A Learning Perspective on Selfish Behavior in Games

Katrina Ligett

CMU-CS-09-149
July 31, 2009

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Avrim Blum (chair)
Anupam Gupta
R. Ravi
Eva Tardos, Cornell University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant numbers DGE-0750271, CCF-0514922, IIS-0121678, and through a generous gift from the Microsoft Corporation. This work was also supported in part by an AT&T Labs Graduate Fellowship.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: algorithmic game theory, online algorithms, regret minimization

Abstract

Computer systems increasingly involve the interaction of multiple self-interested agents. The designers of these systems have objectives they wish to optimize, but by allowing selfish agents to interact in the system, they lose the ability to directly control behavior. What is lost by this lack of centralized control? What are the likely outcomes of selfish behavior?

In this work, we consider learning dynamics as a tool for better classifying and understanding outcomes of selfish behavior in games. In particular, when such learning algorithms exist and are efficient, we propose “regret-minimization” as a criterion for self-interested behavior and study the system-wide effects in broad classes of games when players achieve this criterion. In addition, we present a general transformation from offline approximation algorithms for linear optimization problems to online algorithms that achieve low regret.

Acknowledgments

As a new graduate student attending conferences, I would introduce myself to senior members of the field and mention my advisor's name. Time and time again, they would congratulate me on my good fortune in having Avrim Blum as an advisor. More than one said that if they could choose any advisor in the world for themselves, they would choose Avrim. They were right, of course. Avrim is everything an advisor should be—challenging, thoughtful, encouraging, and inspiring.

Thank you to the members of my thesis committee, Eva, Anupam, and Ravi. I'm very grateful for their thoughtful comments, questions, and suggestions. The work in this thesis is joint with an amazing set of colleagues: Avrim Blum, Eyal Even-Dar, Mohammad Taghi Hajiaghayi, Sham Kakade, Adam Kalai, and Aaron Roth. It has been such a pleasure working with all of them, and the work here incorporates their many insights. David Johnson has been a wonderful co-author, mentor, and source of wisdom. Adam Kalai and Steve Chien each hosted me for delightful and productive summers, and continue to be two of my favorite people to talk with about research. My fellow grad student, Aaron Roth, has been the ideal collaborator, through many, many hours of research conversations. I also owe much to my entire set of co-authors and collaborators, to the theory group at Carnegie Mellon, and to colleagues at institutions that have hosted me for visits and internships—AT&T Labs Research, Toyota Technological Institute in Chicago, the Max Planck Institut, and Microsoft Research.

Friends, you've made these past five years great. You joined me for sports, music, cooking, and other adventures. You watered my plants and fed my cats when I was out of town. You called (and emailed, Google chatted, texted, Skyped, and showed up unannounced). You listened, challenged, inspired, mentored, and gave advice. Thank you.

My mother, father, and brother have always encouraged me, cheered for me, supported me, and made me laugh. Dan has been my partner in everything, the whole way through.

Thank you all.

Contents

1 Introduction **1**

1.1 Overview of the thesis 2

1.1.1 Approximate Online Linear Optimization 2

1.1.2 Regret-Minimization as a Definition of Selfish Behavior in Games 3

2 Background, Definitions, and Related Work **5**

2.1 Background: Game theory 5

2.1.1 Algorithmic Game Theory 7

2.2 Background: Learning in games 8

2.2.1 Evolutionary dynamics 9

2.2.2 Bayesian learning 10

2.2.3 Further results on adaptive learning 11

2.2.4 Regret 11

2.3 Subsequent work on regret minimization in games 16

2.4 Work not in this thesis 16

3 Approximate Online Linear Optimization **19**

3.1 Introduction 19

3.1.1 Hannan's approach 20

3.1.2 Example where "follow-the-perturbed-leader" fails 21

3.1.3 Informal statement of results 22

3.2 Formal definitions 24

3.3 Full-information algorithm 25

3.3.1 Approximate Projection 27

3.3.2 Constructing the Algorithm 29

3.3.3 The approximate projection algorithm 31

3.3.4 Analysis 32

3.4 Bandit algorithm 33

3.4.1 Difficulty of the black-box reduction 36

3.5 Conclusions 37

4 Convergence to Nash Equilibria in Routing Games **39**

4.1 Introduction 39

4.1.1 Regret and Nash equilibria 40

4.2 Preliminaries 40

4.2.1 Nonatomic congestion games 40

4.2.2 Approaching Nash Equilibria 42

4.2.3 Dependence on slope 42

4.3 Infinitesimal Users: Linear Latency Functions 43

4.4 Infinitesimal Users: General Latency Functions 43

4.5 Infinitesimal Users: Bounds on Most Timesteps 46

4.6 Regret Minimization and the Price of Anarchy 47

4.7 Discrete Users: Parallel Paths 50

4.8 Conclusions 51

5 The Price of Total Anarchy 53

5.1 Introduction 53

5.1.1 Our results 54

5.2 Preliminaries 55

5.3 Hotelling games 55

5.3.1 Definition and price of anarchy 56

5.3.2 Price of total anarchy 56

5.3.3 The price of total anarchy in generalized Hotelling games 58

5.3.4 Regret minimization need not converge 60

5.4 Valid games 61

5.4.1 Definitions and price of anarchy 61

5.4.2 Price of total anarchy 62

5.5 Atomic Congestion Games 65

5.5.1 Atomic congestion games with sum social utility 66

5.5.2 Parallel link congestion game with makespan social utility 68

5.5.3 Parallel links congestion game with sum social utility 71

5.6 Algorithmic efficiency 73

5.7 Conclusions 73

6 Conclusions and Directions for Future Work 75

Chapter 1

Introduction

Computer systems increasingly involve the interaction of multiple self-interested agents. The designers of these systems have objectives they wish to optimize, but by allowing selfish agents to interact in the system, they lose the ability to directly control behavior. What is lost by this lack of centralized control? What are the likely outcomes of selfish behavior?

Economists have long studied games with self-interested players, and game theoretic equilibrium concepts have recently received attention from computer scientists for their potential to model the complex systems that arise in our modern computing environment. A *Nash equilibrium* in such a game is a profile of strategies for each player such that, given the strategies of the other players, no player prefers to deviate from her strategy in the profile. In much of the work in algorithmic game theory, Nash equilibrium strategies have been used as a tool for studying selfish behavior; for example, the *price of anarchy* [87] of a game is defined to be the ratio of the value of the social welfare in the worst Nash equilibrium to the social optimum value.

It may not be realistic, however, to assume that all agents in a system will necessarily play strategies that form a Nash equilibrium. Even with centralized control, Nash equilibria can be computationally difficult (PPAD-hard) to find [23]. Moreover, even when Nash equilibria are easy to find computationally, it seems unreasonably optimistic to assume that distributed self-interested agents, often with limited information about the overall state of the system, will necessarily converge to Nash. One would therefore like a different, natural, computationally meaningful model of agent behavior that allows better understanding of overall system behavior.

In this dissertation, I propose that in games where adaptive learning algorithms give good guarantees on individual performance, such guarantees can be seen as a minimal criterion for selfish behavior. I support this claim by presenting novel work on adaptive learning algorithms, and by analyzing the global consequences in broad classes of games when selfish agents use such algorithms.

1.1 Overview of the thesis

1.1.1 Approximate Online Linear Optimization

In the 1950's, Hannan gave an algorithm for playing repeated two-player games against an arbitrary opponent [69]. His was one of the earliest algorithms with the *no-regret* property: against any opponent, his algorithm achieved expected performance asymptotically near that of the best single action, where the best is chosen with the benefit of hindsight. Put another way, after sufficiently many rounds, someone using his algorithm would not benefit (significantly) by being able to change his actions to any single action, even if this action could be chosen after observing the opponent's play. An algorithm is called regret-minimizing, or no-regret, if the expected regret it incurs goes to zero as a function of time. There is a rich literature from machine learning and game theory on adaptive *no-regret algorithms* [4, 35, 59, 69, 80, 81, 85, 91, 92, 122].

Kalai and Vempala showed that Hannan's approach can be used to *efficiently* solve online linear optimization problems as well [81]. Hannan's algorithm relied on the ability to find best responses to an opponent's play history. Informally speaking, Kalai and Vempala [81] replaced this best-reply computation with an efficient black-box optimization algorithm. However, the above approach breaks down when one can only approximately solve the offline optimization problem efficiently or one can only compute approximate best responses.

In an online linear optimization problem, on each period t, an online algorithm chooses $s_t \in \mathcal{S}$ from a fixed (possibly infinite) set $\mathcal{S}$ of feasible decisions. Nature (who may be adversarial) chooses a weight vector $w_t \in \mathbb{R}^n$, and the algorithm incurs cost $c(s_t, w_t)$, where c is a fixed cost function that is linear in the weight vector. In the *full-information* setting, the vector w_t is then revealed to the algorithm, and in the *bandit* setting, only the cost experienced, $c(s_t, w_t)$, is revealed. The goal of the online algorithm is to perform nearly as well as the best fixed $s \in \mathcal{S}$ in hindsight. Many repeated decision-making problems with weights fit naturally into this framework, such as online shortest-path, online TSP, online clustering, and online weighted set cover.

Previously, it was shown how to convert any efficient *exact* offline optimization algorithm for such a problem into an efficient online algorithm in both the full-information and the bandit settings, with average cost nearly as good as that of the best fixed $s \in \mathcal{S}$ in hindsight. However, in the case where the offline algorithm is an approximation algorithm with ratio $\alpha > 1$, the previous approach only worked for special types of approximation algorithms.

In Chapter 3, based on joint work with Sham Kakade and Adam Kalai that appeared in STOC 2007 [80], I show how to convert *any* offline approximation algorithm for a linear optimization problem into a corresponding online approximation algorithm, with a polynomial blowup in runtime. If the offline algorithm has an α-approximation guarantee, then the expected cost of the online algorithm on any sequence is not much larger than α times that of the best $s \in \mathcal{S}$, where the best is chosen with the benefit of hindsight. Our new approach is inspired by Zinkevich's algorithm for the problem of minimizing convex functions over a convex feasible set $\mathcal{S} \subseteq \mathbb{R}^n$ [122]. However, the application is not direct and requires a geometric transformation that can be applied to any approximation algorithm.

The algorithm can also be viewed as a method for playing large repeated games, where one can only compute *approximate* best-responses, rather than best-responses.

1.1.2 Regret-Minimization as a Definition of Selfish Behavior in Games

In this thesis, we propose regret-minimization as a minimal criterion for selfish behavior and study the consequences of individual regret-minimizing guarantees for the system as a whole. No-regret algorithms are very compelling from the point of view of individuals: if a person uses a no-regret algorithm in choosing which route to take to work each day, she will get a good guarantee on her commute time no matter what is causing congestion (other drivers, road construction, or unpredictable events). We consider repeated play of the game and allow agents to play any sequence of actions with only the assumption that this action sequence has low regret with respect to the best fixed action in hindsight. Regret minimization is a realistic assumption because there exist a number of efficient algorithms for playing a wide variety of games that guarantee regret that tends to zero, because it requires only localized information, and because in a game with many players in which the actions of any single player do not greatly affect the decisions of other players (as is often studied in the network setting), players can only improve their situation by switching from a strategy with high regret to a strategy with low regret. Regret minimization can be done via simple, efficient algorithms even in many settings where the number of action choices for each player is exponential in the natural parameters of the problem.

In Chapter 4, based on joint work with Avrim Blum and Eyal Even-Dar that appeared at PODC 2006 [15], we apply regret-minimizing algorithms to the well-studied Wardrop setting for multicommodity flow and infinitesimal agents, which models traffic on a network where the cost of an edge is a function of the amount of traffic using that edge [30, 33, 47, 87, 108]. We show that flows comprised of regret-minimizing players will approach Nash equilibria in the sense that, over time, a $1 - \epsilon$ fraction of the daily flows will have the property that at most an ϵ fraction of the agents in them have more than an ϵ incentive to deviate from their chosen path, where ϵ approaches 0 at a rate that depends polynomially on (1) the size of the graph, (2) the regret-bounds of the algorithms, and (3) the maximum slope of any latency function. Our results imply that in the Wardrop routing model, so long as edge latencies have bounded slope, we can view Nash equilibria as not just a stable steady-state or the result of adaptive procedures specifically designed to find them, but in fact as the inevitable result of individual selfishly adaptive behavior by agents that do not necessarily know (or care) what policies other agents are using.

Even in games where regret-minimizing players may not approach a Nash equilibrium, it may be possible to analyze the the social cost of regret minimization (as opposed to centralized control) directly. In Chapter 5, based on joint work with Avrim Blum, Mohammad Taghi Hajiaghayi, and Aaron Roth that appeared at STOC 2008 [16], we propose weakening the assumption made when studying the price of anarchy: Rather than assume that self-interested players will play according to a Nash equilibrium (which may even be computationally hard to find), we assume only that selfish players play so as to minimize their own regret. We prove that despite our weakened assumptions, in several broad classes of games, this “price of total anarchy” matches the Nash price of anarchy, even though play may never converge to Nash equilibrium.

This *price of total anarchy* is strictly a generalization of price of anarchy, since in a Nash equilibrium, all players have zero regret. In this chapter, we consider generalized Hotelling games [75], in which players compete with each other for market share; valid games [118] (a broad class of games that includes among others facility location, market sharing [65], traffic routing, and multiple-item auctions); linear congestion games with atomic players and unsplittable flow

[6, 25]; and parallel link congestion games [87]. We prove that in the first three cases, the price of total anarchy matches the price of anarchy exactly even if the play itself is not approaching equilibrium; for parallel link congestion and makespan social cost, we get an exact match for $n = 2$ links but an exponentially greater price for general n, highlighting a natural setting where these concepts differ.

In contrast to the price of anarchy and the recently introduced price of sinking [63], which require all players to behave in a prescribed manner, we show that the price of total anarchy is in many cases resilient to the presence of Byzantine players, about whom we make no assumptions. Finally, because the price of total anarchy is an upper bound on the price of anarchy even in mixed strategies, for some games our results yield as corollaries previously unknown bounds on the price of anarchy in mixed strategies.

Chapter 2

Background, Definitions, and Related Work

Traditional approaches to system design assume a single, centralized administrator who assumes perfect compliance with her proposed protocols. Modern, networked systems, however, bring all of these assumptions into question: Large-scale, distributed systems and protocols run by autonomous agents raise issues of selfish incentives and private information. Automation makes it easy for agents to adapt their actions in response to the system or to actions of others, but any individual agent may have a very limited understanding of the game she is playing. The huge amounts of money at stake bring economic incentives to the fore. Issues of errors, collusion, and malicious activity all threaten to undermine the quality and stability of outcomes.

Learning protocols and game theory are thus natural tools for the analysis of large-scale networked systems. In this chapter, we introduce the relevant terms, tools, and related work from game theory and learning theory that underlie this thesis.

2.1 Background: Game theory

In this thesis, we consider strategic games in which all players in a game act simultaneously and without knowledge of other players' actions. Economists have long studied games with self-interested players. A *Nash equilibrium* in such a game is a profile of strategies for each player such that, given the strategies of the other players, no player prefers to deviate from her strategy in the profile. A Nash equilibrium can be *pure* or *mixed*, depending on whether the players all play pure, deterministic strategies, or they randomize over pure strategies to give a mixed strategy. While not every game has a pure Nash equilibrium, every game has at least one mixed Nash equilibrium. There are a wide variety of proofs of this; many, including the original proof by Nash [101] are applications of fixed point theorems. A closely related concept, a *correlated equilibrium* is a probability distribution over players' joint actions that is enforceable by an external signal: if players were assigned to actions and knew their assignments came from this joint distribution, based on their resulting expected value, they would have no incentive to deviate and play a different action.

In Chapter 4, we consider games with a particular measure of infinitesimal players, each with one of k player types; in Chapter 5, we consider k-player games. For each player (or player type) i, we denote by $\mathcal{P}_i$ the set of pure strategies available to that player. A mixed strategy is a probability distribution over actions in $\mathcal{P}_i$; we denote by $\mathcal{S}_i$ the set of mixed strategies available to a player (of type) i. Every game has an associated social utility function γ that takes a set containing a pure action for each player to some real value. Each player (type) i has an individual utility function $\alpha_i : \mathcal{P} \to \mathbb{R}$.

We often want to talk about the social or individual utility of a strategy profile that assigns each player a mixed action. To this end, we denote by $\bar{\gamma}$ the expected social utility over the randomness of the players (this is equivalent to γ in the case of infinitesimal players) and by $\bar{\alpha}$ the expected value of the utility of a strategy profile to player i. We denote the social value of the socially optimum strategy profile by $\mathbf{OPT}$.

One can define a class of games by restricting the generality of the social utility functions or the individual utility functions, or both. Two general classes of games, potential games and congestion games, appear frequently in the literature; we introduce them here.

Potential games (definition from Monderer and Shapley [99]) A function $\Phi : S \to \mathbb{R}$ is called an ordinal potential function for the game G if for all i and all $s_{-i} \in S_{-i}$,

$$\alpha_i(x, s_{-i}) - \alpha_i(z, s_{-i}) > 0 \text{ iff } \Phi(x, s_{-i}) - \Phi(z, s_{-i}) > 0, \text{ for all } x, z \in S_i.$$

A function $\Phi : S \to \mathbb{R}$ is called a potential function for the game G if for all i and all $s_{-i} \in S_{-i}$,

$$\alpha_i(x, s_{-i}) - \alpha_i(z, s_{-i}) = \Phi(x, s_{-i}) - \Phi(z, s_{-i}), \text{ for all } x, z \in S_i.$$

G is called an ordinal (exact) potential game if it admits an ordinal (exact) potential. In an ordinal potential game, a global maximum of the potential function is a pure Nash equilibrium (there may be other pure Nash equilibria, which are local maxima).

Potential functions, when they exist, give us a path from any game state to an equilibrium state, but the length of the path can be exponential, and each step of the path involves a change in only one player's actions. Thus, potential functions do not immediately imply efficient, distributed algorithms for equilibrium computation.

Congestion games A congestion game is a game with k players and m resources, where the strategies available to each player are subsets of the resources. For any player using a resource j, the cost of that resource depends only on the total number of players who are using that resource (not on their identities); a player's total cost is the total cost of the resources she selects.

Every congestion game has an exact potential function, and Rosenthal [107] shows that every congestion game has a pure Nash equilibrium, using a potential function argument; thus, any sequence of better response moves in a congestion game eventually reaches a pure Nash equilibrium, but this sequence can be exponentially long.

Congestion games are sometimes referred to as “atomic congestion games”, and the similar class of games involving a continuum of infinitesimal players are referred to as “nonatomic congestion games.” One may also consider a weighted variant of congestion games, where each player has a weight, and the cost of a resource is a monotone function of the total weight of players selecting that resource. Milchtaich [95] considers a subclass of weighted congestion games similar to the weighted load balancing games we treat in Chapter 5, and shows that while the games in this class are not potential games, they still possess pure Nash equilibria. A sequence of potential function improvements converges in polynomial time to a pure equilibrium in symmetric network congestion games.

2.1.1 Algorithmic Game Theory

The field of Algorithmic Game Theory seeks to apply algorithmic and computational tools and perspectives to game theoretic problems. Productive lines of work in this area include studying

- the social quality of outcomes under selfish behavior in games,
- the centralized computability of equilibria, and
- implications of computational restrictions on the agents and other entities involved in a game (for example, on the buyers or on the auctioneer in an auction setting; this can sometimes be seen as a question of efficient distributed computation).

The work in this thesis touches on all three of these issues. Inspired in part by negative results on equilibrium computation, we propose a shift away from static equilibria as a definition of selfishness. We instead study the social consequences when we make relatively weak assumptions about the player actions; these assumptions on the players are computationally achievable in broad classes of games, even some games with an exponential number of pure strategies for each player. In this section, we briefly survey the relevant results in the algorithmic game theory literature.

Outcomes of selfish behavior

In 1999, Koutsoupias and Papadimitriou [87] introduced the notion of the *price of anarchy* as a measure of the effects of selfish behavior: they studied the ratio between the social welfare of the optimum solution and that of the worst Nash equilibrium. Many subsequent results have studied the price of anarchy in a wide range of computational problems from job scheduling to facility location to network creation games, and especially to problems of routing in the Wardrop model, where the cost of an edge is a function of the amount of traffic using that edge [29, 32, 47, 87, 108]. Such work implicitly assumes that selfish individual behavior results in Nash equilibria.

We consider both maximization and minimization games in this thesis. In *maximization* games the goal is to *maximize* the social utility function and the players wish to *maximize* their individual utility functions; in *minimization* games, both quantities minimized. We define the price of anarchy so that its value is always greater than or equal to one, regardless of whether we are discussing a maximization or a minimization game:

Definition 2.1.1. *The price of anarchy for an instance of a maximization game is defined to be* $\frac{\mathbf{OPT}}{\bar{\gamma}(S)}$*, where* S *is the worst Nash equilibrium for the game (the equilibrium that maximizes the price of anarchy). The price of anarchy for an instance of a minimization game is defined to be* $\frac{\bar{\gamma}(S)}{\mathbf{OPT}}$*, where* S *is the worst Nash equilibrium for the game (the equilibrium that maximizes the price of anarchy).*

In this thesis, we propose an alternative tool for understanding the consequences of selfish behavior in games that avoids some of the computational difficulties associated with studying Nash equilibria.

