ has less dis-utility than any $\boldsymbol{x} \in H_p$.

Suppose that $\langle \boldsymbol{p}, \boldsymbol{z}_i \rangle < \gamma^{-2} \langle \boldsymbol{p}, \boldsymbol{z}_{i'} \rangle$. Then we could replace the allocation as follows: Construct $\hat{\boldsymbol{z}}$ by setting $\hat{\boldsymbol{z}}_i = \frac{1}{2}\boldsymbol{z}_i$, and $\hat{\boldsymbol{z}}_{i'} = \left(1 + \frac{\langle \boldsymbol{p}, \boldsymbol{z}_i \rangle}{2\langle \boldsymbol{p}, \boldsymbol{z}_{i'} \rangle}\right) \boldsymbol{z}_{i'}$. Notice that $b = \sum_{i,j} p_j z_{ij} = \sum_{i,j} p_j \hat{z}_{ij}$, since the payment subtracted from agent i is equal to the payment added to agent

i', and so $\hat{\boldsymbol{z}} \in H_p$. By $\boldsymbol{z}$'s minimiality in H_p, we must have $\langle \boldsymbol{a}, \boldsymbol{d}(\hat{\boldsymbol{z}})\rangle \geq \langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{z})\rangle$. However,

$$\begin{aligned} \langle \boldsymbol{a}, \boldsymbol{d}(\hat{\boldsymbol{z}})\rangle - \langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{z})\rangle &= -\tfrac{1}{2} a_i d_i(\boldsymbol{z}_i) + \tfrac{\langle \boldsymbol{p}, \boldsymbol{z}_i\rangle}{2\langle \boldsymbol{p}, \boldsymbol{z}_{i'}\rangle} a_{i'} d_{i'}(\boldsymbol{z}_{i'}) \\ &\leq -\tfrac{1}{2}\gamma^{-1} + \tfrac{\langle \boldsymbol{p}, \boldsymbol{z}_i\rangle}{2\langle \boldsymbol{p}, \boldsymbol{z}_{i'}\rangle} \cdot \gamma \\ &< -\tfrac{1}{2}\gamma^{-1} + \tfrac{1}{2}\gamma^{-1} = 0 \end{aligned} \tag{8.6}$$

which is a contradiction. The first inequality is due to the definition of γ-approximate KKT, and the claim that $\boldsymbol{d}(\boldsymbol{z}) = \boldsymbol{d}$ from the statement.

Condition $\langle \boldsymbol{iii}\rangle$ of Definition 8.1. We have that

$$\sum_{i,j} p_j z_{ij} =_{(a)} \sum_j p_j =_{(b)} \sum_j \min_i c_{ij} =_{(c)} \min_{\boldsymbol{x}\in\mathcal{F}} \langle \boldsymbol{c}, \boldsymbol{x}\rangle =_{(d)} b \ , \tag{8.7}$$

where (a) is by the feasibility of $\boldsymbol{z}$, *i.e.* that every chore is allocated exactly one unit, (b) is by definition of $\boldsymbol{p}$, and (c) is by the optimality in $\boldsymbol{c}$ of assigning each chore fully to the agent that has the smallest c_{ij} value for it, and (d) is by assumption on $\boldsymbol{c}$.

Now, suppose condition $\langle iii\rangle$ is not satisfied, *i.e.* $\exists \boldsymbol{x}_i$ for some player i such that $\langle \boldsymbol{p}, \boldsymbol{x}_i\rangle \geq \langle \boldsymbol{p}, \boldsymbol{z}_i\rangle$, and $d_i(\boldsymbol{z}_i) > d_i(\boldsymbol{x}_i)$. If we set $\boldsymbol{z}' := \boldsymbol{z}_1 \circ \cdots \circ \boldsymbol{z}_{i-1} \circ \boldsymbol{x}_i \circ \boldsymbol{z}_{i+1} \circ \cdots \circ \boldsymbol{z}_n$, we will have $\langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{z})\rangle > \langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{z}')\rangle$. However, $\boldsymbol{z}' \in H_p$, which contradicts $\boldsymbol{z}$'s minimality in $H_c \supseteq H_p$. *Q.E.D.*

This concludes the proof of approximately competitive equilibria supported by approximately KKT points. The rest of this writeup consists of an algorithm for finding such disutility profiles. We show first that, when dis-utilities are linear, the above proof is constructive, and the necessary values can be found efficiently.

Corollary 8.2. Let $\boldsymbol{d}$ be a γ-approximate KKT point for the problem of minimizing $\mathcal{L}$ on $\mathcal{D}$, and suppose agent dis-utilities are linear. Let the matrix D be as in the introduction. Then we can efficiently find a γ^2-approximate competitive equilibrium.

Proof. Notice that in the proof of the above result, it suffices to find the separating hyperplane $\{\boldsymbol{x} \mid \langle \boldsymbol{c}, \boldsymbol{x}\rangle \geq b\}$ between the sets S^* and $\mathcal{F}$, and a point $\boldsymbol{z}$ which lies in their intersection. The rest of the proof of the result derives from the knowledge of these values.

Now, when disutilities are linear, we have that $\boldsymbol{d}(\boldsymbol{x}) = D\boldsymbol{x}$, and $\langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{x})\rangle = \left\langle \boldsymbol{a}^\top D, \boldsymbol{x}\right\rangle$. Therefore, the set

$$S_\lambda := \{\boldsymbol{x} \in \mathbb{R}^{nm} \mid \langle \boldsymbol{a}, \boldsymbol{d}(\boldsymbol{x})\rangle \leq \lambda\} \tag{8.8}$$

is exactly the half-space given by $\{\boldsymbol{x} \mid \langle \boldsymbol{a}^\top D, \boldsymbol{x} \rangle \leq \lambda\}$, and we may use $c = D^\top \boldsymbol{a}$. We can find $\boldsymbol{z}$ by minimizing $\langle \boldsymbol{a}^\top D, \boldsymbol{x} \rangle$ on the polytope $\mathcal{F}$, by linear programming methods. *Q.E.D.*

8.3 ALGORITHM, AND CONVERGENCE GUARANTEES

We wish to show that approximate KKT points can be efficiently found. Observe that we could have tried gradient descent for $\mathcal{L}$ or even NSW, with projection on the constraint set $\mathrm{conv}(\mathcal{D})$, since any limit point would satisfy the KKT conditions. Unfortunately, the gradient drives towards points with at least one zero coordinate, which are not meaningful.

Instead, the idea will be to perform a gradient-ascending procedure, on the outside of the feasible region, until we settle on a stable point. Due to the nature of the objective function, we alternate between finding supporting hyperplanes, and finding NSW-maximizing points on these hyperplanes, until we find a point whose gradient is approximately in line with the supporting hyperplane.

In what follows, define $\mathrm{RelDist}(\boldsymbol{x}, \boldsymbol{y}) := \sum_i |\log(x_i/y_i)|$. Notice that if $\mathrm{RelDist}(\boldsymbol{x}, \boldsymbol{y}) \leq \varepsilon$, then $(1+\varepsilon)^{-1} \leq x_i/y_i \leq (1+\varepsilon)$ for all i, since $\log(1+a) \leq a$ for all $a > -1$. We will find a point which is a $(1+\varepsilon)$-approximate KKT point following Algorithm 8.1.

Correctness. We begin by proving here that the algorithm truly returns an approximate KKT point, and we will later show that it will terminate efficiently.

We must first discuss the requirement that step 2 return the nearest point. Note that when cast in allocation space, we seek to find a point $\boldsymbol{x} \in \mathcal{F}$ that minimizes $\sum_{i=1}^n (d_i(\boldsymbol{x}) - (\boldsymbol{d}^k)_i)^2$. Since the d_i's are convex functions, this is a convex objective, that we are minimizing on a polytope defined by $O(m)$ linear constraints. It is therefore not unreasonable that this could be solved exactly. When the d_i's are linear, this can in fact be solved exactly by noticing that the KKT conditions for the problem form a linear feasibility problem [170]. Thus, we assume that we are given black-box access to this nearest point, in the general case.

We now show that in both stopping conditions, we return an approximate KKT point.

Lemma 8.1. Algorithm 8.1 returns a $(1+\varepsilon)$-approximate KKT point for minimizing $\mathcal{L}$ on $\mathcal{D}$.

Proof. We argue later in Claim 8.3 that $\langle \boldsymbol{a}^k, \boldsymbol{y} \rangle \geq n$ can be well-defined as a supporting hyperplane for the set $\mathcal{D}$. Now, suppose the algorithm terminates and returns on line 3. Recall that if $\mathrm{RelDist}(\boldsymbol{x}, \boldsymbol{y}) < \varepsilon$, then $(1+\varepsilon)^{-1} \leq x_i/y_i \leq 1+\varepsilon$ for all i. Thus, by definition of $\boldsymbol{d}^{k+1}$ as the inverse of $\boldsymbol{a}^k$, we have that $\boldsymbol{a}^k$ is a $(1+\varepsilon)$-approximation of the gradient of

Algorithm 8.1: Finding Approximate KKT

```
Let $\boldsymbol{d}^0$ be any infeasible, strictly positive, disutility profile, near $\mathbf{0}$;
for $k = 0, 1, 2, \ldots$ do
    Set $\boldsymbol{d}_*^k$ to be the nearest dominating point in $\mathcal{D}$ to $\boldsymbol{d}^k$, i.e.
        $$\arg\min \left\{ \|\boldsymbol{y} - \boldsymbol{d}^k\|_2^2 \;\middle|\; \boldsymbol{y} \in \mathcal{D},\ \boldsymbol{y} \geq \boldsymbol{d}^k \right\} \tag{8.9}$$
    Set $\boldsymbol{a}^k \leftarrow (\boldsymbol{d}_*^k - \boldsymbol{d}^k)$, the direction from $\boldsymbol{d}^k$ to $\mathcal{D}$;
    Rescale $\boldsymbol{a}^k$ so that $\langle \boldsymbol{a}^k, \boldsymbol{d}_*^k \rangle = n$;
    if RelDist$(\boldsymbol{d}^{k+1}, \boldsymbol{d}_*^k) < \varepsilon$ then return $(\boldsymbol{d}_*^k, \boldsymbol{a}^k)$);
    Set $\boldsymbol{d}^{k+1} \leftarrow \mathbf{1}/\boldsymbol{a}^k$ ;
end
```

$\mathcal{L}$ at $\boldsymbol{d}^k$, and is a supporting hyperplane passing through $\boldsymbol{d}_*^k$, satisftying the conditions of Definition 8.2. *Q.E.D.*

Efficiency. We wish to argue that (a) the log-NSW $\mathcal{L}$ is always increasing throughout Algorithm 8.1, and (b) it will terminate in few iterations. We will first show (a) below, then bounding the rate at which it is advancing the objective, to conclude (b). We must however begin with a technical claim.

Claim 8.3. For all $k \geq 0$, the point $\boldsymbol{d}^k$ has strictly positive coordinates, lies outside $\mathcal{D}$, and is dominated by some point in $\mathcal{D}$.

Proof. We first argue that $\mathcal{D}$'s lower-hull is convex. Note that $\mathcal{F}$ is convex, and we have assumed that the d_i's are convex functions. Therefore, for any $0 < \lambda < 1$, we have $d_i(\lambda \boldsymbol{x}_i + (1-\lambda)\widehat{\boldsymbol{x}}_i) \leq \lambda d_i(\boldsymbol{x}_i) + (1-\lambda) d_i(\widehat{\boldsymbol{x}}_i)$. Thus, the vector $\boldsymbol{d}(\lambda \boldsymbol{x} + (1-\lambda)\widehat{\boldsymbol{x}})$ is dominated by $\lambda \boldsymbol{d}(\boldsymbol{x}) + (1-\lambda)\boldsymbol{d}(\widehat{\boldsymbol{x}})$. We conclude therefore that the minkowski sum $\mathcal{D} + \mathbb{R}_{\geq 0}$ is a convex set.

We also note that $\mathcal{D}$ contains every rescaled basis vector: For every player i, we can feasibly allocate all chores to i, and so $d_i(\mathbf{1}) \cdot \boldsymbol{e}_i$ is contained in $\mathcal{D}$. Furthemore, for any allocation $\boldsymbol{x} \in \mathcal{F}$, each agent has disutility less than $d_i(\mathbf{1})$, by sub-additivity. Finally, if $\mathcal{D} + \mathbb{R}_{\geq 0}$ has a supporting hyperplane $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$, where $\boldsymbol{a}$ has a 0 component, then we must have $b \leq 0$, as otherwise we would exclude the basis vector for that component.

Now, we prove the statement, by induction on k. It is ensured for $k = 0$ by construction. Suppose $\boldsymbol{d}^k$ has strictly positive coordinates, does not lie in $\mathcal{D}$, and there is some $\boldsymbol{y} \in \mathcal{D}$ such that $\boldsymbol{y} \geq \boldsymbol{d}^k$, coordinate-wise. By the above discussion, $(\boldsymbol{d}^k)_i \leq (\boldsymbol{y})_i \leq d_i(\mathbf{1})$ for all i, and so any hyperplane separating $\boldsymbol{d}^k$ from $\mathcal{D} + \mathbb{R}_{\geq 0}$ must be positively oriented, as otherwise, it could not separate $\boldsymbol{d}^k$ from the basis vectors.

Now, since $\boldsymbol{d}_*^k$ is the closest point in $\mathcal{D}$ to $\boldsymbol{d}^k$, it will also be the closest point in $\mathcal{D}+\mathbb{R}_{\geq 0}$ to $\boldsymbol{d}^k$, and the vector between them, namely $\boldsymbol{a}^k$, must be orthogonal to the boundary of $\mathcal{D}$ at $\boldsymbol{d}_*^k$. Thus, $\langle \boldsymbol{a}^k, \boldsymbol{y} \rangle \geq n$ is a supporting hyperplane for $\mathcal{D}+\mathbb{R}_{\geq 0}$. By the above discussion, this implies that $\boldsymbol{a}^k$ is a strictly positive vector, and therefore that $\boldsymbol{d}^{k+1}$ is also strictly positive. Since $\langle \boldsymbol{a}^k, \boldsymbol{y} \rangle \geq n$ is supporting for $\mathcal{D}$, and $\langle \boldsymbol{a}^k, \boldsymbol{d}^{k+1} \rangle = n$, then $\boldsymbol{d}^{k+1}$ is also not contained in $\mathcal{D}$ and dominated by it. As desired. *Q.E.D.*

Claim 8.4. Steps 2. and 4. always increase the Nash social welfare.