Computational issues

Computational issues call into question the suitability of Nash equilibria as a definition of selfish behavior: it seems unreasonable to expect that selfish, independent agents will be able to compute equilibria in a distributed fashion if they cannot even be computed centrally. And, in fact, this is the case: In 2-player, n-action games, Nash equilibria are PPAD-hard to compute [23].

Goldberg and Papadimitriou [67] first reduced the problem of finding a Nash equilibrium in k-player games to the 4-player case; Daskalakis, Goldberg, and Papadimitriou [38] then showed that the 4-player case is PPAD-complete. Independently, manuscripts by Chen and Deng [21] and Daskalakis and Papadimitriou [36] then demonstrated PPAD-hardness for three players, followed by the Chen-Deng hardness result [23] for two players. Chen, Deng, and Teng [22] rule out the possibility of a FPTAS for finding even an approximate Nash equilibrium. In addition, Hart and Mansour [71] present a communication complexity result that amounts to showing hardness of the distributed computation of Nash equilibria.

Finding pure equilibria in congestion games is PLS-complete [48], even with linear latency functions [2]. Further, Skopalik and Vocking [114] show that finding even an ϵ-approximate equilibrium is PLS-complete, and reaching approximate equilibrium by ϵ-improving steps from a given initial state is PSPACE-complete.

Positive results for Nash equilibrium computation are scarce. Equilibria in zero-sum games (where the sum of the players' utilities is always zero) can be computed efficiently by linear programming. Anonymous games, where only the counts (but not the identities) of players playing each strategy affect each agent's utility, have a PTAS, due to Daskalakis and Papadimitriou [37]. Of the classes of games discussed in this thesis, anonymous games encompass the class of congestion games, but not the classes of generalized Hotelling games nor the class of valid games. By contrast, correlated equilibria can be computed efficiently in a wide variety of succinctly representable games [103].

2.2 Background: Learning in games

The field of game theory has been developed primarily for the study of small-scale, sophisticated interactions with large amounts of information available. Typical assumptions include common knowledge and common priors. Thus, the traditional approach is to study Nash equilibria, under the assumption that sophisticated players with full information about the game calculate a Nash equilibrium and play it, assuming their opponents will act similarly. However, these assumptions are not always a good fit for dynamic, distributed interactions involving partial information. In addition, this approach fails to address issues of computational complexity and of selection among equilibria. Instead of studying static notions of equilibria, one can use learning dynamics as a tool for understanding complex, distributed games with selfish players.

There are three main lines of research involving learning dynamics and games. Research on *evolutionary dynamics* studies the fitness of strategies or outcomes against opponents and in the face of random mutations. In *Bayesian learning*, each agent maintains a set of beliefs about the state of the game or about her opponents, and updates and plays optimally according to this information at every time step. In addition, there are a variety of *adaptive learning* techniques that do not model the opponent directly; these include regret testing and regret minimization policies.

Much of the previous work on learning dynamics in games seeks to show convergence-type results to static notions of equilibria. As such, this work can be seen as an attempt to justify the study of Nash equilibria under less controlled conditions. Although we do present convergence results in this thesis (in Chapter 4), one of our contributions is in proposing that a certain class of individual welfare guarantees (obtainable by adaptive learning algorithms) is in itself a good definition of selfish behavior; we study the social costs of such behavior *even when the behavior does not converge to equilibrium*.

As the main focus of this thesis is on a particular class of adaptive learning algorithms, we briefly present the work on games from Bayesian and evolutionary game theoretic perspectives, and then present the adaptive learning literature in more depth.

2.2.1 Evolutionary dynamics

Biological evolution is one obvious model of selfish individual adaption in a complex environment. Evolutionary and evolution-inspired approaches have found their way into the game theory literature in the study of

- agents who “evolve” their strategies over time using updates inspired in some way by evolution,
- strategies that are “stable” under an evolution-based notion of stability, and
- game states that are “stable” under a stability notion derived from evolutionary techniques.

For a survey of algorithmic results that have employed or studied other evolutionary game theory techniques and concepts, see Suri [116]; we summarize a few of the results here.

In the first category, Fischer and Vöcking [50] show that under replicator dynamics in the routing game studied by Roughgarden and Tardos [108], players converge to Nash equilibria. Fisher et al. [52] went on to show that using a simultaneous adaptive sampling method, play converges quickly to a Nash equilibrium. Sandholm [111] considers convergence in potential games (which include routing games), and shows that a very broad class of evolutionary dynamics is guaranteed to converge to Nash equilibrium. Note that such dynamics do not include general no-regret dynamics.

An *evolutionarily stable strategy (ESS)* is a strategy that, if adopted by a population of players, cannot be invaded by any alternative strategy that does not initially have significant representation in the population. ESS are a refinement of Nash equilibria, and so do not always exist, and are not necessarily associated with a natural play dynamic. In addition, ESS are resilient only to single shocks, whereas stochastically stable states are resilient to persistent noise.

The evolutionary game theory literature on *stochastic stability* studies repeated games where on each round, each player observes her action and its outcome, and then uses simple rules to select her action for the next round based only on her size-restricted memory of the past rounds. In any round, players have a small probability of deviating from their prescribed decision rules. The state of the game is the contents of the memories of all the players. The *stochastically stable states* in such a game are the states with non-zero probability in the limit of this random process, as the probability of error approaches zero. Stochastic stability and its adaptive learning model were first defined by Foster and Young [54]. Stochastic stability and has been widely studied in the economics literature (see, for example, [17, 43, 79, 83, 90, 106, 119]). In contrast with the standard game theory solution concept of evolutionarily stable strategies (ESS), a game always has stochastically stable states that result (by construction) from natural dynamics. In joint work not presented in this thesis but discussed briefly in Section 2.4, we initiate the study of the social utility of stochastically stable states.

2.2.2 Bayesian learning

In a Bayesian (or, as characterized by Young [121], *model-based*) learning framework, each player is assumed to have subjective beliefs about her opponents' strategies, and then uses these beliefs to compute her optimal strategy. After each time step in a repeated game, players receive information about their payoffs and potentially also about the actions taken by their opponents, and use this information to update their beliefs in a Bayesian fashion [70, 89]. Each player is assumed to have and be aware of her own discount factor on future earnings, and each player's objective is to maximize her long-term expected discounted payoff, relative to her beliefs.

In this setting, when agents have perfect monitoring (every player is informed of the entire history of play of all of his opponents), Kalai and Lehrer [82] show that there exist update procedures that converge in finite time to arbitrarily good approximations to a Nash equilibrium of the repeated game, provided that players' strategies are optimal given their beliefs and that their beliefs put nonzero probability mass on every event that has positive probability under their strategies.

Fictitious play [18] is another well-studied example of a model-based learning setting. In fictitious play, each player observes the empirical frequency distribution over opponent play and chooses her action at each timestep to maximize her expected payoff under that distribution. The choice is myopic, in that each player seeks to maximize payoff for the next day only, without concern for future payoffs. One advantage of this approach is that each player does not need to know her opponents' utility functions. However, playing in this manner when opponents are behaving arbitrarily doesn't provide any guarantees to the individual. Also, the model assumes perfect monitoring. Fictitious play converges to Nash equilibrium in zero-sum two-person games [105], in potential games [100] and in two-person two-strategy games [98], but not in two-person three-strategy games [113].

Foster and Young [56] show that there exist no general, model-based procedures that always converge to Nash equilibria of the repeated game when the players are perfectly rational (they play perfectly optimally given their beliefs) and the unknown opponent payoffs are distributed over some continuous space. If the rationality assumption is relaxed, though, Foster and Young [57] give a simple procedure based on hypothesis testing that results in convergence of the period-by-period play to the set of Nash equilibria of the stage game. But, Foster and Young [57] present a class of uncoupled learning procedures that converge in probability to the set of Nash equilibria in any finite game. The basic idea is to track the empirical distribution on opponent actions in recent history and periodically update so that you are always playing as if against a hypothetical frequency distribution that is consistent with the empirical one.

2.2.3 Further results on adaptive learning

As we have seen above, Bayesian update procedures cannot have good convergence properties in general settings with continuous payoff functions; this has led researchers to explore more general learning dynamics. In this section, we discuss adaptive learning dynamics that do not explicitly model opponent behavior; Young [121] terms this *model-free learning*. Although the literature on learning dynamics is too vast to cover every variation and result in detail, we survey the results here. We emphasize that most of the positive results on learning dynamics convergence presented both in the previous section and in this one do not provide *efficient* convergence. In addition, many require strict adherence of all players to a very specific protocol, and do not give any performance guarantees to the individual agents unless all agents comply with the protocol.

Possibility and impossibility for inefficient dynamics

A learning rule is called uncoupled if the player using it does not condition her strategy on the payoffs of her opponents. A radically uncoupled learning rule is one that does not condition on opponents' past actions or payoffs.

Hart MasColell [73] show that in general, uncoupled dynamics do not lead to period-by-period convergence to an approximate Nash equilibrium of the stage game if the player states are histories of bounded length, even for two-person games. In their work on regret testing, however, Foster and Young and Germano and Lugosi [58, 62] demonstrate a family of radically uncoupled learning rules whose period-by-period behavior comes arbitrarily close to Nash equilibrium, for any finite, two-person game. Regret testing depends only on the players own history of realized payoffs (radically uncoupled). In this model, in every time step, each player has some fixed probability of making a "mistake" (one can think of this as playing uniformly at random). These results circumvent the Hart-MasColell impossibility result by not restricting themselves to bounded length histories.

2.2.4 Regret

The *regret* of a sequence of actions in a repeated game is defined with respect to a particular class of transformations Ψ over the agent's action set; it is defined as the difference between the average cost incurred and the average cost the best transformation would have incurred, where the best is chosen with the benefit of hindsight.

Definition 2.2.1. *The regret of player i in a maximization game given action sets $P^1, P^2, \ldots, P^T$ is*

$$\max_{\{p_i^1, p_i^2, \ldots, p_i^t\} \in \Psi(P_i)} \frac{1}{T} \sum_{t=1}^{T} \alpha_i(P^t \oplus p_i^t) - \frac{1}{T} \sum_{t=1}^{T} \alpha_i(P^t).$$

The regret of player i in a minimization game given action sets $P^1, P^2, \ldots, P^T$ is

$$\frac{1}{T}\sum_{t=1}^{T}\alpha_i(P^t) - \min_{\{p_i^1,p_i^2,\ldots,p_i^t\}\in\Psi(P_i)}\frac{1}{T}\sum_{t=1}^{T}\alpha_i(P^t \oplus p_i^t).$$

An algorithm is called regret-minimizing, or no-regret, if the expected regret it incurs goes to zero as a function of time. A regret-minimizing algorithm is one with low expected regret.

Definition 2.2.2. *When a player i uses a regret-minimizing algorithm or achieves low regret, for any sequence $P^1, \ldots, P^T$, she achieves the property*

$$\max_{\{p_i^1,p_i^2,\ldots,p_i^t\}\in\Psi(P_i)}\frac{1}{T}\sum_{t=1}^{T}\alpha_i(P^t \oplus p_i^t) \leq R(T) + E\left[\frac{1}{T}\sum_{t=1}^{T}\alpha_i(P^t)\right]$$

for maximization games and

$$E\left[\frac{1}{T}\sum_{t=1}^{T}\alpha_i(P^t)\right] \leq R(T) + \min_{\{p_i^1,p_i^2,\ldots,p_i^t\}\in\Psi(P_i)}\frac{1}{T}\sum_{t=1}^{T}\alpha_i(P^t \oplus p_i^t)$$

for minimization games, where expectation is over the internal randomness of the algorithm, and where $R(T) \to 0$ as $T \to \infty$. The function $R(T)$ may depend on the size of the game or a compact representation thereof. We then define T_ϵ to be the number of time steps required to get $R(T) = \epsilon$.

Note that this implies that, for any sequence $S^1, \ldots, S^T$, a player with the regret-minimizing property achieves

$$\max_{\{p_i^1,p_i^2,\ldots,p_i^t\}\in\Psi(P_i)}\frac{1}{T}\sum_{t=1}^{T}\bar{\alpha}_i(S^t \oplus p_i^t) \leq R(T) + \frac{1}{T}\sum_{t=1}^{T}\bar{\alpha}_i(S^t)$$

for *maximization* games and

$$\frac{1}{T}\sum_{t=1}^{T}\bar{\alpha}_i(S^t) \leq R(T) + \min_{\{p_i^1,p_i^2,\ldots,p_i^t\}\in\Psi(P_i)}\frac{1}{T}\sum_{t=1}^{T}\bar{\alpha}_i(S^t \oplus p_i^t)$$

for *minimization* games. See Greenwald, Li, and Marks [68] for examples of generalized *regret matching* regret minimization algorithms.

Internal regret minimization

A variety of adaptive learning algorithms, including the algorithms that are the focus of this thesis, are based on or achieve notions of low regret. One class of action transformations focuses on the question, "on all occasions when you selected a particular action, how good a response was it to the actual actions of the other players?" This class consists of all transformations of action histories that transform all instances of a particular action p into some other action p'. Algorithms that achieve low regret with respect to this set of transformations are said to minimize *internal* *regret*, or to be *universally calibrated*. A weaker criterion is that algorithms may achieve low internal regret when played against themselves, rather than against an arbitrary adversary; such algorithms are said to be *calibrated*, but not universally so.

It is known that certain regret-matching algorithms such as that of Hart and Mas-Colell [72, 74], as well as any algorithms satisfying the stronger property of no internal regret [55], have the property that the empirical distribution of play approaches a *correlated* equilibrium. The algorithms of Hart and MasColell are polynomial time for settings in which action choices are explicitly given. In addition, although Neyman [102] does show that the only correlated equilibrium in atomic congestion games is the unique Nash equilibrium, there is no known efficient implementation for internal regret minimization for routing problems.

External regret minimization

The focus in this thesis is on *external regret* minimization algorithms, where the set Ψ consists of each of the feasible fixed actions. Here the regret is with respect to the best single action over the entire play history.

Definition 2.2.3. *The external regret of player i in a maximization game given action sets $P^1, P^2, \ldots, P^T$ is*

$$\max_{p_i \in \mathcal{P}_i} \frac{1}{T} \sum_{t=1}^{T} R(T) = \alpha_i(P^t \oplus p_i) - \frac{1}{T} \sum_{t=1}^{T} \alpha_i(P^t).$$

The external regret of player i in a minimization game given action sets $P^1, P^2, \ldots, P^T$ is

$$R(T) = \frac{1}{T} \sum_{t=1}^{T} \alpha_i(P^t) - \min_{p_i \in \mathcal{P}_i} \frac{1}{T} \sum_{t=1}^{T} \alpha_i(P^t \oplus p_i).$$

One way to assess the quality of a regret-minimizing algorithm is by the number of time steps T_ϵ it requires before its expected regret is at most ϵ.

We henceforth use the term "regret" generically to refer to external regret. Algorithms that achieve low regret in hindsight are referred to as *universally consistent* or *Hannan consistent*.

Internal regret minimization is more difficult to achieve than external regret minimization, and as such, there are fewer efficient algorithms and they impose more restrictive assumptions on the players. External regret-minimizing algorithms have been known since the 1950's, when Hannan [69] and Blackwell [14] developed such algorithms for repeated two-player games. In cases where each player has only a polynomial number of strategies, Littlestone and Warmuth's weighted majority algorithm [91] can be used to efficiently minimize regret. Recent work on regret minimization has focused on algorithmic efficiency and convergence rates as a function of the number of actions available, and has broadened the set of situations in which no-regret algorithms are known. For example, Kalai and Vempala [81] show that Hannan's algorithm can be used to solve online linear optimization problems with regret approaching 0 at a rate $O(1/\sqrt{T})$, given access to an exact best-response oracle.

Hannan's algorithm and variants of it are known as "follow the (perturbed) leader"-style algorithms, because the approach they take is to always choose the action that has performed the best in hindsight (the "leader"), where the measurements of which action is best have been slightly perturbed. This approach requires that there be an efficient way for the algorithm to constantly track the average performance of all past actions. Another type of regret minimization algorithms pick their action at time step t based only on the action and cost vector of the previous step. For example, Zinkevich [122] develops a regret-minimizing algorithm for online *convex* optimization problems that uses a gradient descent style approach. The algorithm we present in Chapter 3 takes a similar approach.

So-called *bandit* algorithms have also been developed [4, 35, 85, 92], which achieve low regret even in the situation where the algorithm receives very limited information after each round of play. Specifically, those results provide efficient algorithms for many situations in which the number of strategies for each player is exponential in the size of the natural representation of the game.

The convergence rates achieved by modern regret minimizing algorithms are quite good: in Hannan's original algorithm [69], the number of time steps needed to achieve a gap of ϵ with respect to the best fixed strategy in hindsight—the "per time step regret"—is linear in the size of the game N. This was reduced to $O(\log N)$ in more recent exponential-weighting algorithms for this problem [19, 59, 91] (also called the problem of "combining expert advice"). Most recently, a number of algorithms have been developed for achieving such guarantees *efficiently* in many settings where the number of choices N is exponential in the natural description-length of the problem [81, 117, 122]. For example, for the case of a routing game consisting of only two nodes and m parallel edges, exponential-weighting algorithms [19, 59, 91] give $T_\epsilon = O(\frac{1}{\epsilon^2} \log m)$. For general graphs, results of Kalai and Vempala yield $T_\epsilon = O(\frac{mn \log n}{\epsilon^2})$ [81]. For general graphs where an agent can observe only its path cost, results of Awerbuch and Kleinberg yield $T_\epsilon = \tilde{O}(\frac{n^7 m}{\epsilon^3})$ [4].

In Chapter 3, we show how to use an α-approximate best-response oracle to achieve online performance in linear optimization problems that is close to α times that of the best static solution.

Freund and Schapire [60] show that in a zero-sum game, if all agents use a no external regret minimizing algorithm, the empirical distribution of play converges to the set of minimax equilibria. The set of outcomes to which the empirical distribution of regret minimizing play converges is known as the coarse correlated equilibria of the game [120]. However, there are examples [122] of even quite simple games where regret-minimizing algorithms exhibit cycling behavior and incur costs arbitrarily worse than the cost of the worst Nash equilibrium. Researchers in the AI community have also been interested in the outcomes of regret minimization and have empirically shown that play sometimes converges to Nash equilibrium, and sometimes not [78].

Rather than require that dynamics converge to Nash equilibrium in all games, we can choose to focus on broad classes of games that capture natural models of collaboration and competition, and try to understand their consequences in these games. In this thesis, we propose regret minimization as a reasonable definition of self-interested behavior and study the outcome of such behavior in a variety of classes of repeated games.

Specific approaches to adaptive learning in the computer science literature

Mirrokni and Vetta [97] and Goemans et al. [63] introduce the notion of sink equilibria, which generalize Nash equilibria in a different way than we do in this thesis. In doing so, they abandon simultaneous play, and instead consider sequential myopic best response plays. They analyze sink equilibria in the class of valid games and show that valid games have a price of sinking of between n and $n + 1$. In contrast, we prove that valid games have a price of total anarchy of 2, matching the (Nash) price of anarchy. One reason for this gap is that myopic best responses provide no guarantee about the payoff of any individual player. Indeed, the example in [63] of a valid game with price of sinking n demonstrates that myopic best response is not always rational: In their example, myopic best response players each expect average payoff tending to zero as the number of players increases, whereas they could each easily guarantee themselves payoffs of one on every turn (and would do so if they minimized regret). Additionally, because sink equilibria rely on play entering and never leaving sinks of a best response graph, the price of sinking is brittle to *Byzantine* players who may not be playing best responses. In contrast, in Chapter 5 of this thesis, we show that valid games have a price of total anarchy of 2 even in the presence of arbitrarily many Byzantine players, about whom we make no assumptions.

Fischer and Vöcking [50] consider a specific adaptive dynamics (a particular functional form in which flow might naturally change over time) in the context of selfish routing and prove results about convergence of this dynamics to an approximately stable configuration. In more recent work, they study the convergence of a class of routing policies under a specific model of stale information [51]. Most recently, Fischer, Raecke, and Vöcking [52] give a distributed procedure with especially good convergence properties. The key difference between that work and ours is that those results consider specific adaptive strategies designed to quickly approach equilibrium. In contrast, we are interested in showing convergence for *any* algorithms satisfying the no-regret property. That is, even if the players are using many different strategies, without necessarily knowing or caring about what strategies others are using, then so long as all are no-regret, we show they achieve convergence. In addition, because efficient no-regret algorithms exist even in the bandit setting where each agent gets feedback only about its own actions [4, 92], our results can apply to scenarios in which agents adapt their behavior based on only very limited information and there is no communication at all between different agents.

Convergence time to Nash equilibrium in load balancing has also been studied. Earlier work studied convergence time using potential functions, with the limitation that only one player is allowed to move in each time step; the convergence times derived depended on the appropriate potential functions of the exact model [45, 94]. The work of Goldberg [66] studied a randomized model in which each user can select a random delay over continuous time. This implies that only one user tries to reroute at each specific time; therefore the setting was similar to that mentioned above. Even-Dar and Mansour [44] considered a model where many users are allowed to move concurrently, and derived a logarithmic convergence rate for users following a centrally-moderated greedy algorithm. Most recently, Berenbrink et al. [13] showed weaker convergence results for a specific distributed protocol. To summarize, previous work studied the convergence time to pure Nash equilibria in situations with a centralized mechanism or specific protocol. In contrast, in this thesis we present fast convergence results for approximate Nash equilibria in a non-centralized setting, and our only assumption about the player strategies is that they are all no-regret.