Proof. Step 2 is moving in a positive direction, and the log-NSW is monotone in each coordinate, so this will always be an improvement. We prove that Step 4 is an improvement by showing that $\boldsymbol{d}^{k+1}$ is the maximizing point on the hyperplane $\langle \boldsymbol{a}^k, \boldsymbol{y} \rangle = n$. In fact, since log-NSW is a concave function, it is maxmized on this hyperplane when $\nabla \mathcal{L}$ is proportional to $\boldsymbol{a}^k$, *i.e.* when $a_i^k = c/d_i$ for some $c > 0$, for all i. Since we need $\langle \boldsymbol{a}^k, \boldsymbol{d} \rangle = n$, it suffices to set $c = 1$. Thus, $\boldsymbol{d}^{k+1}$ is the $\mathcal{L}$-maximizing point on the supporting hyperplane which contains $\boldsymbol{d}_*^k$, and so this move is an $\mathcal{L}$-improvement. *Q.E.D.*

Lemma 8.2. If Algorithm 8.1 does not return at step 3, then the logarithm of the Nash social welfare increases by at least $\frac{1}{16}(\varepsilon/n)^2$

Proof. We wish to show that if $\operatorname{RelDist}(\boldsymbol{d}^{k+1}, \boldsymbol{d}_*^k) > \varepsilon$, then $\mathcal{L}(\boldsymbol{d}^{k+1}) - \mathcal{L}(\boldsymbol{d}_*^k)$ is large. Let $A = \operatorname{diag}(\boldsymbol{a}^k)$, and note that $\langle \mathbf{1}, A\boldsymbol{d} \rangle = \langle \boldsymbol{a}^k, \boldsymbol{d} \rangle$, and furthermore, $A\boldsymbol{d}^{k+1} = \mathbf{1}$. Let $\boldsymbol{\Delta} = A\boldsymbol{d}_*^k - \mathbf{1}$, and notice that

$$\langle \mathbf{1}, \boldsymbol{\Delta} \rangle = \left\langle \mathbf{1}, A(\boldsymbol{d}_*^k - \boldsymbol{d}^{k+1}) \right\rangle = 0 \tag{8.10}$$

Note that $\boldsymbol{d}_*^k = (\mathbf{1}+\boldsymbol{\Delta})/\boldsymbol{a}^k$, where we take the quotient componentwise. With $\boldsymbol{d}^{k+1} = \mathbf{1}/\boldsymbol{a}^k$, this gives $\operatorname{RelDist}(\boldsymbol{d}_*^k, \boldsymbol{d}^{k+1}) = \operatorname{RelDist}((\mathbf{1} + \boldsymbol{\Delta}), \mathbf{1})$. Thus, we have $\sum_{i=1}^n |\log(1 + \Delta_i)| > \varepsilon$. We also get

$$\mathcal{L}(\boldsymbol{d}^{k+1}) - \mathcal{L}(\boldsymbol{d}_*^k) = \sum_{i=1}^n \log(1/a_i^k) - \log((1 + \Delta_i)/a_i^k) = -\sum_{i=1}^n \log(1 + \Delta_i) \ . \tag{8.11}$$

Define:

$$F(a) := \begin{cases} \frac{1}{4}a^2 & \text{if } -1 < a \leq 1 \\ \frac{1}{2}a - \frac{1}{4} & \text{if } a \geq 1 \\ +\infty & \text{otherwise} \end{cases} \tag{8.12}$$

At $a = 0$, we have that $-a + F(a) = 0 = \log(1+a)$ and $\frac{\mathrm{d}}{\mathrm{d}a}(-a+F(a)) = -1 = \frac{\mathrm{d}}{\mathrm{d}a}(-\log(1+a))$. By comparing derivatives for the other values of $a > -1$, we can show that $-\log(1+a) \geq -a + F(a)$ for all a. Thus,

$$\mathcal{L}(\boldsymbol{d}^{k+1}) - \mathcal{L}(\boldsymbol{d}^k_*) \; = \; -\sum_{i=1}^n \log(1+\Delta_i) \; \geq \; \sum_{i=1}^n -\Delta_i + \sum_{i=1}^n F(\Delta_i) \; = \; \sum_{i=1}^n F(\Delta_i) \tag{8.13}$$

Now, since we have $\sum_{i=1}^n |\log(1+\Delta_i)| > \varepsilon$, there must be some i such that $|\log(1+\Delta_i)| > \varepsilon/n$. If $\Delta_i > 0$, then $\Delta_i \geq \log(1+\Delta_i) \geq \varepsilon/n$. Conversely, if $\Delta_i < 0$, we being by noting that for $|a| < 0.5$, we have $-\log(1+a) \leq -a + a^2$ for reasons similar to the above. Thus, we get

$$\varepsilon/n < -\log(1+\Delta_i) \leq -\Delta_i + \Delta_i^2 \tag{8.14}$$

We must have $\Delta_i > -1$, since the argument can't be negative, so we have $2|\Delta_i| > \Delta_i^2 - \Delta_i > \varepsilon/n$, or $\Delta_i < -\frac{1}{2}\varepsilon/n$. Noting that $F(a) \geq 0$ for all a, we can then conclude

$$\mathcal{L}(\boldsymbol{d}^{k+1}) - \mathcal{L}(\boldsymbol{d}^k_*) \geq \textstyle\sum_i F(\Delta_i) \geq \max_i F(\Delta_i) \geq \frac{1}{16}\varepsilon^2/n^2 \tag{8.15}$$

as desired. *Q.E.D.*

Corollary 8.3. Suppose $d_i(\boldsymbol{x}_i) \leq M$ for every agent i, at every feasible $\boldsymbol{x}$. Then the above procedure finds a $(1+\varepsilon)$-approximate competitive equilibrium in $O(n^2/\varepsilon^2 \cdot (n \log M - \mathcal{L}(\boldsymbol{d}^0)))$ iterations.

Proof. If we can bound the range of the log-NSW, then the above lemma will suffice. By assumption, we have that for any feasible $\boldsymbol{x}$, $\mathcal{L}(\boldsymbol{d}(\boldsymbol{x})) \leq n \log M$. Since each round of the above algorithm that doesn't terminate increases the log-NSW by at least $\frac{2}{9}(\varepsilon/n)^2$, then the total number of rounds possible is at most

$$16 \cdot \frac{n^2}{\varepsilon^2} \cdot (n \log(M) - \mathcal{L}(\boldsymbol{d}^0)) \tag{8.16}$$

as desired. *Q.E.D.*

Note that when disutilities are linear, we can bound M by $\max_i \sum_{j=1}^m d_{ij}$, and assuming $d_{ij} > \delta$ for all i and j, then any feasible dis-utility profile must assign disutility at least $m\delta/n$ to one player, and so an allocation $\boldsymbol{x}$ such that $\boldsymbol{d}(\boldsymbol{x}) = \frac{1}{2}(m\delta/n)\mathbf{1}$ can not be feasible. Such an $\boldsymbol{x}$ can be constructed by assigning a small fraction of any of the items to each player, and therefore it is a valid starting position for the algorithm, and the total running time will therefore be bounded by $O\big(n^3(\log(M) + \log(n/\delta m)/\varepsilon^2\big)$.

CHAPTER 9: CONCLUSIONS AND FUTURE DIRECTIONS

9.1 SMOOTHED ANALYSIS FOR NASH EQUILIBRIUM COMPUTATION

The Smoothed Complexity of 2-Player Nash. In the first part of this thesis, we considered the smoothed analysis of Nash equilibrium computation. In the classic setting of 2-player, normal-form games with $n \times n$ payoff matrices, it was known that when payoff values are $O(1)$, then no smoothed-efficient algorithm would exist for poly$(1/n)$-sized perturbations under standard complexity assumptions, as it implies efficient algorithms for PPAD-complete problems. We showed in Chapter 2 that under these same assumptions, there exists a constant $\epsilon > 0$ such that no smoothed-efficient algorithm would exist for ϵ-sized perturbations, settling a conjecture of [35]. In a classic result, [25] show that when the entries are fully random — or equivalently, when randomness is $\Omega(1)$-sized and payoffs are 0 — then equilibria are small and can be found by exhaustive search. Along with our work, this leaves the large-noise regime as an open problem: When payoffs are poly$(1/n)$, and noise is $\Omega(1)$, do the results of [25] extend? Furthermore, does there exist $\delta > 0$ constant such that when payoffs lie in $[0, \delta]$, and perturbations are $\Omega(1)$, equilibria can be found efficiently?

Smoothed Efficient Algorithms for PLS. In Chapter 3, we gave smoothed-efficient algorithms for finding pure Nash equilibria in network coordination games, establishing the commonality of the framework introduced in Section 3.1. For n-player, m-edge, network coordination games, with $k \times k$ edge-games, we showed that pure equilibria could be found in time respectively polynomial and quasi-polynomial in n^k, when game graphs were complete, or not complete. We leave as an open question whether the exponential dependence on k can be improved, and note that the reduction to local-max-bisection is one possible method for this.

The common framework of Section 3.1 had, prior to our work, been used to prove multiple results on the smoothed complexity of finding a locally maximal cut in a graph [31, 32, 107, 108]. In Chapter 4 to give a first smoothness-preserving reduction between problems. We show that if a problem can be reduced by a linear transformation to any problem whose smoothed-efficiency is proven via the common framework, then it inherits this proof, and also admits smoothed-efficient algorithms.

It is a natural question to ask whether other PLS-complete problems can be analyzed in this way. In a brief collaboration with Karthik Chandrasekaran, we have shown that the analysis for local max cut extends almost exactly to Local Max Cut for 3-regular hypergraphs.

Though this problem seems of tangential interest, it was one of the first PLS-complete problems, and may be a good candidate for reductions. It is, therefore, also worth asking whether other problems admit smoothness-preserving reductions to local max cut.

9.2 DECISION THEORY PROBLEMS AND PRICING PROBLEMS

The Pandora's Box Problem with Order Constraints. In Chapter 5, we considered the classical Pandora's box problem of Weitzman [64], and added order constraints. These constraints are modelled as a DAG over the boxes, where a box can be opened if one of its parents has been opened. In the case when the constrain graph formed a forest of rooted trees, we showed the optimal order was fixed *a priori*, and the optimal strategy could be efficiently computed. When the constraint graph was a general DAG, we showed that the optimal search order may not be fixed in advance, and that optimal adaptive policies are NP-hard to approximate.

This presents many future directions of work, chief among which is asking what happens in the more natural order constraint model where *all* parents of a node must be opened before it can be. In this model, it is not known whether the optimal order of search is fixed or not, and it is not known whether optimal strategies can be efficiently found, or approximated. Further questions include any of the generalizations of the model and objective function found in the literature [116, 127, 133, 171, 172], or refining the analyses of Chapter 5 to improve the constant of approximation-hardness.

Optimal Binary Classifier Search from Pairwise Comparisons. In Chapter 6, we showed how to efficiently find optimal binary classifiers, when the objective function is not known to the algorithm, but instead we have access to a noisy comparison oracle. Naturally, one may ask whether the results can be extended to k-ary classifiers. This was the subject of follow-up work [95]. It may also be interesting to ask whether the results can be extended to more general classes of objective functions.

Optimal Server Pricing for Revenue Maximization. In Chapter 7, we studied the problem of pricing time on a single server to maximize revenue. We reduced the problem to solving a Markov decision process, and give natural assumptions under which the decision process can be optimally determined efficiently. The model restricted the attention to a single-server setting with one job arriving per time step. Future directions of work would extend this model to a many-server setting, with multiple jobs arriving at a time. The naive, exponential Markov chain would not greatly grow in size, but it would require showing that the simplifications which lead to polynomial-time performance will extend in these settings, and that the optimal solution remains monotone.

Competitive Equilibria for Bads with Almost Equal Income. Finally, in Chapter 8, we gave an efficient algorithm to find a fair and efficient allocation of chores to agents, in an approximately-competitive equilibrium. This algorithm uses a recent characterization by [87] of competitive equilibria in the language of convex optimization, and finds points which approximately satisfy the constraints. The results are only explicit in the linear setting, and require exact solutions to convex optimization problems. A more robust result would extend the methods to show that approximately-competitive equilibria can be found when all steps of the algorithm are only given approximately.

REFERENCES

[1] R. B. Myerson, *Game Theory: Analysis of Conflict.* Harvard University Press, 1997.

[2] C. H. Papadimitriou, “On the complexity of the parity argument and other inefficient proofs of existence,” *Journal of Computer and system Sciences*, vol. 48, no. 3, pp. 498–532, 1994.

[3] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, “The complexity of computing a nash equilibrium,” *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[4] X. Chen, X. Deng, and S.-H. Teng, “Settling the complexity of computing two-player Nash equilibria,” *J. ACM*, vol. 56, no. 3, pp. 14:1–14:57, 5 2009. [Online]. Available: http://doi.acm.org/10.1145/1516512.1516516

[5] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior, 2nd rev.* Princeton University Press, 1947.

[6] T. Roughgarden et al., “Complexity theory, game theory, and economics: The barbados lectures,” *Foundations and Trends® in Theoretical Computer Science*, vol. 14, no. 3–4, pp. 222–407, 2020.

[7] J. F. Nash, “Equilibrium points in n-person games,” *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[8] C. E. Lemke and J. T. Howson, Jr, “Equilibrium points of bimatrix games,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, 1964.