Chien and Sinclair [24] study convergence of decentralized dynamics to approximate equilibria in atomic congestion games. They show that a dynamics wherein players take turns making improving deviations of at least ϵ improvement converges efficiently to an ϵ-approximate equilibrium, assuming that the game is symmetric and that the latency functions satisfy a bounded jump condition.[1] Skopalik and Vocking [114] show that this result does not extend to asymmetric congestion games. Despite this, Awerbuch et al. [5] show polynomial time convergence of ϵ-improvement dynamics in asymmetric games with the bounded jump condition to approximately *optimal* solutions, where the approximation factor achieved is the price of anarchy of the game. A number of other positive results exist for these dynamics, for much more specific classes of games.

2.3 Subsequent work on regret minimization in games

Since the initial publication of the results in this thesis, a number of publications have built on our work. Roughgarden [110] explores the outcomes of regret-minimizing behavior in a variety of classes of games; they are able to show Price of Anarchy style bounds on the social cost, but do not prove convergence results. Kleinberg et al. [86] study agents in atomic congestion games employing a *particular* class of regret-minimization algorithms and show that in many cases, the additional assumptions on the player algorithms allow convergence to *pure* Nash equilibria. Even-Dar et al. [46] demonstrate convergence of general regret-minimizing algorithms to Nash equilibria in a general class of games they call “socially-concave” games. Awerbuch et al. [5] show that a certain type of best response dynamics converges quickly to approximate Nash equilibria in congestion games. General no regret dynamics are much more complex than the dynamics they study, and perhaps better motivated from an individual’s perspective in realistic settings where it is not clear that your opponents will cooperate by also playing best response.

2.4 Work not in this thesis

In some classes of games (such as the well-studied load balancing game [3, 7, 49]), the worst Nash equilibria can result in arbitrarily bad social welfare. However, in some of these games, the bad equilibria are unnatural or artificial, and when modeled realistically, agents might never find or settle at such equilibria. In these cases, one would like tools to understand the stability of equilibria and to better characterize the likely outcomes of selfish behavior.

In joint work with Christine Chung, Kirk Pruhs, and Aaron Roth [27], we employ the stochastic stability framework from evolutionary game theory to study simple dynamics of computationally efficient, imperfect agents. This approach allows us to define a natural dynamic, and from it derive the stable states. We define the *price of stochastic anarchy* to be the ratio of the worst stochastically stable solution to the optimal solution. In games for which the stochastically stable states are a subset of the Nash equilibria, studying the ratio of the worst stochastically stable state to the optimal state can be viewed as a smoothed analysis of the price of anarchy, distinguishing Nash equilibria that are brittle to small perturbations in perfect play from those that are resilient to noise.

[1] Note that the definition of approximate equilibrium that they consider is slightly different from the (more standard) one we use in this thesis; for them, an ϵ-approximate Nash equilibrium is a state from which no player has a deviation with ϵ *multiplicative* improvement.

The evolutionary game theory literature on *stochastic stability* studies n-player games that are played repeatedly. In each round, each player observes her action and its outcome, and then uses simple rules to select her action for the next round based only on her size-restricted memory of the past rounds. In any round, players have a small probability of deviating from their prescribed decision rules. The state of the game is the contents of the memories of all the players. The *stochastically stable states* in such a game are the states with non-zero probability in the limit of this random process, as the probability of deviating approaches zero.

To illustrate the utility of stochastic stability, we study the price of stochastic anarchy of the classic "unrelated load balancing" game [3, 7, 49] under the imitation dynamics of Josephson and Matros [79]. In the load balancing game on unrelated machines, even with only two players and two machines, there are Nash equilibria with arbitrarily high cost, and so the price of anarchy is unbounded. We show that these equilibria are inherently brittle, and that for two players and two machines, the price of stochastic anarchy is 2. This result matches the strong price of anarchy [3] without requiring coordination (at strong Nash equilibria, players have the ability to coordinate by forming coalitions). We further show that in the general n-player, m-machine game, the price of stochastic anarchy, unlike the traditional price of anarchy, is bounded.

The approach in this work is similar to that of the work presented in this thesis: we consider learning algorithms in games not from a *prescriptive* perspective, but instead with the hope that their outcomes are useful *descriptive* tools for understanding the outcomes of repeated game play. One advantage of the work presented in this thesis is that we make extraordinarily minimal assumptions on the learning algorithms (simply that they have no regret in hindsight). Our work on the price of stochastic anarchy, by contrast, is based on particular learning dynamics; it would be interesting to extend this work on understanding the relative stability of outcomes in games by making less restrictive assumptions.

Chapter 3

Approximate Online Linear Optimization

3.1 Introduction

In an offline optimization problem, one must select a single (randomized) decision s from a known set of decisions $\mathcal{S}$, in order to minimize a known cost function. In an offline *linear* optimization problem, a weight vector $w \in \mathbb{R}^n$ is given as input, and the cost function $c(s, w)$ is assumed to be linear in w. Many combinatorial optimization problems fit into this framework, including traveling salesman problems (where $\mathcal{S}$ consists of tours in a graph and w is the assignment of weights to the edges), weighted set cover ($\mathcal{S}$ is the set of covers and w the costs of the sets), and knapsack ($\mathcal{S}$ is the set of feasible sets of items and weights w correspond to item valuations).

Each of these problems has an *online* sequential version, in which on every period the player must select her decision without knowing that period's cost function. That is, there is an unknown sequence of weight vectors $w_1, w_2, \ldots \in \mathbb{R}^n$ and for each $t = 1, 2, \ldots$, the player must select $s_t \in \mathcal{S}$ before w_t is revealed, and pay $c(s_t, w_t)$. In the *full-information* version, the player is then informed of w_t, while in the *bandit* version she is only informed of the value $c(s_t, w_t)$. (The name *bandit* refers to the similarity to the classic multi-armed bandit problem [104]).

The player's goal is to achieve low average cost. In particular, we compare her cost with that of the best fixed decision: she would like her average cost to approach that of the best single point in $\mathcal{S}$, where the best is chosen with the benefit of hindsight. This difference, $\frac{1}{T}\sum_{t=1}^{T} c(s_t, w_t) - \min_{s \in \mathcal{S}} \frac{1}{T}\sum_{t=1}^{T} c(s, w_t)$, is termed *regret*.

For example, in the Online TSP problem, every day, a delivery company serves the same n customers. The company must schedule its daily route without foreknowledge of the traffic on each street. The time on any street may vary unpredictably from day to day due to traffic, construction, accidents, or even competing delivery companies. In *online TSP*, we are given a undirected graph G, and on every period t, we must output a tour that starts at a specified vertex, visits all the vertices at least once, then returns to the initial vertex. After we announce our tour, the traffic patterns are revealed (in the full-information setting, the costs on all the edges; in the bandit setting, just the cost of the tour) and we pay the cost of the tour.

As another example, in the Online Weighted Set Cover problem, every financial quarter, our company hires vendors from a fixed pool of subcontractors to cover a fixed set of tasks. Each subcontractor can handle a known, fixed subset of the tasks, but their price is only announced at the end of the quarter and varies from quarter to quarter. In *online weighted set cover*, the vendors are fixed sets $V_1, \ldots, V_n \subseteq [m]$. Each period, we choose a legal cover $s_t \subseteq [n]$; that is, $\bigcup_{i \in s_t} P_i = [m]$. There is an unknown sequence of cost vectors $w_1, w_2, \ldots \in [0,1]^n$, indicating the quarterly vendor costs. Each quarter, our total cost $c(s_t, w_t)$ is the sum of the costs of the vendors we chose for that quarter. In the full-information setting, at the end of the quarter we find out the price charged by each of the subcontractors; in the bandit setting, we receive a combined bill showing only our total cost.

Prior work showed how to convert an *exact* algorithm for the offline problem into an online algorithm with low regret, both in the full-information setting and in the bandit setting. In particular, Kalai and Vempala showed [81] that using Hannan's approach [69], one can guarantee $O(T^{-1/2})$ regret for any linear optimization problem, in the full-information version, as the number of periods T increases. It was later shown [4, 35, 92] how to convert exact algorithms to achieve $O(T^{-1/3})$ regret in the more difficult bandit setting.

This prior work was actually a reduction showing that one can solve the online problem *nearly as efficiently* as one can solve the offline problem. (They used the offline optimizer as a black box.) However, in many cases of interest, such as the TSP or online combinatorial auction problems [10], even the offline problem is NP-hard. Hannan's "follow-the-perturbed-leader" approach can also be applied to some special types of approximation algorithms, but fails to work directly in general. Finding a reduction that maintains good asymptotic performance using *general* approximation algorithms was posed as an open problem [81]; we resolve this problem.

In this chapter, we show how to convert *any* approximation algorithm for a linear optimization problem into an algorithm for the online sequential version of the problem, both in the full-information setting and in the bandit setting. Our reduction maintains the asymptotic approximation guarantee of the original algorithm, relative to the average performance of the best static decision in hindsight. Our new approach is inspired by Zinkevich's algorithm for the problem of minimizing convex functions over a convex feasible set $\mathcal{S} \subseteq \mathbb{R}^n$ [122]. However, the application is not direct and requires a geometric transformation that can be applied to any approximation algorithm.

3.1.1 Hannan's approach

In this section, we briefly describe the previous approach [81] for the case of exact optimization algorithms based on Hannan's idea of adding perturbations. We begin with the obvious "follow-the-leader" algorithm which, each period, picks the decision that is best against the total (equivalently, average) of the previous weight vectors. This means, on period t, choosing $s_t = A\big(\sum_{\tau=1}^{t-1} w_\tau\big)$, where A is an algorithm that, given a cost vector w, produces the best $s \in \mathcal{S}$.[1] Hannan's perturbation idea, in our context, suggests using $s_t = A\big(p_t + \sum_{\tau=1}^{t-1} w_\tau\big)$ for uniformly random perturbation $p_t \in [0, \sqrt{t}]^n$. One can bound the expected regret of following-the-perturbed-leader to be $O(T^{-1/2})$, disregarding other parameters of the problem.

Kalai and Vempala [81] note that Hannan's approach maintains an asymptotic α-approximation

[1] This approach fails even on a two-decision problem, where the costs of the two decisions are (0.5,0) during the first period and then alternate $(1,0), (0,1), (1,0), \ldots$, thereafter.

guarantee when used with α-approximation algorithms with a special property they call *α-pointwise approximation*, meaning that on any input, the solution they find differs from the optimal solution by a factor of at most α in every coordinate. They observe that a number of algorithms, such as the Goemans-Williamson max-cut algorithm [64], have this property. Balcan and Blum [10] observe that the previous approach applies to another type of approximation algorithm: one that uses an optimal decision for another linear optimization problem, for example, using MST for metric TSP. It is also not difficult to see that a FPTAS can be used to get a $(1+\epsilon)$-competitive online algorithm. We further note that the Hannan-Kalai-Vempala approach extends to approximation algorithms that perform a simple type of randomized rounding where the randomness does not depend on the input.

In the next section, we use an explicit example based on the greedy set-cover approximation algorithm to illustrate how Hannan's approach fails on more general approximation algorithms.

3.1.2 Example where "follow-the-perturbed-leader" fails

First consider the set $\mathcal{S} = \{1, 2, \ldots, n\}$ and the cost sequence $(1, 1, \ldots, 1)$ (repeated $T/(n+1)$ times), $(1, 0, \ldots, 0)$ (repeated $T/(n+1)$ times), $(0, 1, 0, \ldots, 0)$ (repeated $T/(n+1)$ times),…, $(0, \ldots, 0, 1)$ (repeated $T/(n+1)$ times). Notice that selecting a decision with cost 1 is always a valid $(\alpha = 2)$-approximation to the leader on the previous examples. Moreover, its cost is T while the cost of the best (in fact *every*) $s \in \mathcal{S}$ is $2T/(n+1)$, hence giving large α-regret. Unfortunately, adding perturbations of $O(\sqrt{T})$ as in follow-the-perturbed-leader will not significantly improve matters: when $T/(n+1) \gg \sqrt{T}$, choosing a decision that costs 1 each period is still an α-approximation for, say, $\alpha = 3$.

Of course, one may be suspicious that no common approximation algorithms would have such peculiar behavior. We now give a similar example based on the standard greedy set cover approximation algorithm ($\alpha = \log m$) applied to the online set cover problem described earlier. The example has $n/2$ covers of size 2: $S_i = S \setminus S_{n+1-i}$, for $i = 1, 2, \ldots, n$. Furthermore, suppose the sets are of increasing size $|S_i| = \left(0.4 + 0.2\frac{i-1}{n-1}\right)m$ and $|S_i \cup S_j| \leq 0.9m$ for all $1 \leq i, j \leq n$ where $i \neq n+1-j$.[2] The sequence of costs (weight) vectors is divided into $n/2$ phases $j = 0, 1, \ldots, n/2 - 1$, each consisting of $2T/n$ identical cost vectors. In phase $j = 0$, all sets have cost 1. For phase $j = 1, \ldots, n/2 - 1$: the cost of the j sets $S_1, \ldots, S_j$ and the $j - 1$ sets $S_{n-j+2}, \ldots, S_n$ are all 1, while the costs of the remaining sets are all 0.

In this example, following the leader with greedy set cover will have an average per-period cost of at least 0.1. In particular, during the first 10% of any phase $j \geq 1$, either greedy's first choice will be S_{n-j+1}, in which case its second choice will be S_j (because any other set covers at most 90% of the remaining items, and S_j's cost so far is at most 10% more than that of any other set), or greedy's first choice will be one of $S_{n-j}, \ldots, S_n$. In either case it pays at least 1 during that period. Hence, following the leader pays at least $0.1 + \frac{19}{5}n$ in expectation on average, while the cover $S_{n/2} \cup S_{n/2+1}$ has an average cost of only $4/n$, which is far from matching greedy's $\alpha = \log m$ approximation ratio (for $n = \Theta(m)$).

[2] To design such a collection of sets (for even n and $m = 5(n-1)$), take S_i to be a uniformly random set of the desired size m for $i = 1, \ldots, n/2$, and S_{n+1-i} to be its complement. It is not hard to argue that, with high probability, the randomized construction obeys the stated properties.

Also note that perturbations on the order of $O(\sqrt{T})$ will not solve this problem. It would be very interesting to adapt Hannan's approach to work for approximation algorithms, especially because it is more efficient than our approach. However, we have not found a solution that works across problems.

3.1.3 Informal statement of results

The main result of this chapter is a general conversion from any approximate linear optimization algorithm to an approximate online version in the full-information setting (§3.3). The extension to the bandit setting (§3.4) uses well-understood techniques, modulo one new issue that arises in the case of approximation algorithms. We summarize the problem, our approach, and our results here.

We assume there is a known compact convex set $\mathcal{W} \subseteq \mathbb{R}^n$ of legal weight vectors (in many cases $\mathcal{W} = [0,1]^n$), and a cost function $c : \mathcal{S} \times \mathcal{W} \to [0,1]$ that is *linear* in its second argument, that is, $c(s, av + bw) = ac(s,v) + bc(s,v)$ for all $s \in \mathcal{S}$, $a, b \in \mathbb{R}$, and $v, w, av + bw \in \mathcal{W}$. The generalization to $[0, M]$-bounded cost functions for $M > 0$ is straightforward.[3] We assume that we have a black-box α-approximation algorithm, which we abstract as an oracle A such that, for all $w \in \mathcal{W}$, $c(A(w), w) \leq \alpha \min_{s \in \mathcal{S}} c(s, w)$. That is, we do not assume that our approximation oracle can optimize in every direction, but only that it can be called on weights in $\mathcal{W}$. For example, approximation algorithms for graph problems can often only handle inputs with non-negative edge weights. In the full-information setting, we assume our only access to $\mathcal{S}$ is via the approximation algorithm; in the bandit setting, we need an additional assumption, which we describe below.

In this chapter, we focus on the *non-adaptive setting*, in which the adversary's choices of w_t can be arbitrary but must be chosen in advance. In the *adaptive setting*, on period t, the adversary may choose w_t based on $s_1, w_1, \ldots, s_{t-1}, w_{t-1}$. In the bandit case, extension of these results to the adaptive setting and the conversion from results in expectation to high probability results remain open questions.

For α-approximation algorithms, it is natural to consider the following notion of α-*regret*, in both the full-information and the bandit-settings. It is the difference between the algorithm's average cost and α times the cost of the best $s \in \mathcal{S}$, that is,

$$\frac{1}{T}\sum_{t=1}^{T} c(s_t, w_t) - \alpha \min_{s \in \mathcal{S}} \frac{1}{T}\sum_{t=1}^{T} c(s, w_t).$$

Note that if there is a hardness of approximation result with ratio α for the offline version of a problem, one cannot expect to obtain better than α-regret efficiently in the online setting.

Full-information results

Our approach to the full-information problem is inspired by Zinkevich's algorithm (for a somewhat different problem) [122], which uses an exact projection oracle to create an online algorithm with low regret. An exact projection oracle Π_J is an algorithm which can produce

[3] In [81], the set $\mathcal{W} = \{w \in \mathbb{R}^n \mid |w|_1 \leq 1\}$ was assumed.

$\operatorname{argmin}_{x \in J} ||x - y||$ for all $y \in \mathbb{R}^n$, where J is the "feasible region" (in Zinkevich's setting, a compact convex subset of $\mathbb{R}^n$). The main algorithm presented in Zinkevich's paper, GREEDY PROJECTION, determines its decision x_t at time t as $x_t = \Pi_J(x_{t-1} - \eta w_{t-1})$, where η is a parameter called the learning rate and w_{t-1} is the cost vector at time $(t-1)$. One can view the approach presented here as providing a method to simulate a type of "approximate" projection oracle using an approximation algorithm. In §3.3 we show the following:

Result 3.1.1. *Given any α-approximation oracle to an offline linear-optimization problem and any $T, T_0 \geq 1$, $w_1, w_2, \ldots \in \mathcal{W}$, our (full-information) algorithm (Algorithm 3.3.1) outputs $s_1, s_2, \ldots \in \mathcal{S}$ achieving*

$$\mathrm{E}\left[\frac{1}{T}\sum_{t=T_0+1}^{T_0+T} c(s_t, w_t)\right] - \alpha \min_{s \in \mathcal{S}} \frac{1}{T}\sum_{t=T_0+1}^{T_0+T} c(s, w_t) = \frac{O(\alpha n)}{\sqrt{T}}.$$

The algorithm makes poly(n, T) calls to the approximation oracle.

Note that the above bound on expected α-regret holds simultaneously for every window of T consecutive periods (T must be known by the algorithm). We easily inherit this useful adaptation property of Zinkevich's algorithm. It is not clear to us whether one could elegantly achieve this property using the previous approach.

Bandit results

Previous work in the bandit setting constructs an "exploration basis" to allow the algorithm to discover better decisions [4, 35, 92]. In particular, Awerbuch and Kleinberg [4] introduce a so-called Barycentric Spanner (BS) as their exploration basis and show how to construct one from an optimization oracle $A : \mathbb{R}^n \to \mathcal{S}$. However, in the case where the oracle (exact or approximate) only accepts inputs in, say, the positive orthant, it may be impossible to extract an exploration basis. Hence, we assume that we are given a β-BS ($\beta \geq 1$ is an approximation factor for the BS) for the problem at hand as part of the input. We define and discuss these concepts further in Section 3.4. Note that the β-BS only needs to be computed once for a particular problem and then can be reused for all future instances of that problem. Given a β-BS, the standard reduction from the bandit setting to the full-information setting gives:

Result 3.1.2. *For any β-BS and any α-approximation oracle to an offline linear-optimization problem and any $T, T_0 \geq 1$, $w_1, w_2, \ldots \in \mathcal{W}$, the (bandit) algorithm in Figure 3.4 outputs $s_1, s_2, \ldots \in \mathcal{S}$ achieving*

$$\mathrm{E}\left[\frac{1}{T}\sum_{t=T_0+1}^{T_0+T} c(s_t, w_t)\right] - \alpha \min_{s \in \mathcal{S}} \frac{1}{T}\sum_{t=T_0+1}^{T_0+T} c(s, w_t) = \frac{O(n(\alpha\beta)^{2/3})}{\sqrt[3]{T}}.$$

The algorithm makes poly(n, T) calls to the approximation oracle.

We also show, in §3.4.1, that the assumption of a BS is necessary.

Result 3.1.3. *There is no polynomial-time black-box reduction from an α-approximation algorithm for a general linear optimization problem (without additional input) to a bandit algorithm guaranteeing low α-regret.*

We note that the above regret is sub-optimal in terms of the T dependence. Furthermore, recent work [1, 11, 34] presents algorithms for online linear optimization that achieve the optimal $\sqrt{T}$ regret even in the bandit setting (these results either do not explicitly consider the computational issues or assume access to an exact optimization oracle). Achieving improved regret for bandit algorithms using approximation oracles remains an open problem.

3.2 Formal definitions

We formalize the natural notion of an n-dimensional linear optimization problem.

Definition 3.2.1. *An n-dimensional linear optimization problem consists of a convex compact set of feasible weight vectors $\mathcal{W} \subset \mathbb{R}^n$, a set of feasible decisions $\mathcal{S}$, and a cost function $c : \mathcal{S} \times \mathcal{W} \to [0, 1]$ that is linear in its second argument.*

Due to the linearity of c, there must exist a mapping $\Phi : \mathcal{S} \to \mathbb{R}^n$ such that $c(s, w) = \Phi(s) \cdot w$ for all $s \in \mathcal{S}, w \in \mathcal{W}$. In the case where the standard basis is contained in $\mathcal{W}$, we have

$$\Phi(s) = \big(c(s, (1, 0, \ldots, 0)), \ldots, c(s, (0, \ldots, 0, 1))\big).$$

More generally, the mapping Φ can be computed directly from c by evaluating c at any set of vectors whose span includes $\mathcal{W}$. We will assume that we have access to Φ and c interchangeably. Note that previous work represented the problem directly as a geometric problem in $\mathbb{R}^n$, but in our case we hope that making the mapping Φ explicit clarifies the algorithm.

An *α-approximation algorithm* A ($\alpha \geq 1$) for such a problem takes as input any vector $w \in \mathcal{W}$ and outputs $A(w) \in \mathcal{S}$ such that $c(A(w), w) \leq \alpha \min_{s \in \mathcal{S}} c(s, w)$. To ensure that the $\min$ is well-defined, we also assume $\Phi(\mathcal{S}) = \{\Phi(s) \mid s \in \mathcal{S}\}$ is compact.

The performance of an online algorithm is measured by comparing its cost on a sequence of weight vectors with the (approximate) cost of the best static decision for that sequence.

Definition 3.2.2. *The α-regret of an algorithm that selects decisions $a_1, \ldots a_T \in A$ is defined to be*

$$\alpha\text{-}\textit{regret}(a_1, w_1 \ldots, a_T, w_T) = \frac{1}{T} \sum_{t=1}^{T} c(a_t, w_t) - \alpha \min_{a \in A} \frac{1}{T} \sum_{t=1}^{T} c(a, w_t).$$

The term regret by itself refers to 1-regret. The α-regret of a randomized algorithm is defined analogously in terms of the expected costs of its actions.

Define a *projection oracle* $\Pi_J : \mathbb{R}^n \to J$, where $\Pi_J(x) = \operatorname{argmin}_{z \in J} \|x - z\|$ is the unique projection of x to the closest point z in the convex set J.

Define $\mathcal{W}_+ = \{aw | a \geq 0, w \in \mathcal{W}\} \subseteq \mathbb{R}^n$. Note that $\mathcal{W}_+$ is convex, which follows from the convexity of $\mathcal{W}$. We assume that we have an exact projection oracle $\Pi_{\mathcal{W}_+}$. This is generally straightforward to compute. In many cases, $\mathcal{W} = [0, 1]^n$, in which case $\mathcal{W}_+$ is the positive orthant and $\Pi_{\mathcal{W}_+}(w)[i]$ is simply $\max(w[i], 0)$, where $w[i]$ denotes the ith component of vector w. More generally, given a membership oracle to $\mathcal{W}_+$ (or to a $\mathcal{W}$ with a smoothness guarantee), a point $w_0 \in \mathcal{W}$, and appropriate bounds on the radii of contained and containing balls, one can approximate the projection to within any desired accuracy $\epsilon > 0$ in time poly$(n, \log(1/\epsilon))$. Note that we will later be dealing with the difficulty of projecting essentially onto $\mathcal{S}$, which is a more difficult problem because our only access to it is via an approximation oracle.

We also assume, for convenience, that $A : \mathcal{W}_+ \to \mathcal{S}$ because we know that $A(w)$ can be chosen to be equal to $A(aw)$ for any $a > 0$, and finding a such that $aw \in \mathcal{W}$ is a one-dimensional problem. (Again, given a membership oracle to $\mathcal{W}$ one can find $v \in \mathcal{W}$ which is within ϵ of being a scaled version of w using time poly($n, 1/\epsilon$)). However, the restriction on the approximation algorithm's domain is important because many natural approximation algorithms only apply to restricted domains such as non-negative weight vectors.

In a nonadaptive *online linear optimization* problem, there is a sequence $w_1, w_2, \ldots, \in \mathcal{W}$ of weight vectors. Due to the linearity of the problem, an *offline optimum* can be computed using an exact optimizer, that is, $\min_{s \in \mathcal{S}} \frac{1}{T} \sum_{t=1}^{T} \Phi(s) \cdot w_t = \min_{s \in \mathcal{S}} \Phi(s) \cdot \left(\frac{1}{T} \sum_{t=1}^{T} w_t \right)$ gives the average cost of the best single decision if one had to use a single decision during all time periods $t = 1, 2, \ldots, T$. Similarly, an α-approximation algorithm, when applied to $\frac{1}{T} \sum_{t=1}^{T} w_t$, gives a decision whose average cost is not more than a factor α larger than that of the offline optimum.

Definition 3.2.3. *In a full-information online linear optimization problem, there is an unknown sequence of weight vectors $w_1, w_2, \ldots \in \mathcal{W}$ (possibly chosen by an adversary). On each period, the decision-maker chooses a decision $s_t \in \mathcal{S}$ based on $s_1, w_1, s_2, w_2, \ldots, s_{t-1}, w_{t-1}$. Then w_t is revealed and the decision-maker incurs cost $c(s_t, w_t)$.*

Finally, we define the bandit version of the problem, in which the algorithm finds out only the cost of its decision, $c(s_t, w_t)$, but *not* w_t itself.

Definition 3.2.4. *In a bandit online linear optimization problem, there is an unknown sequence of weight vectors $w_1, w_2, \ldots \in \mathcal{W}$ (possibly chosen by an adversary). On each period, the decision-maker chooses a decision $s_t \in \mathcal{S}$ based only upon $s_1, c(w_1, s_1), \ldots, s_{t-1}, c(w_{t-1}, s_{t-1})$. Then only the cost $c(s_t, w_t)$ is revealed.*

For $x, y \in \mathbb{R}^n$ and $\mathcal{W} \subseteq \mathbb{R}^n$, we say x *dominates* y if $x \cdot w \leq y \cdot w$ for all $w \in \mathcal{W}$ (equivalently, for all $w \in \mathcal{W}_+$).[4]

Define $K \subseteq \mathbb{R}^n$ to be the convex hull of $\Phi(\mathcal{S})$,

$$K = \left\{ \sum_{i=1}^{n+1} \lambda_i \Phi(s_i) \middle| \; s_i \in \mathcal{S}, \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\}.$$

Note that $\min_{x \in K} x \cdot w = \min_{s \in \mathcal{S}} c(s, w)$ for all $w \in \mathcal{W}$. The cost of any point in K can be achieved by choosing a randomized combination of decisions $s \in \mathcal{S}$. However, we must find such a combination of decisions and compute projections in our setting, where our only access to $\mathcal{S}$ is via an approximation oracle.

3.3 Full-information algorithm

We now present our algorithm for the full-information setting. Define $z_t = x_t - \eta w_t$. Intuitively, one might like to play z_t on period $t+1$ because z_t has less cost than x_t against w_t. Unfortunately, z_t may not be feasible. In the GREEDY PROJECTION algorithm of Zinkevich, the decision played on period $t + 1$ is the projection of z_t into the feasible set. Our basic approach is to implement an approximate projection algorithm and play the approximate projection of z_t on step $t + 1$.

[4] Note that this definition differs from the standard definition in $\mathbb{R}^n$ where x dominates y if $x[i] \geq y[i]$ for all i but resembles the game-theoretic notion of dominant strategies.

```
Input: $x, z \in \mathbb{R}^n$, $s \in \mathcal{S}$, and an $\alpha$-approximation algorithm $A$ (and parameters $\delta > 0$, $\lambda \in [0,1]$).
Output: $(x', s') \in \Pi^{\delta}_{\alpha K} \times \mathcal{S}$
Define $B$ to be the extended approximation oracle obtained from $A$ using Lemma 3.3.5.
APPROX-PROJ$(z, s, x)$
Let $(t, y) := B(x - z)$
 if $x \cdot (x - z) \leq \delta + y \cdot (x - z)$
   then return $(x, s)$
   else $q = \begin{cases} s & \text{with probability } 1 - \lambda \\ t & \text{with probability } \lambda \end{cases}$
        return APPROX-PROJ$(z, q, \lambda y + (1 - \lambda)x)$
```

Figure 3.1: A recursive algorithm for computing approximate projections.

There are a number of technical challenges to this approach. First, we only have access to an α-approximation oracle with which to implement this. Due to the multiplicative nature of this approximation, we proceed by attempting to project into the set αK, where $\alpha K = \{\alpha x | x \in K\}$. Second, even if we could do this perfectly (which is not possible), this would still not result in a feasible decision. We then must find a way to play a feasible decision.

We can intuitively view our algorithm as follows. The algorithm keeps track of a parameter x_t, which we can think of as the attempt to project z_{t-1} into αK (though this is not done exactly, as x_t is not even in αK). We show that if the algorithm actually were allowed to play x_t then it would have low α-regret. Our algorithm uses this x_t to find a randomized feasible decision s_t. We show that the expected cost of this random feasible decision s_t is no larger than that of the (potentially) infeasible x_t.

Our algorithm for the full-information setting is based on the approximate projection routine defined in Figure 3.1.

Algorithm 3.3.1. *The algorithm is given a learning parameter η. On period 1, we choose an arbitrary s_1 (which could be selected by running the approximation oracle on any input) and let $x_1 = \Phi(s_1)$. On period t, we play s_t and let*

$$(x_{t+1}, s_{t+1}) = \text{APPROX-PROJ}(x_t - \eta w_t, s_t, x_t).$$

It may be helpful to the reader to note that the sequence x_t is deterministically determined (if the approximation oracle is deterministic) by the sequence of weights $w_1, \ldots, w_{t-1}$, while s_t is necessarily randomized.

In §3.3.1, we show that if we had a particular kind of approximate projection algorithm, then the x_t values produced by that algorithm would have (hypothetical) low α-regret. In §3.3.2, we show how to extend the domain of any approximation algorithm, which allows us to construct such an approximate projection algorithm: the APPROX-PROJ algorithm used in Algorithm 3.3.1. We also show that the cost of the (infeasible) decision x_{t+1} it produces can only be larger than the expected cost incurred by the feasible decision s_{t+1} it also generates. This will allow us to prove our main theorem in the full-information setting:

Theorem 3.3.2. *Consider an n-dimensional online linear optimization problem with feasible set $\mathcal{S}$ and mapping $\Phi : \mathcal{S} \to \mathbb{R}^n$. Let A be an α-approximation algorithm and take $R, W > 0$ such that $\|\Phi(A(w))\| \leq R$ and $\|w\| \leq W$ for all $w \in \mathcal{W}$.*

For any fixed $w_1, w_2, \dots w_T \in \mathcal{W}$ and any $T \geq 1$, with learning parameter $\eta = \frac{(\alpha+1)R}{W\sqrt{T}}$, approximate projection tolerance parameter $\delta = \frac{(\alpha+1)R^2}{T}$, and learning rate parameter $\lambda = \frac{(\alpha+1)}{4(\alpha+2)^2 T}$, Algorithm 3.3.1 achieves expected α-regret at most

$$\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T} c(s_t, w_t)\right] - \alpha \min_{s \in \mathcal{S}} \frac{1}{T}\sum_{t=1}^{T} c(s, w_t) \leq \frac{(\alpha+2)RW}{\sqrt{T}}.$$

Each period, the algorithm makes at most $4(\alpha+2)^2 T$ calls to A and Φ.

We present the proof of Theorem 3.3.2 in §3.3.4. To get Result 3.1.1 in the introduction, we note that it is possible to get a priori bounds on W and R by a simple change of basis so that $RW = O(n)$. It is possible to do this from the set $\mathcal{W}$ alone. In particular, one can compute a 2-barycentric spanner (BS) $e_1, \dots, e_n$ for $\mathcal{W}$ [4] and perform a change of basis so that $\Phi(e_1), \dots, \Phi(e_n)$ is the standard basis (as we describe in greater detail in §3.4). By the definition of a 2-BS, this implies that $\mathcal{W} \subseteq [-2, 2]^n$ and hence $W = 2\sqrt{n}$ is a satisfactory upper bound. Since we have assumed that all costs are in $[0, 1]$ and the standard basis is in $\mathcal{W}$, this implies that $\Phi(S) \subseteq [0, 1]^n$ and hence $R = \sqrt{n}$ is also a valid upper bound. The guarantees with respect to every window of T consecutive periods hold because our algorithm's guarantees hold starting at arbitrary (s_t, x_t) such that $\mathrm{E}[\Phi(s_t)]$ dominates x_t (recall, s_t is necessarily randomized).

3.3.1 Approximate Projection

We first define the notion of approximate projection. Because we only have access to an α-approximate oracle, given $z \in \mathbb{R}^n$, we cannot find the closest point to z in K or even in $\alpha K = \{\alpha x | x \in K\}$.

Note that for a closed convex set $J \subseteq \mathbb{R}^n$, if $\Pi_J(z) = x$, then

$$(x - z) \cdot x \leq \min_{y \in J} (x - z) \cdot y.$$

This is essentially the separating hyperplane theorem (where $x - z$ is the normal vector to the separating hyperplane). Also note that $\Pi_J(x) = x$ if $x \in J$.

Our approximate projection property, illustrated in Figure 3.2, relaxes the above condition. Due to the computational issues associated with optimizing over K even with access to an *exact* optimization oracle ($\alpha = 1$), [5] our projections will be parametrized by an additional δ. Define the set of δ-approximate projections to be, for $\delta \geq 0$ and any $z \in \mathbb{R}^n$,

$$\Pi_J^{\delta}(z) = \{x \in \mathbb{R}^n \mid (x - z) \cdot x \leq \min_{y \in J} (x - z) \cdot y + \delta\}.$$

It is important to note that we have not required an approximate projection to be in J However, note that in the case where the projection is in J, and $\delta = 0$, it is exactly the projection, that is,

[5] We are not assuming that K is defined by a polynomial number of hyperplanes—it can be quite round.

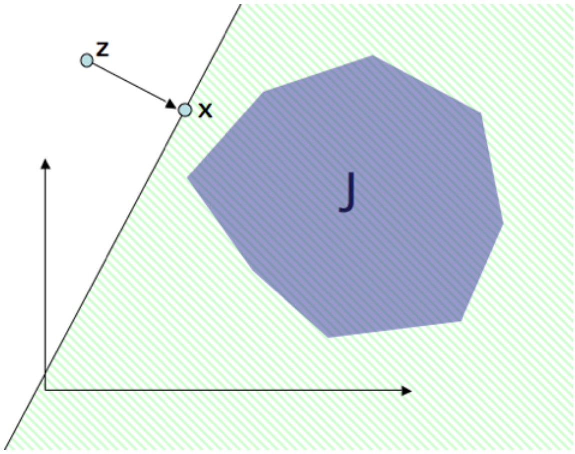

Figure 3.2: An approximate projection oracle, for convex set $J \subseteq \mathbb{R}^n$ and $\delta = 0$, returns a point $\Pi_J^0(z) \in \mathbb{R}^n$ that is closer to any point $y \in J$ than z is, that is, $\forall y \in J \ \|\Pi_J^0(z) - y\| \leq \|z - y\|$.

$\Pi_J^\delta(z) \cap J = \{\Pi_J^0(z)\}$. For $\delta = 0$, the approximate projection is a point $\Pi_J^0(z) \in \mathbb{R}^n$ that is closer to any point $y \in J$ than z is, that is, $\forall y \in J \ \|\Pi_J^0(z) - y\| \leq \|z - y\|$. While we refer to it as an approximate projection, it is also clearly related to a separation oracle. From a hyperplane separating z from J, one can take the closest point on that hyperplane to z as an approximate projection, or in fact $z \in \Pi_J^\delta(z)$. The difficulty we will face is in finding a *feasible* such point.

We now bound the α-regret of the hypothetical algorithm which projects with $\Pi_{\alpha K}^\delta$. The proof is essentially a straightforward extension of Zinkevich's proof [122]. This lemma shows that indeed this hypothetical algorithm has a graceful degradation in quality.

Lemma 3.3.3. *Let $K \subseteq \mathbb{R}^n$ be a convex set such that $\forall x \in K,\ \|x\| \leq R$. Let $w_1, \ldots, w_T \in \mathbb{R}^n$ be an arbitrary sequence. Then, for any initial point $x_1 \in K$, any $\alpha > 1$, and any sequence $x_1, x_2, \ldots, x_T$ such that $x_{t+1} \in \Pi_{\alpha K}^\delta(x_t - \eta w_t)$,*

$$\frac{1}{T}\sum_{t=1}^{T} x_t \cdot w_t - \alpha \min_{x \in K} \frac{1}{T}\sum_{t=1}^{T} x \cdot w_t \leq \frac{(\alpha+1)^2 R^2}{2\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T} w_t^2 + \frac{\delta}{\eta}.$$

Proof. Let $x^* = \alpha \operatorname{argmin}_{x \in K} \sum_{t=1}^{T} x \cdot w_t$, so $x^* \in \alpha K$. We will bound our performance with respect to x^*. Define the sequence x'_t by $x'_1 = x_1$ and $x'_{t+1} = x_t - \eta w_t$, so that $x_t \in \Pi_{\alpha K}^\delta(x'_t)$. We first claim that $\|x_t - x^*\|^2 \leq \|x'_t - x^*\|^2 + 2\delta$, that is, our attempt at setting x_t to be an approximate projection of x_t onto αK does not increase the distance to x^* significantly:

$$\begin{aligned}(x'_t - x^*)^2 &= \big((x'_t - x_t) + (x_t - x^*)\big)^2 \\ &= (x'_t - x_t)^2 + (x_t - x^*)^2 + 2(x'_t - x_t)\cdot(x_t - x^*) \\ &\geq 0 + (x_t - x^*)^2 - 2\delta.\end{aligned}$$

The last line follows from the definition of approximate projection and the fact that $x^* \in \alpha K$.

Hence, for any $t \geq 1$, because $x'_{t+1} = x_t - \eta w_t$ we have

$$\begin{aligned}(x_{t+1} - x^*)^2 &\leq (x_t - \eta w_t - x^*)^2 + 2\delta \\ &= (x_t - x^*)^2 + \eta^2 w_t^2 - 2\eta w_t \cdot (x_t - x^*) + 2\delta\end{aligned}$$

and thus

$$w_t \cdot (x_t - x^*) \leq \frac{(x_t - x^*)^2 - (x_{t+1} - x^*)^2 + \eta^2 w_t^2 + 2\delta}{2\eta}.$$

Using a telescoping sum of the above, we get

$$\sum_{t=1}^{T} w_t \cdot (x_t - x^*) \leq \frac{(x_1 - x^*)^2 - (x_{T+1} - x^*)^2 + \sum_{t=1}^{T} \eta^2 w_t^2 + 2\delta T}{2\eta}.$$

Now using the fact that

$$(x_1 - x^*)^2 \leq (\|x_1\| + \|x^*\|)^2 \leq (\alpha + 1)^2 R^2,$$

we get

$$\sum_{t=1}^{T} x_t \cdot w_t - \alpha \min_{x \in K} \sum_{t=1}^{T} x \cdot w_t \leq \frac{(\alpha+1)^2 R^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} w_t^2 + T\frac{\delta}{\eta}$$

as desired. □

Note that if we set $\eta = 1/\sqrt{T}$, the sum of the first two terms of this bound would be $O(1/\sqrt{T})$. However, the last term, $\frac{\delta}{\eta}$, would be $O(\delta\sqrt{T})$. Hence, we need to achieve an approximation quality of $\delta = O(1/T)$ in order for the α-regret of our (infeasible) x_t values to be $O(1/\sqrt{T})$.

3.3.2 Constructing the Algorithm

One simple method to (approximately) find the projection of z into a convex set J, given an exact optimization oracle for J, is as follows. Start with a point in $x \in J$. Then choose the search direction $v = x - z$, and find a minimal point $x' \in J$ in the direction of v—that is, $x' \in J$ such that $x' \cdot v \leq \min_{y \in J} y \cdot v$ (or, equivalently, such that $(x' - z) \cdot v \leq \min_{y \in J}(y - z) \cdot v$). It can be seen that if x is not minimal in the direction of v, then there must be a point on the segment joining x' and z that is closer to z than x was. Then repeat this procedure starting at x'. In the case where $z \in J$, this will be still be useful in representing z nearly as a combination of points output by the minimization algorithm.[6]

Note that in our case if $v \in \mathcal{W}_+$, then our approximation oracle is able to find a feasible $s \in \mathcal{S}$ such that

$$\Phi(s) \cdot v \leq \alpha \min_{s' \in \mathcal{S}} \Phi(s') \cdot v = \min_{x \in \alpha K} x \cdot v.$$

Loosely speaking, our oracle is able to perform minimization with respect to the set $J = \alpha K$ (or better). This is essentially how our algorithm will use the approximation oracle. However, as mentioned before, many approximation algorithms can only handle non-negative weight vectors or weight vectors from some other limited domain. Hence, we must extend the domain of the oracle when $v \notin \mathcal{W}_+$.