[9] R. J. Lipton, E. Markakis, and A. Mehta, “Playing large games using simple strategies,” in *Proceedings of the 4th ACM Conference on Electronic Commerce*, ser. EC ’03. New York, NY, USA: ACM, 2003. [Online]. Available: http://doi.acm.org/10.1145/779928.779933 pp. 36–41.

[10] A. Rubinstein, “Settling the complexity of computing approximate two-player Nash equilibria,” *SIGecom Exch.*, vol. 15, no. 2, pp. 45–49, 2 2017. [Online]. Available: http://doi.acm.org/10.1145/3055589.3055596

[11] S. Barman, “Approximating nash equilibria and dense subgraphs via an approximate version of carathéodory’s theorem,” *SIAM J. Comput.*, vol. 47, no. 3, pp. 960–981, 2018. [Online]. Available: https://doi.org/10.1137/15M1050574

[12] R. Kannan and T. Theobald, “Games of fixed rank: A hierarchy of bimatrix games,” *Economic Theory*, vol. 42, no. 1, pp. 157–173, 2010.

[13] B. Adsul, J. Garg, R. Mehta, and M. Sohoni, "Rank-1 bimatrix games: a homeomorphism and a polynomial time algorithm," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 2011, pp. 195–204.

[14] N. Alon, T. Lee, A. Shraibman, and S. S. Vempala, "The approximate rank of a matrix and its algorithmic applications: approximate rank," in *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, 2013. [Online]. Available: https://doi.org/10.1145/2488608.2488694 pp. 675–684.

[15] C. Daskalakis and C. H. Papadimitriou, "Approximate nash equilibria in anonymous games," *J. Econ. Theory*, vol. 156, pp. 207–245, 2015. [Online]. Available: https://doi.org/10.1016/j.jet.2014.02.002

[16] Y. Cheng, I. Diakonikolas, and A. Stewart, "Playing anonymous games using simple strategies," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, 2017. [Online]. Available: https://doi.org/10.1137/1.9781611974782.40 pp. 616–631.

[17] E. Elkind, L. A. Goldberg, and P. W. Goldberg, "Nash equilibria in graphical games on trees revisited," in *Proceedings 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*, 2006. [Online]. Available: https://doi.org/10.1145/1134707.1134719 pp. 100–109.

[18] S. Barman, K. Ligett, and G. Piliouras, "Approximating nash equilibria in tree polymatrix games," in *Algorithmic Game Theory - 8th International Symposium, SAGT 2015, Saarbrücken, Germany, September 28-30, 2015, Proceedings*, 2015. [Online]. Available: https://doi.org/10.1007/978-3-662-48433-3_22 pp. 285–296.

[19] L. E. Ortiz and M. T. Irfan, "FPTAS for mixed-strategy nash equilibria in tree graphical games and their generalizations," *CoRR*, vol. abs/1602.05237, 2016. [Online]. Available: http://arxiv.org/abs/1602.05237

[20] X. Chen, X. Deng, and S.-H. Teng, "Sparse games are hard," in *International Workshop on Internet and Network Economics.* Springer, 2006, pp. 262–273.

[21] Z. Liu and Y. Sheng, "On the approximation of nash equilibria in sparse win-lose games," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[22] R. Mehta, "Constant rank bimatrix games are ppad-hard," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 545–554.

[23] X. Chen, D. Durfee, and A. Orfanou, "On the complexity of nash equilibria in anonymous games," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 2015. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746571 pp. 381–390.

[24] A. Deligkas, J. Fearnley, and R. Savani, "Tree polymatrix games are ppad-hard," *CoRR*, vol. abs/2002.12119, 2020. [Online]. Available: https://arxiv.org/abs/2002.12119

[25] I. Bárány, S. Vempala, and A. Vetta, "Nash equilibria in random games," *Random Structures and Algorithms*, vol. 31, no. 4, pp. 391–405, 12 2007. [Online]. Available: http://dx.doi.org/10.1002/rsa.v31:4

[26] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *J. ACM*, vol. 51, no. 3, pp. 385–463, 5 2004. [Online]. Available: http://doi.acm.org/10.1145/990308.990310

[27] V. Klee and G. J. Minty, "How good is the simplex algorithm," *Inequalities*, vol. 3, no. 3, pp. 159–175, 1972.

[28] M. Englert, H. Röglin, and B. Vöcking, "Worst case and probabilistic analysis of the 2-opt algorithm for the tsp," *Algorithmica*, vol. 68, no. 1, pp. 190–264, 2014.

[29] D. Arthur, B. Manthey, and H. Röglin, "Smoothed analysis of the k-means method," *Journal of the ACM (JACM)*, vol. 58, no. 5, pp. 1–31, 2011.

[30] A. Andoni and R. Krauthgamer, "The smoothed complexity of edit distance," *ACM Transactions on Algorithms (TALG)*, vol. 8, no. 4, pp. 1–25, 2012.

[31] M. Etscheid and H. Röglin, "Smoothed analysis of local search for the maximum-cut problem," *ACM Transactions on Algorithms (TALG)*, vol. 13, no. 2, 2017.

[32] O. Angel, S. Bubeck, Y. Peres, and F. Wei, "Local max-cut in smoothed polynomial time," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 429–437.

[33] H. Röglin and B. Vöcking, "Smoothed analysis of integer programming," *Mathematical programming*, vol. 110, no. 1, pp. 21–56, 2007.

[34] R. Beier and B. Vöcking, "Typical properties of winners and losers [0.2ex] in discrete optimization," *SIAM Journal on Computing*, vol. 35, no. 4, pp. 855–881, 2006. [Online]. Available: https://doi.org/10.1137/S0097539705447268

[35] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms and heuristics," *London Mathematical Society Lecture Note Series*, vol. 1, no. 331, pp. 274–342, 2006.

[36] D. A. Spielman and S.-H. Teng, "Smoothed analysis: an attempt to explain the behavior of algorithms in practice," *Communications of the ACM*, vol. 52, no. 10, pp. 76–84, 2009.

[37] B. Manthey and H. Röglin, "Smoothed analysis: analysis of algorithms beyond worst case," *it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, vol. 53, no. 6, pp. 280–286, 2011.

[38] M.-F. Balcan and M. Braverman, "Nash equilibria in perturbation-stable games," *Theory of Computing*, vol. 13, no. 1, pp. 1–31, 2017.

[39] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[40] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, "How easy is local search?" *Journal of Computer and System Sciences*, vol. 37, no. 1, pp. 79 – 100, 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022000088900463

[41] A. A. Schäffer and M. Yannakakis, "Simple local search problems that are hard to solve," *SIAM J. Comput.*, vol. 20, no. 1, pp. 56–87, 2 1991. [Online]. Available: http://dx.doi.org/10.1137/0220004

[42] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi, "The relative complexity of NP search problems," *Journal of Computer and System Sciences*, vol. 57, no. 1, pp. 3 – 19, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000098915756

[43] N. Bitansky, O. Paneth, and A. Rosen, "On the cryptographic hardness of finding a Nash equilibrium," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, 10 2015, pp. 1480–1498.