[6] Note that representing a given feasible point as a convex combination of feasible points is similar to *randomized metarounding* [29]. It would be interesting to extend the approach in [29], based on the ellipsoid algorithm, to our problem and potentially achieve a more efficient algorithm. Related but simpler issues arise in [20].

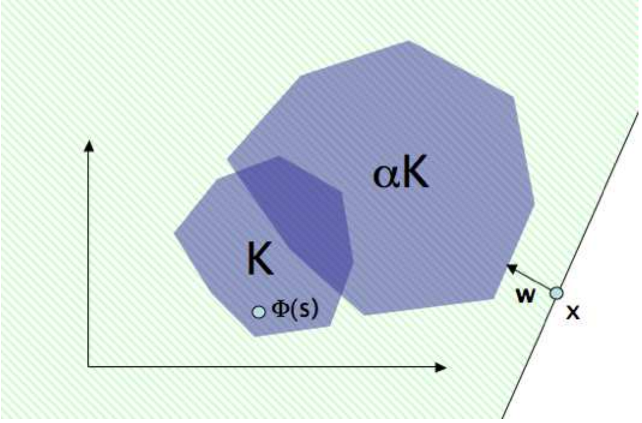

Figure 3.3: An approximation algorithm run on vector $w \in \mathcal{W}$ always returns a point $s \in \mathcal{S}$ such that the set αK is contained in the halfspace tangent to $\Phi(s)$ whose normal direction is w. An extended approximation algorithm, as illustrated here, takes any $w \in \mathbb{R}^n$ as input and returns a point $x \in \mathbb{R}^n$ such that αK is contained in the halfspace tangent to x with normal vector w. In addition, it returns an $s \in \mathcal{S}$ such that $\Phi(s)$ dominates x.

Extending the domain We would like to find a feasible $s \in \mathcal{S}$ that satisfies the search condition $\Phi(s) \cdot v \leq \alpha \min_{s' \in \mathcal{S}} \Phi(s') \cdot v$ for a general $v \in \mathbb{R}^n$, but this is not possible given only an α-approximation oracle that runs on only a subset of $\mathbb{R}^n$. Instead, we attempt to find a (potentially infeasible) $x \in \mathbb{R}^n$ which does satisfy this search condition, and an $s \in \mathcal{S}$ which dominates x, meaning that for all $w \in \mathcal{W}$, $c(s, w) \leq x \cdot w$. More precisely, given any approximation algorithm, we will use it construct the following type of oracle, which we will then use as a tool in our projection algorithm:

Definition 3.3.4. *An extended approximation oracle $B : \mathbb{R}^n \to \mathcal{S} \times \mathbb{R}^n$ is a function such that, for all $v \in \mathbb{R}^n$, if $B(v) = (s, x)$, then $x \cdot v \leq \alpha \min_{s' \in \mathcal{S}} \Phi(s') \cdot v$ and $\Phi(s)$ dominates x.*

Figure 3.3 depicts an extended approximation oracle. The following lemma
demonstrates that one can construct an extended approximation oracle from an approximation oracle.

Lemma 3.3.5. *Let $A : \mathcal{W}_+ \to \mathcal{S}$ be an α-approximation oracle and suppose $\|\Phi(s')\| \leq R$ for all $s' \in \mathcal{S}$. Then the following is an extended approximation oracle: If $v \in \mathcal{W}_+$, then $B(v) = (A(v), \Phi(A(v)))$, else $B(v)$ is*

$$\left(A(\Pi_{\mathcal{W}_+}(v)), \Phi(A(\Pi_{\mathcal{W}_+}(v))) + R(\alpha + 1) \frac{\Pi_{\mathcal{W}_+}(v) - v}{||\Pi_{\mathcal{W}_+}(v) - v||} \right).$$

Proof. For the case where $v \in \mathcal{W}_+$, by definition, $B(v) = (A(v), \Phi(A(v)))$ suffices. Hence, assume $v \notin \mathcal{W}_+$. Let $w = \Pi_{\mathcal{W}_+}(v)$, $s = A(w)$, and $x = \Phi(s) + (\alpha + 1)R\frac{w-v}{||w-v||}$. Then we must show (a) $x \cdot v \leq \alpha \min_{s' \in \mathcal{S}} \Phi(s') \cdot v$ and (b) $\Phi(s)$ dominates x.

We have assumed that A is an α-approximation oracle with domain $\mathcal{W}_+$, and therefore it can accept input w. By the definition of α-approximation, we have $w \cdot \Phi(s) \leq \alpha w \cdot \Phi(s')$ for all $s' \in \mathcal{S}$. By the bound R, we also have that $-\alpha\|v - w\|R \leq \alpha(v - w) \cdot \Phi(s')$ for all $s' \in \mathcal{S}$.

Adding these two gives, for all $s' \in \mathcal{S}$,

$$\begin{aligned}\alpha v \cdot \Phi(s') &\geq w \cdot \Phi(s) - \alpha\|v-w\|R \\ &= v \cdot x + (w-v)\cdot\Phi(s) - (\alpha+1)R\frac{(w-v)}{\|w-v\|}\cdot v - \alpha\|v-w\|R \\ &\geq v\cdot x - \|w-v\|R - (\alpha+1)R\frac{(w-v)}{\|w-v\|}\cdot(v-w) - \alpha\|v-w\|R \\ &= v \cdot x.\end{aligned}$$

This is what we need for part (a) of the lemma. The second-to-last line follows from the fact that $(v-w)\cdot w = 0$. To see this, note that since w is the projection of v onto $\mathcal{W}_+$, we have $(v-w)\cdot(w'-w) \leq 0$ for any $w' \in \mathcal{W}_+$. Since $0 \in \mathcal{W}_+$, this implies that $(v-w)\cdot(-w) \leq 0$. Since $2w \in \mathcal{W}_+$, this implies that $(v-w)\cdot w \leq 0$, and hence $(v-w)\cdot w = 0$.

This also means that $(v-w)\cdot(w'-w) = (v-w)\cdot w' \leq 0$ for all $w' \in \mathcal{W}_+$, which directly implies (b), that is, $(x-\Phi(s))\cdot w' \geq 0$ for all $w' \in \mathcal{W}$. □

Note that the magnitude of the output x is at most $\|\Phi(s)\| + (\alpha+1)R \leq (\alpha+2)R$; this bound will be useful for bounding the runtime of our algorithm.

3.3.3 The approximate projection algorithm

Using this extended approximation oracle, we can define our Approx-Proj algorithm, which we present in Figure 3.1. The following lemma shows that the algorithm returns both a valid approximate projection (which could be infeasible) and a random feasible decision that dominates the approximate projection (assuming that Φ of the algorithm's input s dominated the algorithm's input x).

Lemma 3.3.6. *Suppose* Approx-Proj(z, s, x) *returns* (x', s'). *Then* $x' \in \Pi^{\delta}_{\alpha K}(z)$. *If* s *is a random variable such that* $\mathrm{E}[\Phi(s)]$ *dominates* x*, then* $\mathrm{E}[\Phi(s')]$ *will dominate* x'*.*

It is straightforward to see that the x returned by Approx-Proj satisfies the approximate projection condition. The subtlety is in obtaining a feasible solution with the desired properties. It turns out that t returned by B in line 1 does not suffice, as this t only dominates y, but not necessarily x. However, our randomized scheme does suffice.

of Lemma 3.3.6. The return condition of Approx-Proj states that $x'\cdot(x'-z) \leq \delta + y\cdot(x'-z)$. Using the definition of an extended approximation oracle, we then get

$$\begin{aligned} x'\cdot(x'-z) &\leq \delta + \alpha\min_{s'\in\mathcal{S}}\Phi(s')\cdot(x'-z) \\ &\leq \delta + \min_{y'\in\alpha K} y'\cdot(x'-z)\end{aligned}$$

as desired.

The proof of the second property proceeds by induction on the number of recursive calls made by Approx-Proj. The base case holds trivially. Now suppose the inductive hypothesis holds ($\mathrm{E}[\Phi(s)]$ dominates x). We will show that if $(t, y) = B(x-z)$, the resulting $\mathrm{E}[\lambda\Phi(t) + (1-\lambda)\Phi(s)]$ dominates $\lambda y + (1-\lambda)x$.

We observe:

$$\begin{aligned}
x' \cdot w &= (\lambda y + (1-\lambda)x) \cdot w \\
&= \lambda y \cdot w + (1-\lambda)x \cdot w \\
&\geq \lambda \Phi(t) \cdot w + (1-\lambda)x \cdot w \\
&\geq \lambda \Phi(t) \cdot w + (1-\lambda)\mathrm{E}[\Phi(s)] \cdot w \\
&= \mathrm{E}[\lambda \Phi(t) + (1-\lambda)\Phi(s)] \cdot w \\
&= \mathrm{E}[\Phi(s')] \cdot w.
\end{aligned}$$

Thus, if APPROX-PROJ terminates, the desired conditions will hold. □

3.3.4 Analysis

Our next lemma allows us to bound the number of calls Algorithm 3.3.1 makes to A and Φ on each period.

Lemma 3.3.7. *Suppose that $\lambda, \delta > 0$, the magnitudes of all vectors output by the extended approximation oracle are $\leq \frac{1}{2}\sqrt{\frac{\delta}{\lambda}}$, and $\|x\| \leq \frac{1}{2}\sqrt{\frac{\delta}{\lambda}}$. Then* APPROX-PROJ*($z, s, x$) terminates after at most $\frac{\|x-z\|^2}{\lambda\delta}$ iterations.*

Proof. The analysis is reminiscent of that of the perceptron algorithm (see, e.g., Dunagan and Vempala [42]). Let $H = \frac{1}{2}\sqrt{\frac{\delta}{\lambda}}$. To bound the number of recursive calls to APPROX-PROJ, it suffices to show that the non-negative quantity $\|x - z\|^2$ decreases by at least an additive $\lambda\delta$ on each call and that $\|x\|$ remains below H on successive calls. The latter condition holds because $\|x\|, \|y\| \leq H$ so $\|\lambda y + (1-\lambda)x\| \leq \lambda H + (1-\lambda)H = H$.

Notice that if the procedure does not terminate on a particular call, then

$$(x - y) \cdot (x - z) > \delta.$$

This means that the decrease in $(x-z)^2$ in a single recursive call is

$$\begin{aligned}
(x-z)^2 - (\lambda y + (1-\lambda)x - z)^2 &= (x-z)^2 - (\lambda(y-x) + (x-z))^2 \\
&= 2\lambda(x-y)\cdot(x-z) - \lambda^2(y-x)^2 \\
&> 2\lambda\delta - \lambda^2(y-x)^2.
\end{aligned}$$

Also, $\|y - x\| \leq 2H$. Combining this with the previous observation gives

$$(x-z)^2 - (\lambda y + (1-\lambda)\, x - z)^2 > 2\lambda\delta - 4\lambda^2 H^2 = \lambda\delta.$$

Hence the total number of iterations of APPROX-PROJ on each period is at most $\|x - z\|^2/(\lambda\delta)$. □

This lemma gives us a means of choosing λ. We are now ready to prove our main theorem about full-information online optimization.

of Theorem 3.3.2. Take $\eta = \frac{(\alpha+1)R}{W\sqrt{T}}$, $\delta = \frac{(\alpha+1)R^2}{T}$, and $\lambda = \frac{(\alpha+1)}{4(\alpha+2)^2T}$. Since $x_1 = \Phi(s_1)$, by induction and Lemma 3.3.6, we have that $\mathrm{E}[\Phi(s_t)]$ dominates x_t for all t. Hence, it suffices to upper-bound $\sum_{t=1}^{T} x_t \cdot w_t$. By Lemma 3.3.6, we have that $x_t \in \Pi^{\delta}_{\alpha K}(z_{t-1})$ on each period, so by Lemma 3.3.3 we get

$$\mathrm{E}[\alpha\text{-regret}] \leq \frac{1}{T}\left(\frac{(\alpha+1)^2R^2}{2\eta} + T\frac{\delta}{\eta} + \frac{\eta}{2}TW^2\right).$$

Applying our chosen values of η and δ, this gives an α-regret bound of $\frac{1}{T}((\alpha+1)RW\sqrt{T} + RW\sqrt{T}) = \frac{(\alpha+2)RW}{\sqrt{T}}$ as desired.

Now, as mentioned, the extended approximation oracle from Lemma 3.3.5 has the property that it returns vectors of magnitude at most $H = \frac{1}{2}\sqrt{\frac{\delta}{\lambda}} = (\alpha+2)R$. Furthermore, it is easy to see that all vectors x_t have $\|x_t\| \leq H$, by induction on t. Then by Lemma 3.3.7, the total number of iterations of APPROX-PROJ period t is at most $(2H\|x-z\|/\delta)^2 \leq (2(\alpha+2)R\eta W/\delta)^2 = 4(\alpha+2)^2T$. □

3.4 Bandit algorithm

We now describe how to extend Algorithm 3.3.1 to the partial-information model, where the only feedback we receive is the cost we incur at each period. Flaxman et al. [53] also use a gradient descent style algorithm for online optimization in the bandit setting, but the details of their approach differ significantly from ours. The algorithm we describe here requires access to an *exploration basis* $e_1, \ldots, e_n \in \mathcal{S}$, which is simply a set of n decisions such that $\Phi(e_1), \ldots, \Phi(e_n)$ span $\mathbb{R}^n$. (If no such decisions exist, one can reduce the problem to a lower-dimensional problem.) Following previous approaches, we will (probabilistically) try each of these decisions from time to time. As in the work of Dani and Hayes [35], we will assume that $\Phi(e_i)$ is the standard ith basis vector, that is, $e_i[i] = 1$ and $e_i[j] = 0$ for $j \neq i$. This assumption makes the algorithm cleaner to present, and is without loss of generality because we can always use $\Phi(e_i)$ as our basis for representing $\mathbb{R}^n$.

Definition 3.4.1. *A set $\{x_1, x_2, \ldots x_m\} \subseteq S$ is a β-barycentric spanner (BS) for $S \subset \mathbb{R}^n$ if, for every $x \in S$, x can be written as $x = \beta_1 x_1 + \ldots + \beta_m x_m$ for some $\beta_1, \ldots, \beta_m \in [-\beta, \beta]$.*

Note that we only need to construct a BS once for any problem, and then can re-use it for all future instances of the problem.

Awerbuch and Kleinberg [4] prove that every compact S has a 1-BS of size n, and, moreover, give an algorithm for finding a size-n $(1+\epsilon)$-BS using poly$(n, \log(1/\epsilon))$ calls to an exact minimization oracle $M : \mathbb{R}^n \rightarrow \mathcal{S}$, where $M(v) \in \operatorname{argmin}_{s \in \mathcal{S}} \Phi(s) \cdot v$. Unfortunately, as we show in §3.4.1, one cannot find such a BS using a minimizer (exact or approximate) whose domain is not all of $\mathbb{R}^n$. Moreover, we show that one cannot guarantee low regret for the bandit problem using just a black-box optimization algorithm $A : \mathcal{W}_+ \rightarrow \mathcal{S}$.

Hence, we assume that we are given a β-BS for the problem at hand as part of the input. We feel that this is a reasonable assumption. For example, note that it is easy to find such a basis for TSP and set cover with $\beta =$poly(n): In the case of set cover, one can take the n covers consisting

```
Given $\delta, \eta, \gamma > 0$ and an initial point $\hat{s}_1$ as input, set $\hat{x}_1 = \Phi(\hat{s}_1)$. Perform a change of basis so that $\Phi(e_1), \ldots, \Phi(e_n)$ is the standard basis.

for $t = 1, 2, \ldots$:
    With probability $\gamma$,        $\triangleright$ exploration step
            Choose $i \in \{1, \ldots, n\}$ uniformly at random.
            $s_t := e_i; x_t := \Phi(e_i)$.
            Play($s_t$).
            Observe $\ell_t = c(s_t, w_t)$.
            $\hat{w}_t := (n\ell_t/\gamma)\Phi(e_i)$.
            $(\hat{x}_{t+1}, \hat{s}_{t+1}) :=$ APPROX-PROJ$(\hat{x}_t - \eta\hat{w}_t, \hat{s}_t, \hat{x}_t)$.
    else, with probability $1 - \gamma$,        $\triangleright$ exploitation step
            $s_t := \hat{s}_t; x_t := \hat{x}_t$.
            Play($s_t$).
            Observe $\ell_t = c(s_t, w_t)$.
            $\hat{w}_t := 0$.
            $(\hat{x}_{t+1}, \hat{s}_{t+1}) := (\hat{x}_t, \hat{s}_t)$.
```

Figure 3.4: Algorithm for the bandit setting.

of all sets but one.[7] In the case of TSP, we can start with any tour σ that visits all the edges at least once and consider σ_e for each edge e which is the same as σ but traverses e an additional two times.

We present the algorithm for the bandit setting in Figure 3.4. We remark that our approach is essentially the same as previous approaches and can be used as a generic conversion from a black-box full-information online algorithm to a bandit algorithm. Previous approaches also worked in this manner, but the analysis depended on the specific bounds of the black-box algorithm in a way that, unfortunately, we cannot simply reference.

Theorem 3.4.2. *For $\alpha, \beta \geq 1$, integer $T \geq 0$ and any $w_1, \ldots, w_T$, given an α-approximation oracle and a β-BS, the algorithm in Figure 3.4 with $\eta = \frac{(\alpha+1)R}{D\sqrt{T}}$, $\delta = \eta n T^{-1/3}$, and $\gamma = (4\alpha\beta)^{2/3} n T^{-1/3}$ achieves an expected α-regret bound in the bandit setting of*

$$\mathrm{E}[\alpha\text{-}\textit{regret}] \leq 7n(\alpha\beta)^{2/3}T^{-1/3}.$$

The conversion from full-information to bandit is similar to other conversions [4, 35, 92]. Note that in the description of the algorithm, s_t is what is played at step t. Also note that $\hat{x}_{t+1}$ may be viewed as an approximate projection of $\hat{x}_t$ when it is generated in exploitation steps as well as in exploration steps, since $\hat{x}_t \in \Pi^{\delta}_{\alpha J}(\hat{x}_t - \eta\hat{w}_t)$ for $\hat{w}_t = 0$. We first prove a lemma:

Lemma 3.4.3. *Let $J \subseteq \mathbb{R}^n$ be a convex set such that $\forall \hat{x} \in J$, $\|\hat{x}\| \leq R$. Let $w_1, \ldots, w_T \in \mathbb{R}^n$ be an arbitrary sequence and $\hat{w}_1, \ldots, \hat{w}_T$ be a sequence of random variables such that*

[7] If any of these is not a cover, that set must be mandatory in any cover and we can simplify the problem. If this set of covers is not linearly independent, then we can reduce the dimensionality of the problem and use the fact that if T is a (possibly linearly dependent) β-BS for S and R is a γ-BS for T then R is a $(\gamma\beta|T|)$-BS for S.

$\mathrm{E}[\hat{w}_t|\hat{x}_1, \hat{w}_1, \ldots, \hat{x}_{t-1}, \hat{w}_{t-1}, \hat{x}_t] = w_t$ *and* $\mathrm{E}[\hat{w}_t^2] \leq D^2$*. Then, for any initial point* $\hat{x}_1 \in J$ *and any sequence* $\hat{x}_1, \hat{x}_2, \ldots$ *such that* $\hat{x}_{t+1} \in \Pi^{\delta}_{\alpha J}(\hat{x}_t - \eta \hat{w}_t)$*,*

$$\mathrm{E}\left[\sum_{t=1}^{T} \hat{x}_t \cdot w_t\right] - \alpha \min_{x \in J} \sum_{t=1}^{T} x \cdot w_t \leq \frac{(\alpha+1)^2 R^2}{2\eta} + T\frac{\delta}{\eta} + \frac{\eta}{2} D^2 T + 2\alpha R D\sqrt{T}.$$

Proof. By Lemma 3.3.3, we have that

$$\sum_{t=1}^{T} \hat{x}_t \cdot \hat{w}_t - \alpha \min_{x \in J} \sum_{t=1}^{T} x \cdot \hat{w}_t \leq \frac{(\alpha+1)^2 R^2}{2\eta} + T\frac{\delta}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \hat{w}_t^2.$$

Taking expectations of both sides gives

$$\sum_{t=1}^{T} \hat{x}_t \cdot w_t - \alpha \mathrm{E}\left[\min_{x \in J} \sum_{t=1}^{T} x \cdot \hat{w}_t\right] \leq \frac{(\alpha+1)^2 R^2}{2\eta} + T\frac{\delta}{\eta} + \frac{\eta}{2} D^2 T.$$

It thus suffices to show that

$$\mathrm{E}\left[\min_{x \in J} \sum_{t=1}^{T} x \cdot \hat{w}_t\right] \geq \min_{x \in J} \sum_{t=1}^{T} x \cdot w_t - 2RD\sqrt{T}. \tag{3.1}$$

Now, for any $x \in J$,

$$\begin{aligned} \left|\sum_{t=1}^{T} x \cdot (\hat{w}_t - w_t)\right| &\leq |x| \left|\sum_{t=1}^{T} \hat{w}_t - w_t\right| \\ &\leq R \left|\sum_{t=1}^{T} \hat{w}_t - w_t\right|. \end{aligned} \tag{3.2}$$

This gives us a means of upper-bounding the difference between the minima. Namely,

$$\begin{aligned} \mathrm{E}\left[\ \left|\sum_{t=1}^{T} \hat{w}_t - w_t\right|\ \right]^2 &\leq \mathrm{E}\left[\left(\sum_{t=1}^{T} \hat{w}_t - w_t\right)^2\right] \\ &= \sum_{t=1}^{T} \mathrm{E}\left[(\hat{w}_t - w_t)^2\right]. \end{aligned} \tag{3.3}$$

The last equality follows from the fact that

$$\mathrm{E}[(\hat{w}_{t_1} - w_{t_1})(\hat{w}_{t_2} - w_{t_2})] = 0$$

for $t_1 < t_2$, which follows from the martingale-like assumption that $\mathrm{E}[\hat{w}_{t_2} - w_{t_2}|\hat{w}_{t_1}, w_{t_1}] = 0$. Finally,

$$\begin{aligned} \mathrm{E}[(\hat{w}_t - w_t)^2] &\leq \mathrm{E}[\hat{w}_t^2 + 2\|\hat{w}_t\|\|w_t\| + w_t^2] \\ &\leq D^2 + 2D^2 + D^2 \\ &= 4D^2. \end{aligned}$$

In the above we have used the facts that $\mathrm{E}[|\hat{w}_t|]^2 \leq \mathrm{E}[\hat{w}_t^2] \leq D^2$ and $\|w_t\|^2 = \mathrm{E}[\hat{w}_t]^2 \leq \mathrm{E}[\hat{w}_t^2] \leq D^2$. Hence, we have that the quantity in (3.3) is upper bounded by $4TD^2$, which, together with (3.2), establishes (3.1). □

of Theorem 3.4.2. We remark that the parameter γ in the statement of the theorem may be larger than 1, but in this case the regret bound is greater than 1 and hence holds for any algorithm.

Note that in the conversion algorithm the expected value of $\hat{w}_t$ is w_t, and this is true conditioned on all previous information as well as $\hat{x}_t$. Since Lemma 3.3.6 implies $\hat{x}_{t+1} \in \Pi^{\delta}_{\alpha J}(\hat{x}_t - \eta\hat{w}_t)$, we can apply Lemma 3.4.3 to the sequence $\hat{x}_t$. This gives

$$\sum_{t=1}^{T} \mathrm{E}[\hat{x}_t \cdot w_t] - \alpha \min_{x \in J} \sum_{t=1}^{T} x \cdot w_t \leq \frac{(\alpha+1)^2 R^2}{2\eta} + T\frac{\delta}{\eta} + \frac{\eta}{2} D^2 T + 2\alpha R D\sqrt{T}.$$

To apply the lemma, we use the bound $D = n\gamma^{-1/2}$. This holds because $\ell_t \in [0,1]$, so $\mathrm{E}[\hat{w}_t^2] \leq \gamma(n\ell_t/\gamma)^2 + (1-\gamma)0 \leq n^2/\gamma$. Also, we use the bound of $R = \beta\sqrt{n}$. Hence we choose $\eta = \frac{(\alpha+1)R}{D\sqrt{T}}$ and $\delta = \eta n T^{-1/3}$, which simplifies the above equation to

$$\begin{aligned}\sum_{t=1}^{T} \mathrm{E}[\hat{x}_t \cdot w_t] - \alpha \min_{x \in J} \sum_{t=1}^{T} x \cdot w_t &\leq (\alpha+1)RD\sqrt{T} + nT^{2/3} + 2\alpha R D\sqrt{T} \\ &\leq 4\alpha R D\sqrt{T} + nT^{2/3}.\end{aligned}$$

Substituting the values of D and R gives an upper bound of $4\alpha\beta n^{3/2}\gamma^{-1/2}\sqrt{T} + T\frac{\delta}{\eta}$.

Next, as in the analysis of the full-information algorithm, $\mathrm{E}[\Phi(\hat{s}_t)]$ dominates $\mathrm{E}[\hat{x}_t]$ by Lemma 3.3.6. Thus,

$$\sum_{t=1}^{T} \mathrm{E}[c(\hat{s}_t, \cdot w_t)] - \alpha \min_{x \in J} \sum_{t=1}^{T} x \cdot w_t \leq 4\alpha\beta n^{3/2}\gamma^{-1/2}\sqrt{T} + nT^{2/3}.$$

Finally, we have that $\mathrm{E}[c(s_t, w_t)] \leq \mathrm{E}[c(\hat{s}_t, w_t)] + \gamma$ because with probability $1-\gamma$, $\hat{s}_t = s_t$ and in the remaining case the cost is in $[0,1]$. Putting these together implies

$$\sum_{t=1}^{T} \mathrm{E}[c(s_t, \cdot w_t)] - \alpha \min_{x \in J} \sum_{t=1}^{T} x \cdot w_t \leq 4\alpha\beta n^{3/2}\gamma^{-1/2}\sqrt{T} + nT^{2/3} + \gamma T.$$

Choosing $\gamma = (4\alpha\beta)^{2/3} n T^{-1/3}$ (note that if this quantity is larger than 1, then the regret bound in the theorem is trivial) gives a bound of $2n(4\alpha\beta T)^{2/3} + nT^{2/3} \leq 7n(\alpha\beta T)^{2/3}$ as in the theorem. □

3.4.1 Difficulty of the black-box reduction

We now point out that it is impossible to solve the bandit problem with general algorithms (approximation or exact) without an exploration basis (that is, if our only access to $\mathcal{S}$ is through a black-box optimization oracle). The counterexample is randomized. Denote by $w[1]$ the first coordinate of a vector w. We will take

$$\mathcal{W} = \{w \in \mathbb{R}^n \mid w[1] \in [0,1] \text{ and } \|w\|^2 \le 2(w[1])^2\}.$$

The set $\mathcal{S}$ will consist of two points: $s = (1/2, 0, \ldots, 0)$ as well as a second point $s' = (1, 0, \ldots, 0) - u$ where $\|u\| = 1$ and $u[1] = 0$. The mapping Φ is the identity mapping. The cost sequence will be constant $w_t = (1, 0, \ldots, 0) + u$. Hence $c(s, w_t) = 1/2$ while $c(s', w_t) = 0$. Now, suppose we as algorithm designers know that this is the setup but u is chosen uniformly at random from the set of unit vectors with $u[1] = 0$.

Observation 3.4.4. *For any bandit algorithm that makes k calls to black-box optimization oracle A, any $\alpha \ge 0$, with probability $1 - ke^{-0.1n}$ over u, the algorithm has α-regret $1/2$ on a sequence of arbitrary length.*

Proof. No information is conveyed by the costs returned in the bandit setup of our example—they are always 1/2 if s' has not been discovered, while the minimal cost is 0. Thus the algorithm must find some $w \in \mathcal{W}$ such that $c(s, w) > c(s', w)$ (whence an exact optimization algorithm must return s'), but is restricted to querying $w \in \mathcal{W}$. Without loss of generality, we can scale w so that $w[1] = 1$ and $\|w\| \le 2$. Hence, we can write $w = (1, 0, 0 \ldots, 0) + v$ where $v[1] = 0$ and $\|v\| \le 1$. In this case, $w \cdot s = 1/2$, while $w \cdot s' = 1 - u \cdot v$. For u a random unit vector and any fixed $\|v\| \le 1$, it is known that $\Pr[u \cdot v \ge 1/2]$ is exponentially small in n. A very loose bound can be seen directly, since for a ball of dimension n, this probability is

$$\begin{aligned}
\frac{\int_{1/2}^{1} (\sqrt{1-x^2})^{n-2} dx}{\int_{-1}^{1} (\sqrt{1-x^2})^{n-2} dx} &\le \frac{\int_{1/2}^{1} (3/4)^{\frac{n-2}{2}} dx}{\int_{-1/\sqrt{n}}^{1/\sqrt{n}} (1-n^{-1})^{\frac{n-2}{2}} dx} \\
&\le \frac{\sqrt{ne}}{2} \left(\frac{3}{4}\right)^{\frac{n}{2}-1},
\end{aligned}$$

which is $O(e^{-0.1n})$. □

3.5 Conclusions

In this chapter, we present a reduction converting approximate offline linear optimization problems into approximate online sequential linear optimization problems that holds for *any* approximation algorithm, in both in the full-information setting and the bandit setting.

Our algorithm can be viewed as an analog to Hannan's algorithm for playing repeated games against an unknown opponent. In our case, however, we cannot compute best responses but only approximately best responses.

The problem of obtaining similar results for interesting classes of non-linear optimization problems remains open.

Chapter 4

Convergence to Nash Equilibria in Routing Games

4.1 Introduction

One specific setting where efficient regret minimization algorithms exist is online routing. Given a graph $G = (V, E)$ and two distinguished nodes v_{start} and v_{end}, the game for an individual player is defined as follows. At each time step t, the player's algorithm chooses a path P_t from v_{start} to v_{end}, and simultaneously an adversary (or nature) chooses a set of edge costs $\{c_e^t\}_{e \in E}$. The edge costs are then revealed and the player pays the cost of its path. Even though the number of possible paths can be exponential in the size of the graph, because this can be cast as a linear optimization problem whose offline version can be solved exactly in polynomial time (the player selects an indicator vector showing which graph edges are included in her path, and the weight vector that shows up indicates the cost of each edge), prior work can be used to minimize regret in this setting. For example, the algorithms of Kalai and Vempala [81] and Zinkevich [122] achieve running time and convergence rates (to the cost of the best fixed path in hindsight) which are polynomial in the size of the graph and the maximum edge cost. Moreover, a number of extensions [4, 92] have shown how these algorithms can be applied even to the "bandit" setting where only the cost of edges actually traversed (or even just the total cost of P_t) is revealed to the algorithm at the end of each time step t.

In this chapter we consider the question: if all players in a routing game use no-regret algorithms to choose their paths each day, what can we say about the overall behavior of the system? In particular, the no-regret property (also called Hannan Consistency) can be viewed as a natural *definition* of well-reasoned self-interested behavior over time. Thus, if all players are adapting their behavior in such a way, can we say that the system as a whole will approach Nash equilibrium? Our main result is that in the Wardrop setting of multicommodity flow and infinitesimal agents, the flows will approach equilibrium in the sense that a $1-\epsilon$ fraction of the daily flows will have the property that at most an ϵ fraction of the agents in them have more than an ϵ incentive to deviate from their chosen path, where ϵ approaches 0 at a rate that depends polynomially on the size of the graph, the regret-bounds of the algorithms, and the maximum slope of any latency function.[1]

Moreover, we show that the one new parameter—the dependence on slope—is necessary. In addition, we give stronger results for special cases such as the case of n parallel links and also consider the finite-size (non-infinitesimal) load-balancing model of Azar [8]. Our results for nonatomic players also hold for a more general class of games called congestion games, although efficient regret-minimizing algorithms need not exist for the most general of these games.

One way our result can be viewed is as follows. No-regret algorithms are very compelling from the point of view of individuals: if you use a no-regret algorithm to drive to work each day, you will get a good guarantee on your performance no matter what is causing congestion (other drivers, road construction, or unpredictable events). But it would be a shame if, were everyone to use such an algorithm, this produced globally unstable behavior. Our results imply that in the Wardrop routing model, so long as edge latencies have bounded slope, we can view Nash equilibria as not just a stable steady-state or the result of adaptive procedures specifically designed to find them, but in fact as the inevitable result of individual selfishly adaptive behavior by agents that do not necessarily know (or care) what policies other agents are using. Moreover, our results do not in fact require that users follow strategies that are no-regret in the worst-case, as long as their behavior satisfies the no-regret property over the sequence of flows actually observed.

4.1.1 Regret and Nash equilibria

At first glance, a result of this form seems that it should be obvious given that a Nash equilibrium is precisely a set of strategies (pure or mixed) that are all no-regret with respect to each other. Thus if the learning algorithms settle at all, they will have to settle at a Nash equilibrium. In fact, for *zero-sum* games, no-regret algorithms when played against each other will approach a minimax optimal solution [60]. However, it is known that even in small 2-player *general-sum* games, no-regret algorithms need not approach a Nash equilibrium and can instead cycle, achieving performance substantially worse than any Nash equilibrium for all players. Indeed simple examples are known where standard algorithms will have this property with arbitrarily high probability [123].

4.2 Preliminaries

4.2.1 Nonatomic congestion games

Let E be a finite ground set of elements (we refer to them as *edges*). There are k *player types* $1, 2, \ldots, k$, and each player type i has an associated set of feasible paths $\mathcal{P}_i$, where $\mathcal{P}_i$ is a set

[1] A more traditional notion of approximate Nash equilibrium requires that *no* player will have more than ϵ incentive to deviate from her strategy. However, one cannot hope to achieve such a guarantee using arbitrary no-regret algorithms, since such algorithms allow players to occasionally try bad paths, and in fact such experimentation is even necessary in bandit settings. For the same reason, one cannot hope that *all* days will be approximate-Nash. Finally, our guarantee may make one worry that some users could always do badly, falling in the ϵ minority on every day, but as we discuss in §4.5, the no-regret property can be used to further show that no player experiences many days in which her expected cost is much worse than the best path available on that day.

of subsets of E. Elements of $\mathcal{P}_i$ are called *paths* or *strategies*. For example, player type i might correspond to players who want to travel from node u_i to node v_i in some underlying graph G, and $\mathcal{P}_i$ might be the set of all u_i-v_i paths. The continuum A_i of agents of type i is represented by the interval $[0, a_i]$, endowed with Lebesgue measure. We restrict $\sum_{i=1}^{k} a_i = 1$, so there is a total of one unit of flow. Each edge $e \in E$ has an associated traffic-dependent, non-negative, continuous, non-decreasing *latency* function ℓ_e. A *nonatomic congestion game* is defined by $(E, \ell, \mathcal{P}, A)$.

A *flow* determines a path for each player: $f_i : A_i \rightarrow \mathcal{Q}_i$ where $\mathcal{Q}_i$ is the set of $0/1$ vectors in $\mathcal{P}_i$ with exactly one 1. We write $f = (\int_{A_1} f_1, \ldots, \int_{A_k} f_k)$, where $\int_{A_i} f_i$ reflects the amount of flow of type i on each path in $\mathcal{P}_i$. A flow thus induces a distribution over paths, which we write for a path P in $\mathcal{P}_i$ as $f_P = (f_i)^P$ for P of type i. Thus, $\sum_{P \in \mathcal{P}_i} f_P = a_i$ for all i, and f_P is the measure of the set of players selecting path P. Each flow induces a unique flow on edges such that the flow f_e on an edge e has the property $f_e = \sum_{P : e \in P} f_P$. The latency of a path P given a flow f is $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$, i.e., the sum of the latencies of the edges in the path, given that flow. The cost α_i incurred by a player of type i is simply the latency of the path she plays.

The social utility function we consider is the total cost incurred by a flow: $\gamma(f) = \sum_{e \in E} \ell_e(f_e)$.

We define $|E| = m$ and write n for the number of edges in the largest path in $\mathcal{P}$. We will assume all edge latency functions have range $[0, 1]$, so the latency of a path is always between 0 and n. Let $f^1, f^2, \ldots, f^T$ denote a series of flows from time 1 up to time T. We use $\hat{f}$ to denote the time-average flow, i.e., $\hat{f}_e = \frac{1}{T} \sum_{t=1}^{T} f_e^t$.

A flow f is at *Nash equilibrium* if no user would prefer to reroute her traffic, given the existing flow.

Remark 4.2.1. *Network games are a special case of nonatomic congestion games, where there is an underlying graph G and players of type i have a start node u_i and a destination node v_i, and $\mathcal{P}_i$ is the set of all u_i-v_i paths.*

It is useful to note that in this domain, the flows at equilibrium are those for which all flow-carrying paths for a particular player type have the same latency, and this latency is minimal among all paths for players of that type. In addition, given our assumption that all latency functions are continuous and non-decreasing, one can prove the existence of Nash equilibria:

Proposition 4.2.2. *(Schmeidler [112], generalization of Beckman et al. [12]) Every nonatomic congestion game admits a flow at equilibrium.*

In addition, for any nonatomic congestion game, there is a unique equilibrium cost:

Proposition 4.2.3. *(Milchtaich [96], generalization of Beckman et al. [12]) Distinct equilibria for a nonatomic congestion game have equal social cost.*

In this chapter, excluding §4.7, we consider infinitesimal users using a finite number of different algorithms; in this setting, we can get rid of the expectation in the formulation of our low-regret assumption. In particular, if each user is running a no-regret algorithm, then the average regret over users also approaches 0. Thus, since all players have bounded per-timestep cost, applying the strong law of large numbers, we can make the following assumption:

Assumption 4.2.4. *The series of flows $f^1, f^2, \ldots$ satisfies*

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{e \in E} \ell_e(f_e^t) f_e^t \leq R(T) + \frac{1}{T} \sum_{i=1}^{k} a_i \min_{P \in \mathcal{P}_i} \sum_{t=1}^{T} \sum_{e \in P} \ell_e(f_e^t)$$

where $R(T) \rightarrow 0$ as $T \rightarrow \infty$. The function $R(T)$ may depend on the size of the network and its maximum possible latency. We then define T_ϵ as the number of time steps required to get $R(T) \leq \epsilon$.

4.2.2 Approaching Nash Equilibria

We now need to specify in what sense flow will be approaching a Nash equilibrium. The first notion one might consider is the L_1 distance from some true Nash flow. However, if some edges have nearly-flat latency functions, it is possible for a flow to have regret near 0 and yet still be far in L_1 distance from a true Nash flow. A second natural notion would be to say that the flow f has the property that no user has cost much more than the cheapest path given f. However, notice that the no-regret property allows users to occasionally take long paths, so long as they perform well on average (and in fact algorithms for the bandit problem will have exploration steps that do just that [4, 92]). So, one cannot expect that on any time step *all* users are taking cheap paths.

Instead, we require that *most* users be taking a nearly-cheapest path given f. Specifically,

Definition 4.2.5. *A flow f is at ϵ-Nash equilibrium if the average cost under this flow is within ϵ of the minimum cost paths under this flow, i.e. $C(f) - \sum_{i=1}^{k} a_i \min_{P \in \mathcal{P}_i} \sum_{e \in P} \ell_e(f_e) \leq \epsilon$.*

Note that Definition 4.2.5 implies that at most a $\sqrt{\epsilon}$ fraction of traffic can have more than a $\sqrt{\epsilon}$ incentive to deviate from their path, and as a result is very similar to the definition of (ϵ, δ)-Nash equilibria in [52]. We also are able to show that one can apply price-of-anarchy results to ϵ-Nash flows; we discuss this in §4.6.

We will begin by focusing on the *time-average* flow $\hat{f}$, showing that for no-regret algorithms, this flow is approaching equilibrium. That is, for a given T_ϵ we will give bounds on the number of time steps before $\hat{f}$ is ϵ-Nash. After analyzing $\hat{f}$, we then extend our analysis to show that in fact for *most* time steps t, the flow f^t itself is ϵ-Nash. To achieve bounds of this form, which we show in § 4.5, we will however need to lose an additional factor polynomial in the size of the graph. Again, we cannot hope to say that f^t is ϵ-Nash for *all* (sufficiently large) time-steps t, because no-regret algorithms may occasionally take long paths, and an "adversarial" set of such algorithms may occasionally all take long paths at the same time.

4.2.3 Dependence on slope

Our convergence rates will depend on the maximum slope s allowed for any latency function. To see why this is necessary, consider the case of a routing game with two parallel links, where one edge has latency 0 up to a load of $1/3$ and then rises immediately to 1, and the other edge has latency 0 up to a load of $2/3$ and then rises directly to 1. In this case the Nash cost is 0, and moreover for *any* flow f' we have $\min_{P \in \mathcal{P}} \sum_{e \in P} \ell_e(f'_e) = 0$. Thus, the only way f' can be ϵ-Nash is for it to actually have low cost, which means the algorithm must precisely be at a $1/3$-$2/3$ split. If players use no-regret algorithms, traffic will instead oscillate, each edge having cost 1 on about half the days and each player incurring cost 1 on not much more than half the days (and thus not having much regret). However, none of the daily flows will be better than $\frac{1}{3}$-Nash, because on each day, the cost of the flow f is at least 1/3.

4.3 Infinitesimal Users: Linear Latency Functions

We begin as a warm-up with the easiest case, infinitesimal users and linear latency functions, which simplifies many of the arguments. In particular, for linear latency functions, the latency of any edge under the time-average flow $\hat{f}$ is guaranteed to be equal to the average latency of that edge over time, i.e. $\ell_e(\hat{f}_e) = \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)$ for all e.