[44] B. Codenotti, S. De Rossi, and M. Pagan, "An experimental analysis of Lemke-Howson algorithm," 2008, arXiv preprint arXiv:0811.3247.

[45] A. Deligkas, J. Fearnley, T. P. Igwe, and R. Savani, "An empirical study on computing equilibria in polymatrix games," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, ser. AAMAS '16. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2936924.2936955 pp. 186–195.

[46] R. Savani and B. von Stengel, "Exponentially many steps for finding a Nash equilibrium in a bimatrix game," in *IEEE Annual Symposium on Foundations of Computer Science*, 2004, pp. 258–267.

[47] R. Arava, "Social network analysis using coordination games," *arXiv preprint arXiv:1708.09570*, 2017.

[48] A. Montanari and A. Saberi, "The spread of innovations in social networks," *Proceedings of the National Academy of Sciences*, vol. 107, no. 47, pp. 20 196–20 201, 2010.

[49] D. Kasthurirathna, M. Piraveenan, and M. Harré, "Influence of topology in the evolution of coordination in complex networks under information diffusion constraints," *The European Physical Journal B*, vol. 87, no. 1, p. 3, 2014.

[50] M. J. Smith, "The existence, uniqueness and stability of traffic equilibria," *Transportation Research Part B: Methodological*, vol. 13, no. 4, pp. 295–304, 1979.

[51] T. Roughgarden, "Routing games," pp. 461–486, 2007.

[52] T. Harks, M. Hoefer, M. Klimm, and A. Skopalik, "Computing pure Nash and strong equilibria in bottleneck congestion games," *Mathematical Programming*, vol. 141, no. 1, pp. 193–215, 2013.

[53] J. Broere, V. Buskens, J. Weesie, and H. Stoof, "Network effects on coordination in asymmetric games," *Scientific reports*, vol. 7, no. 1, p. 17016, 2017.

[54] G. Ellison, "Learning, local interaction, and coordination," *Econometrica: Journal of the Econometric Society*, pp. 1047–1071, 1993.

[55] S. Choi, D. Gale, S. Kariv, and T. Palfrey, "Network architecture, salience and coordination," *Games and Economic Behavior*, vol. 73, no. 1, pp. 76–90, 2011.

[56] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler, "Near-optimal network design with selfish agents," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing.* ACM, 2003, pp. 511–520.

[57] Y. Cai and C. Daskalakis, "On minmax theorems for multiplayer games," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms.* SIAM, 2011, pp. 217–234.

[58] S. Boodaghians, R. Kulkarni, and R. Mehta, "Smoothed efficient algorithms and reductions for network coordination games," in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, 2020.

[59] K. R. MacCrimmon, "Descriptive and normative implications of the decision-theory postulates," in *Risk and uncertainty.* Springer, 1968, pp. 3–32.

[60] K. Steele and H. O. Stefánsson, "Decision Theory," in *The Stanford Encyclopedia of Philosophy*, Winter 2020 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2020.

[61] M. Gardner, "New mathematical diversions from scientific american," *Simon and Shuster*, 1966.

[62] T. S. Ferguson, "Who solved the secretary problem?" *Statistical science*, vol. 4, no. 3, pp. 282–289, 1989.

[63] M. O. Jackson, "Mechanism theory," *Available at SSRN 2542983*, 2014.

[64] M. L. Weitzman, "Optimal search for the best alternative," *Econometrica: Journal of the Econometric Society*, pp. 641–654, 1979.

[65] I. Steinwart, "How to compare different loss functions and their risks," *Constructive Approximation*, vol. 26, no. 2, pp. 225–287, 2007.

[66] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[67] F. Janssen and J. Furnkranz, “On meta-learning rule learning heuristics,” in *ICDM*. IEEE, 2007, pp. 529–534.

[68] M. Peyrard, T. Botschen, and I. Gurevych, “Learning to score system summaries for better content selection evaluation.” in *Proceedings of the Workshop on New Frontiers in Summarization*, 2017, pp. 74–84.

[69] R. Caruana and A. Niculescu-Mizil, “Data mining in metric space: an empirical analysis of supervised learning performance criteria,” in *ACM SIGKDD*. ACM, 2004, pp. 69–78.

[70] C. Ferri, J. Hernández-Orallo, and R. Modroiu, “An experimental comparison of performance measures for classification,” *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, 2009.

[71] X. Tang, X. Li, and Z. Fu, “Budget-constraint stochastic task sched. on heterogeneous cloud systems,” *Concurrency and Comp.: Practice and Experience*, vol. 29, no. 19, 2017.

[72] M. Babaioff, Y. Mansour, N. Nisan, G. Noti, C. Curino, N. Ganapathy, I. Menache, O. Reingold, M. Tennenholtz, and E. Timnat, “Era: A framework for economic resource allocation for the cloud,” in *Proceedings of the 26th WWW Companion*, 2017, pp. 635–642.

[73] S. Chawla, N. R. Devanur, A. E. Holroyd, A. R. Karlin, J. B. Martin, and B. Sivan, “Stability of service under time-of-use pricing,” in *Procs. of the 49th Annual ACM SIGACT Symp. on Theory of Computing*. ACM, 2017, pp. 184–197.

[74] Y. Azar, I. Kalp-Shaltiel, B. Lucier, I. Menache, J. Naor, and J. Yaniv, “Truthful online scheduling with commitments,” in *EC*, 2015.

[75] N. Jain, I. Menache, J. Naor, and J. Yaniv, “A truthful mechanism for value-based scheduling in cloud computing,” *Theory of Computing Systems*, vol. 54, pp. 388–406, 2013.

[76] T. Sandholm and A. Gilpin, *Sequences of Take-It-or-Leave-It Offers: Near-Optimal Auctions Without Full Valuation Revelation*. Springer, 2004, pp. 73–91.

[77] L. Blumrosen and T. Holenstein, “Posted prices vs. negotiations: an asymptotic analysis,” in *Proceedings of the 9th EC*. ACM, 2008, pp. 49–49.

[78] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan, “Multi-parameter mechanism design and sequential posted pricing,” in *Proc’s of the 42nd STOC*. ACM, 2010.

[79] R. Kleinberg and S. M. Weinberg, “Matroid prophet inequalities,” in *Proceedings of the 44th STOC*. ACM, 2012, pp. 123–136.

[80] P. Dütting and R. Kleinberg, “Polymatroid prophet inequalities,” in *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA)*, 2015, pp. 437–449.

[81] M. Feldman, N. Gravin, and B. Lucier, "Combinatorial auctions via posted prices," in *Proceedings of the 26th SODA*. ACM-SIAM, 2015, pp. 123–135.

[82] H. Steinhaus, "Sur la division pragmatique," *Econometrica: Journal of the Econometric Society*, pp. 315–319, 1949.

[83] R. Abebe, "Cake cutting: Equitable simple allocations of heterogeneous goods," 2014.

[84] K. J. Arrow et al., "An extension of the basic theorems of classical welfare economics," in *Proceedings of the second Berkeley symposium on mathematical statistics and probability*. The Regents of the University of California, 1951.