Theorem 4.3.1. *Suppose the latency functions are linear. Then for $T \geq T_\epsilon$, the average flow $\hat{f}$ is ϵ-Nash, i.e.*

$$C(\hat{f}) \leq \epsilon + \sum_i a_i \min_{P \in \mathcal{P}_i} \sum_{e \in P} \ell_e(\hat{f}_e).$$

Proof. From the linearity of the latency functions, we have for all e, $\ell_e(\hat{f}_e) = \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)$. Since $\ell_e(f_e^t) f_e^t$ is a convex function of the flow, this implies

$$\ell_e(\hat{f}_e)\hat{f}_e \leq \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t) f_e^t.$$

Summing over all e, we have

$$\begin{array}{rcll} C(\hat{f}) & \leq & \frac{1}{T}\sum_{t=1}^{T} C(f^t) & \\ & \leq & \epsilon + \sum_i a_i \min_{P \in \mathcal{P}_i} \frac{1}{T}\sum_{t=1}^{T} \sum_{e \in P} \ell_e(f_e^t) & \text{(by Assumption 4.2.4)} \\ & = & \epsilon + \sum_i a_i \min_{P \in \mathcal{P}_i} \sum_{e \in P} \ell_e(\hat{f}_e). & \text{(by linearity)} \end{array}$$

□

Corollary 4.3.2. *Assume that all latency functions are linear. In general routing games, if all agents use the Kalai-Vempala algorithm [81], the average flow converges to an ϵ-Nash equilibrium at $T_\epsilon = O(\frac{mn \log n}{\epsilon^2})$. On networks consisting of two nodes and m parallel links, if all agents use optimized "combining expert advice"-style algorithms (with each edge an expert), the average flow converges to an ϵ-Nash equilibrium at $T_\epsilon = O(\frac{\log m}{\epsilon^2})$.*

Note that we not only proved that the average flow approaches an ϵ-Nash equilibrium, but as an intermediate step in our proof we showed that *actual* average cost incurred by a user of type i is at most ϵ worse than the best path in $\mathcal{P}_i$ in the average flow.

4.4 Infinitesimal Users: General Latency Functions

The case of general latency functions is more complicated because the first and third transitions in the proof above do not apply. Here, the additive term depends on the maximum slope of any latency function.

Theorem 4.4.1. *Let $\epsilon' = \epsilon + 2\sqrt{s\epsilon n}$. Then for general functions with maximum slope s, for $T \geq T_\epsilon$, the time-average flow is ϵ'-Nash, that is,*

$$\sum_{e \in E} \ell_e(\hat{f}_e)\hat{f}_e \leq \epsilon + 2\sqrt{s\epsilon n} + \sum_i a_i \min_{P \in \mathcal{P}_i} \sum_{e \in P} \ell_e(\hat{f}_e).$$

Before giving the proof, we list several quantities we will need to relate:

$$\sum_{e\in E} \ell_e(\hat{f}_e)\hat{f}_e \qquad \text{(cost of } \hat{f}\text{)} \tag{4.1}$$

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} \ell_e(f_e^t)\hat{f}_e \qquad \text{(“cost of } \hat{f} \text{ in hindsight”)} \tag{4.2}$$

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} \ell_e(f_e^t)f_e^t \qquad \text{(avg cost of flows up to time } T\text{)} \tag{4.3}$$

$$\sum_i a_i \min_{P\in\mathcal{P}_i} \sum_{e\in P} \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t) \qquad \text{(cost of best path in hindsight)} \tag{4.4}$$

$$\sum_i a_i \min_{P\in\mathcal{P}_i} \sum_{e\in P} \ell_e(\hat{f}_e) \qquad \text{(cost of best path given } \hat{f}\text{)} \tag{4.5}$$

Our goal in proving Theorem 4.4.1 is to show that (4.1) is not too much greater than (4.5). We will prove this as follows. We know that $(4.3) \leq \epsilon + (4.4)$ by the no-regret property and that $(4.2) \leq (4.3)$ by the fact that ℓ is non-decreasing. So, what remains to show is that (4.4) is not much greater than (4.5) and that (4.1) is not much greater than (4.2). We prove these in Lemmas 4.4.2 and 4.4.3 below.

Lemma 4.4.2. *For general latency functions with maximum slope s, $(4.4) \leq \sqrt{s\epsilon n} + (4.5)$.*

Proof. First, observe that, because our latency functions are non-decreasing, the average latency of an edge must be less than or equal to the latency of that edge as seen by a random user on a random day. That is, for all e,

$$\frac{1}{T}\hat{f}_e \sum_{t=1}^{T} \ell_e(f_e^t) \leq \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)f_e^t.$$

This can be shown by induction, using the fact that $f_e^a\ell_e(f_e^b) + f_e^b\ell_e(f_e^a) \leq f_e^a\ell_e(f_e^a) + f_e^b\ell_e(f_e^b)$ for any flows f_e^a, f_e^b. Define $\epsilon_e = \frac{1}{T}\sum_{t=1}^{T}\ell_e(f_e^t)f_e^t - \frac{1}{T}\hat{f}_e\sum_{t=1}^{T}\ell_e(f_e^t)$ to be the gap between the above two terms. Now, notice that the right-hand side of the above inequality, summed over all edges, is precisely quantity (4.3). By the no-regret property, this is at most ϵ larger than the time-average cost of the best paths in hindsight, which in turn is clearly at most the time-average cost of $\hat{f}$. Therefore, we have:

$$\begin{aligned}
\frac{1}{T}\hat{f}_e \sum_{t=1}^{T}\sum_{e\in E} \ell_e(f_e^t) &\leq \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} \ell_e(f_e^t)f_e^t \\
&\leq \epsilon + \frac{1}{T}\hat{f}_e \sum_{t=1}^{T}\sum_{e\in E} \ell_e(f_e^t).
\end{aligned}$$

That is, we have “sandwiched” the flow-average latency between the time-average latency and the time-average latency plus ϵ. This implies that for every edge e, its time-average cost must be close to its flow-average cost, namely,

$$\sum_{e \in E} \epsilon_e \leq \epsilon.$$

We now use this fact, together with the assumption of bounded slope, to show that edge latencies cannot be varying wildly over time. Specifically, we can rewrite the definition of ϵ_e as:

$$\epsilon_e = \frac{1}{T} \sum_{t=1}^{T} (\ell_e(f_e^t) - \ell_e(\hat{f}_e))(f_e^t - \hat{f}_e) \geq 0, \tag{4.6}$$

where we are using the fact that $\hat{f}_e = \frac{1}{T} \sum_{t=1}^{T} f_e^t$ and so $\frac{1}{T} \sum_{t=1}^{T} \ell_e(\hat{f}_e)(f_e^t - \hat{f}_e) = 0$.

From the bound on the maximum slope of any latency function, we know that $|f_e^t - \hat{f}_e| \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|/s$ and thus

$$|\ell_e(f_e^t) - \ell_e(\hat{f}_e)| \leq \sqrt{s \left(\ell_e(f_e^t) - \ell_e(\hat{f}_e)\right) \left(f_e^t - \hat{f}_e\right)}$$

for all e.

We then get

$$\frac{1}{T} \sum_{t=1}^{T} \left| \ell_e(f_e^t) - \ell_e(\hat{f}_e) \right| \leq \frac{\sqrt{s}}{T} \sum_{t=1}^{T} \sqrt{(\ell_e(f_e^t) - \ell_e(\hat{f}_e))(f_e^t - \hat{f}_e)}.$$

Using equation (4.6) above and the fact the square root is concave function, an application of the Cauchy-Schwartz inequality yields

$$\frac{1}{T} \sum_{t=1}^{T} \left| \ell_e(f_e^t) - \ell_e(\hat{f}_e) \right| \leq \sqrt{s \epsilon_e}. \tag{4.7}$$

Finally, let P_i^* be the best path of type i given $\hat{f}$. Summing equation (4.7) over the edges in P_i^*, and using the fact that $\sum_i a_i \sum_{e \in P_i^*} \sqrt{s \epsilon_e} \leq \sqrt{s \epsilon n}$, we have

$$(4.5) + \sqrt{s \epsilon n} \ \geq \ \sum_{e \in P^*} \frac{1}{T} \sum_{t=1}^{T} \ell_e(f_e^t) \ \geq \ (4.4),$$

as desired. □

Lemma 4.4.3. *For general latency functions with maximum slope s, (4.1) $\leq \sqrt{s \epsilon n}$ + (4.2).*

Proof. Equation (4.7) above directly gives us

$$(4.1) \leq \sum_{e \in E} \sqrt{s \epsilon_e} \hat{f}_e + (4.2).$$

An application of the Cauchy-Schwartz inequality then gives us

$$\left(\sum_{e\in E}\sqrt{s\epsilon_e}\hat{f}_e\right)^2 \le \left(\sum_{e\in E} s\epsilon_e\right)\left(\sum_{e\in E}\hat{f}_e^2\right).$$

Since $\hat{f}_e \le 1$ for all e, this is at most $\left(\sum_{e\in E} s\epsilon_e\right)\left(\sum_{e\in E}\hat{f}_e\right)$. Since $\sum_{e\in E}\hat{f}_e \le n$, this is at most $s\epsilon n$, and thus

$$\sum_{e\in E}\sqrt{s\epsilon_e}\hat{f}_e \le \sqrt{s\epsilon n},$$

which gives the desired result. □

Given the above lemmas we now present the proof of Theorem 4.4.1.

of Theorem 4.4.1. Since $(4.3) \le \epsilon + (4.4)$ by Assumption 4.2.4, and $(4.2) \le (4.3)$ by the fact that the latency functions are non-decreasing, we get

$$\begin{aligned}(4.1) &\le \sqrt{s\epsilon n} + (4.2)\\ &\le \sqrt{s\epsilon n} + (4.3)\\ &\le \epsilon + \sqrt{s\epsilon n} + (4.4)\\ &\le \epsilon + 2\sqrt{s\epsilon n} + (4.5)\end{aligned}$$

as desired. □

Corollary 4.4.4. *Let $\epsilon' = \epsilon + 2\sqrt{s\epsilon n}$. Assume that all latency functions are positive, non-decreasing, and continuous, with maximum slope s. In general routing games, if all agents use the Kalai-Vempala algorithm [81], the average flow converges to an ϵ'-Nash equilibrium at $T_\epsilon = O(\frac{mn\log n}{\epsilon^2}) = O(\frac{mn^3s^2\log n}{\epsilon'^4})$. On networks consisting of two nodes and m parallel links, if all agents use optimized "combining expert advice"-style algorithms, the average flow converges to an ϵ'-Nash equilibrium at $T_\epsilon = O(\frac{\log m}{\epsilon^2}) = O(\frac{n^2s^2\log m}{\epsilon'^4})$.*

Once again we remark that not only have we proved that the average flow approaches ϵ'-Nash equilibrium, but as an intermediate step in our proof we showed that *actual* average cost obtained by the users is at most ϵ' worse than the best path in the average flow.

4.5 Infinitesimal Users: Bounds on Most Timesteps

Here we present results applicable to general graphs and general functions showing that on *most* time steps t, the flow f^t will be at ϵ-Nash equilibrium.

Theorem 4.5.1. *In general routing games with general latency functions with maximum slope s, for all but a $(ms^{1/4}\epsilon^{1/4})$ fraction of time steps up to time T_ϵ, f^t is a $(\epsilon + 2\sqrt{s\epsilon n} + 2m^{3/4}s^{1/4}\epsilon^{1/4})$-Nash flow. We can rewrite this as: for all but an ϵ' fraction of time steps up to T_ϵ, f^t is an ϵ'-Nash flow for $\epsilon = \Omega\left(\frac{\epsilon'^4}{sm^4+s^2n^2}\right)$.*

Proof. Based on equation (4.6),

$$\sqrt{s\epsilon_e} \geq \frac{1}{T}\sum_{t=1}^{T} |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|$$

for all edges. Thus, for all edges, for all but $s^{1/4}\epsilon_e^{1/4}$ of the time steps,

$$s^{1/4}\epsilon_e^{1/4} \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|.$$

Using a union bound over edges, this implies that on all but a $ms^{1/4}\epsilon^{1/4}$ fraction of the time steps, *all* edges have

$$s^{1/4}\epsilon_e^{1/4} \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|.$$

From this, it follows directly that on most time steps, the cost of the best path given f^t differs from the cost of the best path given $\hat{f}$ by at most $n^{3/4}s^{1/4}\epsilon^{1/4}$. Also on most time steps, the cost incurred by flow f^t differs from the cost incurred by flow $\hat{f}$ by at most $m^{3/4}s^{1/4}\epsilon^{1/4}$. Thus since $\hat{f}$ is an ($\epsilon + 2\sqrt{s\epsilon n}$)-Nash equilibrium, f^t is an ($\epsilon + 2\sqrt{s\epsilon n} + 2m^{3/4}s^{1/4}\epsilon^{1/4}$)-Nash equilibrium on all but a $ms^{1/4}\epsilon^{1/4}$ fraction of time steps. □

Corollary 4.5.2. *In general routing games with general latency functions with maximum slope* s*, for all but a* $(ms^{1/4}\epsilon^{1/4})$ *fraction of time steps up to time* $T = T_\epsilon$*, the expected average cost* $\frac{1}{T}\sum_{t=1}^{T} c^t$ *incurred by any user is at most* $(\epsilon + 2\sqrt{s\epsilon n} + m^{3/4}s^{1/4}\epsilon^{1/4})$ *worse than the cost of the best path on that time step.*

Proof. From the proof of Theorem 4.5.1 we see that on most days, the cost of the best path given the flow for that day is within $m^{3/4}s^{1/4}\epsilon^{1/4}$ of the cost of the best path given $\hat{f}$, which is at most $2\sqrt{s\epsilon n}$ worse than the cost of the best path in hindsight. Combining this with the no-regret property achieved by each user gives the desired result. □

This demonstrates that no-regret algorithms are a reasonable, stable response in a network setting: if a player knows that all other players are using no-regret algorithms, there is no strategy that will significantly improve her expected cost on more than a small fraction of days. By using a no-regret algorithm, she gets the guarantee that on most time steps her expected cost is within some epsilon of the cost of the best path given the flow for that day.

4.6 Regret Minimization and the Price of Anarchy

In this section, we relate the costs incurred by regret-minimizing players in a single-commodity congestion game to the cost of the social optimum. We approach this problem in two ways: First, we show that any ϵ-Nash equilibrium in a single-commodity congestion game is closely related to a true Nash equilibrium in a related congestion game. This is an interesting property of approximate equilibria, and further allows us to apply Price of Anarchy results for the congestion game to the regret-minimizing players in the original game. In our second result in this section, we give an argument paralleling that of Roughgarden and Tardos [109] that directly relates the costs of multi-commodity regret-minimizing users to the cost of the social optimum.

Theorem 4.6.1. *If f is an ϵ-Nash equilibrium flow for a single-commodity nonatomic congestion game Γ, then*

$$C(f) \leq \frac{\rho}{1-\sqrt{\epsilon}} \left(C(OPT) + s\sqrt{\epsilon}n + \sqrt{\epsilon} + \epsilon \right),$$

where OPT is the minimum cost flow and ρ is the price of anarchy in a related congestion game Γ' with the same class of latency functions as Γ but with additive offsets.

For example, Theorem 4.6.1 implies that for linear latency functions of slope less than or equal to one, an ϵ-Nash flow f will have cost at most $\frac{4/3}{1-\sqrt{\epsilon}}(C(OPT) + \sqrt{\epsilon}(n+1) + \epsilon)$. Note that for regret minimizing players, Theorem 4.6.3 below improves this to $\frac{4}{3}C(OPT) + \epsilon$.

The proof idea for this theorem is as follows: For every nonatomic congestion game Γ and flow f at ϵ-Nash equilibrium on Γ, there exists a nonatomic congestion game Γ' that approximates Γ and a flow f' that approximates f such that: (a) f' is a Nash flow on Γ', (b) the cost of f' on Γ' is close to the cost of f on Γ, and (c) the cost of the optimal flow on Γ' is close to the cost of the optimal flow on Γ. These approximations allow one to apply price-of-anarchy results from f' and Γ' to f and Γ.

Proof. Note that since f is a single-commodity flow at ϵ-Nash equilibrium on Γ, then at most a $\sqrt{\epsilon}$ fraction of users are experiencing costs more than $\sqrt{\epsilon}$ worse than the cost of the best path given f; denote by min the cost of this shortest path given f. We can modify Γ to Γ_2 to embed the costs associated with these "meandering" users such that the costs experienced by the remaining users do not change. Call the non-meandering users f_2.

Note that $C(f \text{ on } \Gamma) - min \leq \epsilon$, since f is at an ϵ-Nash equilibrium. Also, the total costs experienced by the meandering users are at most $C(f \text{ on } \Gamma) - (1-\sqrt{\epsilon})min$; that is, every non-meandering user experiences cost at least min, since there is no cheaper path available. This is in turn at most $\epsilon + \sqrt{\epsilon}min \leq \epsilon + \sqrt{\epsilon}C(f \text{ on } \Gamma)$.

We now construct an alternate congestion game Γ_3 (not necessarily a routing game, even if the original game was a routing game) such that f_2 interpreted on Γ_3 is a Nash equilibrium. To do this, we create a new edge and include that edge in every allowable path . We can now assign cost to this new "entry edge" to cause the minimum cost of any available path to be equal to the cost of the worst flow-carrying path in f_2 on Γ_2. The maximum cost we need to assign in order to achieve this is $\sqrt{\epsilon}$, since we already removed all users paying more than $\sqrt{\epsilon}$ plus the cost of the best path available to them. Thus $C(f_2 \text{ on } \Gamma_2) \leq C(f_2 \text{ interpreted on } \Gamma_3)$, so we have

$$C(f \text{ on } \Gamma) \leq \frac{1}{1-\sqrt{\epsilon}} \left(C(f_2 \text{ interpreted on } \Gamma_3) + \epsilon \right).$$

Define ρ to be the price of anarchy of the new congestion game Γ_3 when played with up to one unit of flow. Thus, defining $OPT_\alpha(H)$ to be the min-cost flow of size α in game H, we have

$$C(f \text{ on } \Gamma) \leq \frac{\rho}{1-\sqrt{\epsilon}} \left(C(OPT_{1-\sqrt{\epsilon}}(\Gamma_3)) + \epsilon \right).$$

Since we added at most $\sqrt{\epsilon}$ to the cost of any solution in going from Γ_2 to Γ_3, this gives

$$C(f \text{ on } \Gamma) \leq \frac{\rho}{1-\sqrt{\epsilon}} \left(C(OPT_{1-\sqrt{\epsilon}}(\Gamma_3) \text{ interpreted on } \Gamma_2) + \sqrt{\epsilon} + \epsilon \right),$$

and since $OPT_{1-\sqrt{\epsilon}}(\Gamma_2)$ is the min-cost flow of size $(1-\sqrt{\epsilon})$ on Γ_2,

$$C(f \text{ on } \Gamma) \leq \frac{\rho}{1-\sqrt{\epsilon}} \left(C(OPT_{1-\sqrt{\epsilon}}(\Gamma_2)) + \sqrt{\epsilon} + \epsilon \right),$$

We now must quantify the amount by which the cost of $OPT_{1-\sqrt{\epsilon}}$ on Γ_2 could exceed the cost of OPT_1 on Γ. Since the cost of any edge in Γ_2 is at most $s\sqrt{\epsilon}$ more than the cost of that edge in Γ, this gives

$$C(f \text{ on } \Gamma) \leq \frac{\rho}{1-\sqrt{\epsilon}} \left(C(OPT) + s\sqrt{\epsilon}n + \sqrt{\epsilon} + \epsilon \right).$$

□

In particular, when all latency functions are linear, we can apply results of Roughgarden and Tardos bounding the price of anarchy in a congestion game with linear latency functions by $4/3$ [109].

We can also directly characterize the costs incurred by regret-minimizing players without going through the intermediate step of analyzing ϵ-Nash flows by arguing from scratch paralleling the Price of Anarchy proofs of Roughgarden and Tardos [109].