[85] E. Eisenberg and D. Gale, "Consensus of subjective probabilities: The pari-mutuel method," *The Annals of Mathematical Statistics*, vol. 30, no. 1, pp. 165–168, 1959.

[86] E. Eisenberg, "Aggregation of utility functions," *Management Science*, vol. 7, no. 4, pp. 337–350, 1961.

[87] A. Bogomolnaia, H. Moulin, F. Sandomirskiy, and E. Yanovskaya, "Competitive division of a mixed manna," *ECONOMETRICA*, pp. 1847–1871, 2017.

[88] N. R. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani, "Market equilibrium via a primal-dual-type algorithm," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE, 2002, pp. 389–395.

[89] R. Duan, J. Garg, and K. Mehlhorn, "An improved combinatorial polynomial algorithm for the linear arrow-debreu market," in *Proceedings of the twenty-seventh annual ACM-SIAM symposium on discrete algorithms.* SIAM, 2016, pp. 90–106.

[90] J. Garg and L. A. Végh, "A strongly polynomial algorithm for linear exchange markets," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 54–65.

[91] S. Brânzei and F. Sandomirskiy, "Algorithms for competitive division of chores," *arXiv preprint arXiv:1907.01766*, 2019.

[92] J. Garg and P. McGlaughlin, "Computing competitive equilibria with mixed manna," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 420–428.

[93] B. R. Chaudhury, J. Garg, P. McGlaughlin, and R. Mehta, "Competitive allocation of a mixed manna," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA).* SIAM, 2021, pp. 1405–1424.

[94] X. Deng and Y. Du, "The computation of approximate competitive equilibrium is ppad-hard," *Information Processing Letters*, vol. 108, no. 6, pp. 369–373, 2008.

[95] G. Hiranandani, S. Boodaghians, R. Mehta, and O. O. Koyejo, "Multiclass performance metric elicitation," in *Advances in Neural Information Processing Systems*, 2019, pp. 9351–9360.

[96] S. Boodaghians, F. Fusco, P. Lazos, and S. Leonardi, "Pandora's box problem with order constraints," in *Submitted to EC 2020*, 2020.

[97] S. Boodaghians, J. Brakensiek, S. Hopkins, and A. Rubinstein, "Smoothed complexity of 2-player nash equilibria," in *Submitted to FOCS 2020*, 2020.

[98] S. Boodaghians, F. Fusco, S. Leonardi, Y. Mansour, and R. Mehta, "Online revenue maximization for server pricing," in *IJCAI 2020*, 2020.

[99] D. P. Roberts, "Nash equilibria of cauchy-random zero-sum and coordination matrix games," *International Journal of Game Theory*, vol. 34, no. 2, pp. 167–184, 2006.

[100] J. Jonasson, "On the optimal strategy in a random game," *Electronic Communications in Probability*, vol. 9, pp. 132–139, 2004.

[101] P. Erdös, "On a lemma of littlewood and offord," *Bulletin of the American Mathematical Society*, vol. 51, no. 12, pp. 898–902, 1945.

[102] D. Dzindzalieta, "Tight bernoulli tail probability bounds," Ph.D. dissertation, Vilnius University, 2014.

[103] R. B. Ash, *Information Theory.* Dover Publications, Inc., 1990.

[104] R. Vershynin, *High-dimensional probability: An introduction with applications in data science.* Cambridge university press, 2018, vol. 47.

[105] M. J. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint.* Cambridge University Press, 2019, vol. 48.

[106] M. Rudelson and R. Vershynin, "Small ball probabilities for linear images of high-dimensional distributions," *International Mathematics Research Notices*, vol. 2015, no. 19, pp. 9594–9617, 2015.

[107] X. Chen, C. Guo, E.-V. Vlatakis-Gkaragkounis, M. Yannakakis, and X. Zhang, "Smoothed complexity of local max-cut and binary max-csp," *arXiv preprint arXiv:1911.10381*, 2019.

[108] A. Bibak, C. Carlson, and K. Chandrasekaran, "Improving the smoothed complexity of flip for max cut problems," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 2019, pp. 897–916.

[109] H. Röglin, "The complexity of nash equilibria, local optima, and pareto-optimal solutions," *Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule*, 2008.

[110] A. Haken and M. Luby, "Steepest descent can take exponential time for symmetric connection networks," *Complex Systems*, vol. 2, no. 2, pp. 191–196, 1988.

[111] X. Chen, X. Deng, and S.-H. Teng, "Computing Nash equilibria: Approximation and smoothed complexity," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 10 2006, pp. 603–612.

[112] S. Durand and B. Gaujal, "Complexity and optimality of the best response algorithm in random potential games," in *Algorithmic Game Theory*, M. Gairing and R. Savani, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 40–51.

[113] C. Guo, Z. Huang, Z. G. Tang, and X. Zhang, "Generalizing complex hypotheses on product distributions: Auctions, prophet inequalities, and pandora's problem," *CoRR*, vol. abs/1911.11936, 2019.

[114] H. Beyhaghi and R. Kleinberg, "Pandora's problem with nonobligatory inspection," in *EC*. ACM, 2019, pp. 131–132.

[115] S. Singla, "The price of information in combinatorial optimization," in *SODA*. SIAM, 2018, pp. 2523–2532.

[116] S. Singla, "Combinatorial optimization under uncertainty: Probing and stopping-time algorithms," Ph.D. dissertation, PhD thesis, Carnegie Mellon University, 2018.

[117] J. M. Kleinberg and R. Kleinberg, "Delegated search approximates efficient search," in *EC*. ACM, 2018, pp. 287–302.

[118] H. Esfandiari, M. T. Hajiaghayi, B. Lucier, and M. Mitzenmacher, "Online pandora's boxes and bandits," in *AAAI*. AAAI Press, 2019, pp. 1885–1892.

[119] J. B. Kadane, H. A. Simon et al., "Optimal strategies for a class of constrained sequential problems," *The Annals of Statistics*, vol. 5, no. 2, pp. 237–255, 1977.

[120] J. B. Kadane, "Quiz show problems," *Journal of Mathematical Analysis and Applications*, vol. 27, no. 3, pp. 609–623, 1969.

[121] R. Weber et al., "On the gittins index for multiarmed bandits," *The Ann. Appl. Probab.*, vol. 2, no. 4, pp. 1024–1033, 1992.

[122] E. Frostig and G. Weiss, "Four proofs of gittins' multiarmed bandit theorem," *Annals OR*, vol. 241, no. 1-2, pp. 127–165, 2016.

[123] I. Dumitriu, P. Tetali, and P. Winkler, "On playing golf with two balls," *SIAM J. Discret. Math.*, vol. 16, no. 4, pp. 604–615, 2003.

[124] G. Weiss, "Branching bandit processes," *Probab. in the Eng. and Info. Sciences*, vol. 2, no. 3, pp. 269–278, 1988.

[125] G. Keller and A. Oldale, "Branching bandits: a sequential search process with correlated pay-offs," *J. Econ. Theory*, vol. 113, no. 2, pp. 302–315, 2003.

[126] R. D. Kleinberg, B. Waggoner, and E. G. Weyl, "Descending price optimally coordinates search," in *EC*. ACM, 2016, pp. 23–24.

[127] W. Olszewski and R. Weber, "A more general pandora rule?" *J. Econ. Theory*, vol. 160, pp. 429–437, 2015.

[128] L. Doval, "Whether or not to open pandora's box," *J. Econ. Theory*, vol. 175, pp. 127–158, 2018.

[129] A. Gupta, V. Nagarajan, and S. Singla, "Adaptivity gaps for stochastic probing: Submodular and XOS functions," in *SODA*. SIAM, 2017, pp. 1688–1702.

[130] D. Bradac, S. Singla, and G. Zuzic, "(near) optimal adaptivity gaps for stochastic multi-value probing," in *APPROX-RANDOM*, ser. LIPIcs, vol. 145. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 49:1–49:21.

[131] H. Fu, J. Li, and P. Xu, "A PTAS for a class of stochastic dynamic programs," in *ICALP*, ser. LIPIcs, vol. 107. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 56:1–56:14.

[132] D. Segev and S. Singla, "Efficient approximation schemes for stochastic probing and prophet problems," *CoRR*, vol. abs/2007.13121, 2020.

[133] S. Chawla, E. Gergatsouli, Y. Teng, C. Tzamos, and R. Zhang, "Pandora's box with correlations: Learning and approximation," *To appear in FOCS 2020. Preprint on arXiv*, vol. abs/1911.01632, 2019.

[134] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. Wiley, 1994.

[135] A. Anagnostopoulos, I. R. Cohen, S. Leonardi, and J. Lacki, "Stochastic graph exploration," in *ICALP*, ser. LIPIcs, vol. 132. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 136:1–136:14.

[136] M. Feldman, "Guess free maximization of submodular and linear sums," in *WADS*, ser. Lecture Notes in Computer Science, vol. 11646. Springer, 2019, pp. 380–394.

[137] C. Harshaw, M. Feldman, J. Ward, and A. Karbasi, "Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 2634–2643.

[138] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," *Math. Oper. Res.*, vol. 42, no. 4, pp. 1197–1218, 2017.

[139] A. E. F. Clementi and L. Trevisan, "Improved non-approximability results for minimum vertex cover with density constraints," *Theor. Comput. Sci.*, vol. 225, no. 1-2, pp. 113–128, 1999.

[140] C. H. Papadimitriou and M. Yannakakis, “Optimization, approximation, and complexity classes,” *J. Comput. Syst. Sci.*, vol. 43, no. 3, pp. 425–440, 1991.

[141] K. G. Jamieson, R. Nowak, and B. Recht, “Query complexity of derivative-free optimization,” in *Advances in Neural Information Processing Systems*, 2012, pp. 2672–2680.

[142] L. Qian, J. Gao, and H. Jagadish, “Learning user preferences by adaptive pairwise comparison,” *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1322–1333, 2015.

[143] R. Herbrich, “Large margin rank boundaries for ordinal regression,” in *Advances in large margin classifiers.* The MIT Press, 2000, pp. 115–132.

[144] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133–142.

[145] K. G. Jamieson and R. Nowak, “Active ranking using pairwise comparisons,” in *Advances in Neural Information Processing Systems*, 2011, pp. 2240–2248.

[146] D. M. Kane, S. Lovett, S. Moran, and J. Zhang, “Active classification with comparison queries,” in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE, 2017, pp. 355–366.

[147] B. Settles, “Active learning literature survey,” University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[148] C. Cortes and M. Mohri, “*AUC* optimization vs. error rate minimization,” in *NIPS*, 2004, pp. 313–320.

[149] H. Valizadegan, R. Jin, R. Zhang, and J. Mao, “Learning to rank by optimizing *NDCG* measure,” in *NIPS*, 2009, pp. 1883–1891.

[150] C. Drummond and R. C. Holte, “Severe class imbalance: Why better algorithms aren’t the answer,” in *ECML.* Springer, 2005, pp. 539–546.

[151] I. Steinwart, “How to compare different loss functions and their risks,” *Constructive Approximation*, vol. 26, no. 2, pp. 225–287, 2007.

[152] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[153] P. A. Samuelson, “A note on the pure theory of consumer’s behaviour,” *Economica*, vol. 5, no. 17, pp. 61–71, 1938.

[154] A. Mas-Colell, “The recoverability of consumers’ preferences from market demand behavior,” *Econometrica: Journal of the Econometric Society*, pp. 1409–1430, 1977.

[155] H. R. Varian, “Revealed preference. in samuelsonian economics and the 21st century by m. szenberg and l. ramrattand and a. a. gottesman (editors),” in *In Samuelsonian Economics and the 21st Century*, M. Szenberg, L. Ramrattand, and A. A. Gottesman, Eds. Oxford University Press, 2005, pp. 99–115.

[156] D. Braziunas and C. Boutilier, “Minimax regret based elicitation of generalized additive utilities.” in *UAI*, 2007, pp. 25–32.

[157] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, “Consistent binary classification with generalized performance metrics,” in *NIPS*, 2014, pp. 2744–2752.

[158] H. Narasimhan, H. Ramaswamy, A. Saha, and S. Agarwal, “Consistent multiclass algorithms for complex performance measures,” in *ICML*, 2015, pp. 2398–2407.

[159] B. Qian, X. Wang, F. Wang, H. Li, J. Ye, and I. Davidson, “Active learning from relative queries.” in *IJCAI*, 2013, pp. 1614–1620.

[160] G. Tamburrelli and A. Margara, “Towards automated *A/B* testing,” in *International Symposium on Search Based Software Engineering.* Springer, 2014, pp. 184–198.

[161] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *ACM SIGKDD.* ACM, 2016, pp. 1135–1144.

[162] F. Doshi-Velez and B. Kim, “Towards A Rigorous Science of Interpretable Machine Learning,” *ArXiv e-prints:1702.08608*, 2017.

[163] N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu, “Optimizing f-measure: A tale of two approaches,” *CoRR*, vol. abs/1206.4625, 2012.

[164] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[165] T. H. Cormen, *Introduction to algorithms.* MIT press, 2009.

[166] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence.* Oxford university press, 2013.

[167] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” in *The Collected Works of Wassily Hoeffding.* Springer, 1994, pp. 409–426.

[168] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming (Wiley Series in Probability and Statistics).* Wiley-Interscience, 2005.

[169] M. Bagnoli and T. Bergstrom, “Log-concave probability and its applications,” *Economic theory*, vol. 26, no. 2, pp. 445–469, 2005.

[170] P. Wolfe, “Finding the nearest point in a polytope,” *Mathematical Programming*, vol. 11, no. 1, pp. 128–149, 1976.

[171] A. Gotovos, A. Karbasi, and A. Krause, "Non-monotone adaptive submodular maximization," in *IJCAI*. AAAI Press, 2015, pp. 1996–2003.

[172] G. Amanatidis, F. Fusco, P. Lazos, S. Leonardi, and R. Reiffenhäuser, "Fast adaptive non-monotone submodular maximization subject to a knapsack constraint," *CoRR*, vol. abs/2007.05014, 2020.