Definition 4.6.2. *Let $\mathcal{L}$ be the set of cost functions used by a nonatomic congestion game, with all $\ell(\xi)\xi$ convex on $[0,\infty)$. For a nonzero cost function $\ell \in \mathcal{L}$, we define $\alpha(\ell)$ by*

$$\alpha(\ell) = \sup_{n>0:\ell(n)>0} \left[\lambda\mu + (1-\lambda)\right]^{-1}$$

where the marginal social cost $\ell_e^*(\xi) = \ell_e(\xi) + \xi \cdot \ell_e'(\xi)$, $\lambda \in [0,1]$ satisfies $\ell^*(\lambda n) = \ell(n)$, and $\mu = \ell(\lambda n)/\ell(n) \in [0,1]$. We define $\alpha(\mathcal{L})$ by

$$\alpha(\mathcal{L}) = \sup_{0 \neq \ell \in \mathcal{L}} \alpha(\ell).$$

Theorem 4.6.3. *If Γ is a nonatomic congestion game with cost functions $\mathcal{L}$ with all $\ell(\xi)\xi$ convex on $[0,\infty)$, then the ratio of the costs incurred by regret-minimizing players to the cost of the global optimum flow is asymptotically at most $\alpha(\mathcal{L})$ (which is the Price of Anarchy bound given by Roughgarden and Tardos [109]).*

Proof. Let f^* be an optimal action distribution and $f_1, \ldots, f_T$ be a sequence of action distributions obtained by regret-minimizing players. We can lower bound the optimum social cost using a linear approximation of the function $\ell_e(\xi)\xi$ at the point $\lambda_e^t f_e^t$, where $\lambda_e^t \in [0,1]$ solves $\ell_e^*(\lambda_e^t f_e^t) = \ell_e(f_e^t)$:

$$\begin{aligned} \ell_e(f_e^*)f_e^* &= \ell_e(\lambda_e^t f_e^t)\lambda_e^t f_e^t + \int_{\lambda_e^t f_e^t}^{f_e^*} \ell_e^*(f)\,dx \\ &\geq \ell_e(\lambda_e^t f_e^t)\lambda_e^t f_e^t + (f_e^* - \lambda_e^t f_e^t)\ell_e^*(\lambda_e^t f_e^t) \\ &= \ell_e(\lambda_e^t f_e^t)\lambda_e^t f_e^t + (f_e^* - \lambda_e^t f_e^t)\ell_e(f_e^t) \end{aligned}$$

for all edges and time steps, and thus

$$C(f^*) \geq \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E}[\ell_e(\lambda_e^t f_e^t)\lambda_e^t f_e^t + (f_e^* - \lambda_e^t f_e^t)\ell_e(f_e^t)].$$

We can rewrite this as

$$C(f^*) \geq \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E}[\mu_e^t\lambda_e^t f_e^t + (1-\lambda_e^t)f_e^t]\ell_e(f_e^t) + \sum_{e\in E}[f_e^* - f_e^t]\ell_e(f_e^t),$$

where $\mu_e^t = \ell_e(\lambda_e^t f_e^t)/\ell_e(f_e^t)$. By the regret minimizing property,

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} f_e^t\ell_e(f_e^t) \leq \epsilon + \sum_i a_i \min_{P\in\mathcal{P}_i}\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t)$$

and thus

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} f_e^t\ell_e(f_e^t) \leq \epsilon + \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} f_e^*\ell_e(f_e^t),$$

which gives us

$$C(f^*) + \epsilon \geq \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E}[\mu_e^t\lambda_e^t f_e^t + (1-\lambda_e^t)f_e^t]\ell_e(f_e^t).$$

By definition, $\mu_e^t\lambda_e^t + (1-\lambda_e^t) \geq 1/\alpha(\mathcal{L})$ for each e and t, so $\mu_e^t\lambda_e^t f_e^t + (1-\lambda_e^t)f_e^t]\ell_e(f_e^t)$ and $\ell_e(f_e^t)f_e^t$ differ by at most a multiplicative $\alpha(\mathcal{L})$ factor for every e and t. This gives us

$$C(x^*) + \epsilon \geq \frac{1}{\alpha(\mathcal{L})}\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t)f_e^t = \frac{C(x)}{\alpha(\mathcal{L})},$$

as desired. □

4.7 Discrete Users: Parallel Paths

In contrast with the previous sections, we now consider discrete users, where we denote the ith user weight as w_i. Without loss of generality, we assume that the weights are normalized such that $\sum_{i=1}^n w_i = 1$. We limit ourselves in this section to the single-commodity version of the parallel paths routing game model and to functions with latency equal to the load, that is, for a path e we have $\ell_e = f_e$. For each user i, we let the latency excluding her own path e at time t be $\ell_e(f_e^t \setminus i)$ and her average latency on path e be $\ell_e(\hat{f}_e \setminus i) = \frac{1}{T}\sum_{t=1}^{T}\ell_e(f_e^t \setminus i)$, where $f_e^t \setminus i = f_e^t$ if user i is not routing on path e and $f_e^t \setminus i = f_e^t - w_i$ otherwise. We always exclude the ith player from the latency function, since the ith player always pays for its weight.

Next we observe that at time t, there always exists a path with load at most the average load.

Observation 4.7.1. *At any time step t, for every user i, there exists a path e such that $\ell_e(\hat{f}_e \setminus i) \leq \frac{1-w_i}{m}$.*

The following theorem differs from other theorems in this chapter in the sense that it is an expectation result and holds for every user.

Theorem 4.7.2. *Consider the parallel paths model, with latency functions such that the latency equals the load. Assume that each discrete user i uses an optimized best expert algorithm. Then for all users, for all $T \geq O(\frac{\log m}{\epsilon^2})$,*

$$\frac{1}{T}\sum_{t=1}^{T} E_{e\sim q_t}[\ell_e(f_e^t \setminus i)] \leq \frac{1-w_i}{m} + \epsilon,$$

where q_t is the distribution over the m paths output by the best expert algorithm at time t.

Proof. By Observation 4.7.1 we have that there exists a path with average cost at most $\frac{1-w_i}{m}$. Since user i is using an optimized best expert algorithm and the maximal latency is 1, we have that

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \mathrm{E}_{e\sim q_t}[\ell_e(f_e^t \setminus i)] &\leq \min_{e\in E} \ell_e(\hat{f}_e \setminus i) + \sqrt{\frac{\log m}{T}} \\ &\leq \frac{1-w_i}{m} + \sqrt{\frac{\log m}{T}} \\ &\leq \frac{1-w_i}{m} + \epsilon\end{aligned}$$

where the last inequality holds for $T \geq O(\frac{\log m}{\epsilon^2})$. □

Consider an instance of this model where every user plays uniformly at random. The resulting flow is clearly a Nash equilibrium, and the expected latency for the ith player is $\frac{1-w_i}{m}$ excluding its own weight. We thus have shown that the expected latency experienced by each user i is at most ϵ worse than this Nash latency.

4.8 Conclusions

In this chapter, we consider the question: if each player in a routing game (or more general congestion game) uses a no-regret strategy, will behavior converge to a Nash equilibrium, and under what conditions and in what sense? Our main result is that in the setting of multicommodity flow and infinitesimal agents, a $1-\epsilon$ fraction of the daily flows are at ϵ-Nash equilibrium for ϵ approaching 0 at a rate that depends polynomially on the players' regret bounds and the maximum slope of any latency function. Moreover, we show the dependence on slope is necessary.

Even for the case of reasonable (bounded) slopes, however, our bounds for general nonlinear latencies are substantially worse than our bounds for the linear case. For instance if agents are running the Kalai-Vempala algorithm [81], we get a bound of $O(\frac{mn\log n}{\epsilon^2})$ on the number of time steps needed for the time-average flow to reach an ϵ-Nash equilibrium in the linear case, but $O(\frac{mn^3\log n}{\epsilon^4})$ for general latencies. We do not know if these bounds in the general case can be improved. In addition, our bounds on the daily flows lose additional polynomial factors which we suspect are not tight.

We also show that Price of Anarchy results can be applied to regret-minimizing players in routing games, that is, that existing results analyzing the quality of Nash equilibria can also be applied to the results of regret-minimizing behavior. Recent work [16] shows that in fact Price of Anarchy results can be extended to cover regret-minimizing behavior in a wide variety of games, including many for which this behavior may not approach equilibria and where Nash equilibria may be hard to find.

Chapter 5

The Price of Total Anarchy

5.1 Introduction

As mentioned in the introduction, one of the main thrusts of research in algorithmic game theory has been the study of the ratio between the cost of the worst Nash equilibrium and that of the social optimum (the “Price of Anarchy” [87]), as a tool for understanding the outcomes of selfish behavior. In this chapter, we study the value obtained in games with selfish agents when we make a much weaker and more realistic assumption about their behavior. We consider repeated play of the game and allow agents to play any sequence of actions with only the assumption that this action sequence has low regret with respect to the best fixed action in hindsight. This “price of total anarchy” is strictly a generalization of price of anarchy, since in a Nash equilibrium, all players have zero regret. Regret minimization is a realistic assumption because there exist a number of efficient algorithms for playing games that guarantee regret that tends to zero, because it requires only localized information, and because in a game with many players in which the actions of any single player do not greatly affect the decisions of other players (as is often studied in the network setting), players can only improve their situation by switching from a strategy with high regret to a strategy with low regret.

We consider four classes of games: Hotelling games, in which players compete with each other for market share, valid games [118] (a broad class of games that includes among others facility location, market sharing [65], traffic routing, and multiple-item auctions), linear congestion games with atomic players and unsplittable flow [6] [25], and parallel link congestion games [87]. We prove that in the first three cases, the price of total anarchy matches the price of anarchy exactly even if the play itself is not approaching equilibrium; for parallel link congestion we get an exact match for $n = 2$ links but an exponentially greater price for general n when the social cost function is the makespan. When we consider average load instead, we prove that if the machine speeds are relatively bounded, that the price of total anarchy is $1 + o(1)$, matching the price of anarchy. For linear congestion games and average cost load balancing, the price of anarchy bounds were previously only known for pure strategy Nash equilibria, and as a corollary of our price of total anarchy bounds, we prove the corresponding price of anarchy bound for mixed Nash equilibria as well.

Most of our results further extend to the case in which only some of the agents are acting to minimize regret and others are acting in an arbitrary (possibly adversarial) manner. When studying *anarchy*, it is vital to consider players who behave unpredictably, and yet this has been largely ignored up until now. Since Nash equilibria are stable only if all players are participating, and sink equilibria [63] are defined over state graphs that assume that all players play rationally, such guarantees are not possible under the standard price of anarchy model or the price of sinking model [63].[1]

5.1.1 Our results

In this chapter, we study the price of total anarchy in four classes of games. We emphasize that our analysis does not presume that players play according to any particular class of algorithms; our results hold whenever players happen to experience low regret, which is a strictly weaker assumption than that players play according to a Nash equilibrium. In Section 5.3 we examine a class of generalized Hotelling games, where sellers select locations on a graph and achieve revenues that depend on their own locations as well as the locations chosen by the other sellers. We prove that for such games (and an even broader class, see Section 5.3.3), any regret minimizing player gets at least half of her fair share of the sales, regardless of how the other (Byzantine) players behave.[2] This result exactly matches the price of anarchy in these games. Hotelling games and their generalizations model not only situations involving staking out market share in physical space, but also to the game politicians play in choosing how to position themselves on the political landscape.

Valid games, introduced by Vetta [118], model games where the social utility is submodular, the private utility of each player is at least her Vickrey utility (the amount her presence contributes to the overall welfare), and where the sum of the players' private utilities is at most the total social utility. In Section 5.4 we prove that the price of total anarchy in valid games with nondecreasing social utility functions exactly matches the (Nash) price of anarchy, even if Byzantine players are added to the system.

Finally, in Section 5.5, we analyze atomic congestion games with two types of social welfare functions. First, we consider unweighted atomic congestion games with player-summed social welfare functions, and in both the linear cost and the polynomial cost case, we show price of total anarchy results that match the price of anarchy [6, 25]. Next, we consider a parallel link congestion game with social welfare equal to makespan, the game that initiated the study of the price of anarchy [87], and show that the price of total anarchy of the parallel link congestion game with two links is $3/2$, exactly matching the price of anarchy. We also show that the price of total anarchy in the parallel link game with n links is $\Omega(\sqrt{n})$, which is strictly worse than the price of anarchy. Finally, we show a price of total anarchy matching the known price of anarchy in the load balancing game with the sum social utility function. In the case of load balancing with sum social utility, our price of total anarchy results also yield previously unknown price of anarchy results for mixed strategies.

[1] Babaioff et al. [9] propose a model of network congestion with "malicious" players. Their model defines malicious behavior as optimizing a specific function, however, and is not equivalent to arbitrary play.

[2] We note that robustness to Byzantine players is not inherent in our model. Indeed, there exist games for which the addition of Byzantine players can make the social welfare, as well as the utilities of individual regret-minimizing players, arbitrarily bad.

In Section 5.6, we discuss techniques for minimizing regret in each of these settings.

5.2 Preliminaries

In this chapter, we consider k-player games. For each player i, we denote by $\mathcal{A}_i$ the set of pure strategies available to that player. A mixed strategy is a probability distribution over actions in $\mathcal{A}_i$; we denote by $\mathcal{S}_i$ the set of mixed strategies available to player i. Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_k$ and $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_k$. Every game has an associated social utility function $\gamma : \mathcal{A} \to \mathbb{R}$ that takes a set containing an action for each player to some real value. Each player i has an individual utility function $\alpha_i : \mathcal{A} \to \mathbb{R}$.

We often want to talk about the social or individual utility of a strategy profile $S = \{s_1, \ldots, s_k\} \in \mathcal{S}$. To this end, we denote by $\bar{\gamma} : \mathcal{S} \to \mathbb{R}$ the expected social utility over randomness of the players and by $\bar{\alpha}_i : \mathcal{S} \to \mathbb{R}$ the expected value of the utility of a strategy profile to player i. We denote the social value of the socially optimum strategy profile by $\mathbf{OPT} = \max_{S \in \mathcal{S}} \bar{\gamma}(S)$ in maximization problems. Correspondingly, $\mathbf{OPT} = \min_{S \in \mathcal{S}} \bar{\gamma}(S)$ in minimization problems.

We also sometimes wish to talk about a modification of a particular strategy profile; let $S \oplus s'_i$ be the strategy set obtained if player i changes her strategy from s_i to s'_i. Let $\emptyset_i$ be the null strategy for player i (player i takes no action). We use superscripts to denote time, so S^t is the strategy profile at time t; s_i^t is player i's strategy at time t.

Definition 5.2.1. *The price of total anarchy for an instance of a maximization game is defined to be* $\max \frac{\mathbf{OPT}}{\frac{1}{T}\sum_{t=1}^{T} \bar{\gamma}(S^t)}$*, where the max is taken over all* T *and* $S^1, S^2, \ldots, S^T$*, where* $S_1, \ldots, S_T$ *are play profiles of players with the regret-minimizing property. The price of total anarchy for an instance of a minimization game is defined to be* $\max \frac{\frac{1}{T}\sum_{t=1}^{T} \bar{\gamma}(S^t)}{\mathbf{OPT}}$*, where the max is taken over all* T *and* $S^1, S^2, \ldots, S^T$*, where* $S_1, \ldots, S_T$ *are play profiles of players with the regret-minimizing property.*

Because all players have zero regret when playing a Nash equilibrium, the price of total anarchy of a game is never less than its price of anarchy. In this chapter we study the price of anarchy and the price of total anarchy for general classes of games. The price of (total) anarchy for a class of games is defined to be the maximum price of (total) anarchy over any instance in that class. Bounds on the price of (total) anarchy for a class of games may not be tight for particular instances in that class.

5.3 Hotelling games

Hotelling games [75] are well studied in the economics literature; see, for example, [61] and [84] for surveys. Hotelling games are traditionally location games played on a line, but we generalize them to an arbitrary graph and a broad class of behaviors on the part of the customers. We prove our result first for a specific Hotelling game, and then observe that our proof still holds in a much more general setting.

5.3.1 Definition and price of anarchy

Imagine a set of souvenir stand owners in Paris who must decide where to set up their souvenir stands each day. Every day, n tourists buy a souvenir from whichever stand they find first. Each stand operator wishes to maximize her own sales. Every day there are n sales, and we wish to maximize fairness: The social welfare function is the minimum sales of any souvenir stand. Formally, this maximization game is defined by an n vertex graph $G = (V, E)$ and a number k of players, called sellers. Every seller i among the k sellers has strategy set $\mathcal{A}_i = V$, that is, every day she sets up her stand on some vertex of the graph. Each day, every tourist chooses a path from some private distribution over orderings of the nodes of the graph, and buys from the seller he encounters first (for instance, as a special case, we could have one tourist at each vertex of the graph who purchases from the nearest souvenir stand). If two sellers are reached at the same time, we assume the tourist splits his contribution among them equally. At any time t the social welfare is $\bar{\gamma}(S^t) = \min_i \bar{\alpha}_i(S^t)$. The social optimum is obtained by splitting all vertices equally among all k players (this can be achieved if all players play on the same vertex). Therefore $\mathbf{OPT} = n/k$. This tension between the objective of a franchiser to locate outlets in such a way that each individual franchisee has sufficient demand and the desires of the individual franchisees to maximize profits, has been studied in the business and operations research literature [28].

In general, Hotelling-style games need not have pure equilibria: consider a continuous version of the game, where sellers can select any location on $[0, 1]$ and receive revenue equal to the total region to which they are closest. Again, if multiple sellers choose the same location, they evenly split the corresponding revenue. Now, no matter how we fix the locations of three players, at least one of them will wish to move, to undercut the others. Despite this, we can study the quality of the mixed Nash equilibria of the game.

Theorem 5.3.1. *The price of anarchy of the Hotelling game is* $(2k - 2)/k$.

Proof. Given a strategy set S, consider the alternate set $(S \oplus \emptyset_i)$. There are $k - 1$ active players in this alternate set and the total payoff is still n, so there must be some player h who achieves expected payoff $\bar{\alpha}_h(S \oplus \emptyset_i) \geq n/(k - 1)$. If player i played the same strategy as player h, she would achieve expected payoff $\bar{\alpha}_i(S \oplus s_h) \geq \frac{n}{(2k-2)}$. Thus, any strategy achieving expected payoff less than $\frac{n}{(2k-2)}$ is not an equilibrium strategy, since in a Nash equilibrium, no player wishes to change her strategy.

This bound is tight: Consider a game on a graph with $k-1$ identical stars, where we identify tourists with vertices of the graph and each patronizes the nearest souvenir stand. In this example, $k - 1$ of the players play deterministically at the center of their own star; player k plays uniformly at random over all $k - 1$ star centers. This strategy set S is a Nash equilibrium, and the randomizing player earns $\bar{\alpha}_k(S) = n/(2k - 2)$ (the other players do better), so the social welfare $\gamma(S) = n/(2k - 2)$. Since $\mathbf{OPT} = n/k$, this demonstrates that the price of anarchy is $\frac{\mathbf{OPT}}{\bar{\gamma}(S)} = \frac{(2k-2)}{k}$. □

5.3.2 Price of total anarchy

Since at a Nash equilibrium, no player has regret, the price of total anarchy for the Hotelling game is at least $(2k - 2)/k$. In this section, we show that this value is tight; that is:

Theorem 5.3.2. *The price of total anarchy in the Hotelling game is $(2k-2)/k$, matching the price of anarchy.*

The proof of this theorem relies on the symmetry of the game; this property was similarly useful to Chien and Sinclair [24] in the context of studying convergence to Nash equilibria in symmetric congestion games.

Let O_i^t be the set of plays at time t by all players *other* than player i. Let $O_i = \sum_{t=1}^{T} O_i^t$, the union with multiplicity of all plays of players other than i over all time periods.

Definition 5.3.3. *Let $\Delta_i^{t\to u}$ be the quantity such that if player i plays an action uniformly at random from O_i^t at time step u, she achieves expected payoff $n/(2k-2)+\Delta_i^{t\to u}$. Note that $\Delta_i^{t\to t}$ is always non-negative because the $k-1$ other players have average payoff exactly $n/(k-1)$ when player i is removed.*

Lemma 5.3.4. *For all i, for all $1 \leq t, u \leq T$: $\Delta_i^{u\to t}+\Delta_i^{t\to u} \geq 0$.*

Proof. If $t=u$, the claim follows easily, as noted in the definition. Otherwise, imagine a $(2k-2)$-player game in which there is a time-t player and a time-u player for each original player other than i. The time-t version of a player j plays strategy s_j^t; the time u version plays s_j^u. Since the sum of all players' payoffs is n, if player i picks a random strategy from among those already being played and plays it in this imaginary game *replacing* the player she copies, i expects to have payoff $n/(2k-2)$. Half of the time, player i will select a time-t strategy and replace that time-t player. It can only improve i's payoff in this case to remove all of the other time-t players and only play against time-u players. This leaves i playing a strategy uniformly selected from O_i^t at time u. A parallel argument holds the other half of the time, when player i selects a time-u strategy, and thus

$$\begin{aligned}\frac{n}{(2k-2)} &\leq \frac{1}{2}\left(\frac{n}{(2k-2)}+\Delta_i^{t\to u}\right)+\frac{1}{2}\left(\frac{n}{(2k-2)}+\Delta_i^{u\to t}\right)\\ &= \frac{n}{(2k-2)}+\frac{1}{2}(\Delta_i^{t\to u}+\Delta_i^{u\to t})\end{aligned}$$

as desired. □

Proof of Theorem 5.3.2. Fix a sequence of plays $S^1,\ldots,S^T$. Recall that $O_i = O_i^1+\ldots+O_i^T$. Define o_i^t to be the uniform distribution over O_i^t. Picking an action a uniformly at random from O_i is equivalent to picking a random time step u and then picking a strategy $a \in O_i^u$ uniformly at random. Player i's expected payoff had she randomly selected o_i^u and played it over all T rounds is

$$\begin{aligned}\frac{1}{T}\sum_{u=1}^{T}\sum_{t=1}^{T}\bar{\alpha}_i(S^t\oplus o_i^u) &= \frac{1}{T}\sum_{u=1}^{T}\sum_{t=1}^{T}\left(\frac{n}{(2k-2)}+\Delta_i^{u\to t}\right)\\ &= \frac{Tn}{(2k-2)}+\frac{1}{T}\sum_{u=1}^{T}\sum_{t=1}^{T}\Delta_i^{u\to t}\\ &\geq \frac{Tn}{(2k-2)},\end{aligned}$$

where the last inequality holds because of Lemma 5.3.4. Therefore, there must be some single fixed action $a^* \in S$ that achieves at least $\frac{Tn}{(2k-2)}$ when played over T rounds of the above game. Any regret minimizing player achieves expected total payoff at least this much (minus ϵ), and so has expected payoff at least $n/((2k-2)) - \epsilon$, proving the theorem. □

5.3.3 The price of total anarchy in generalized Hotelling games

We note that the proof of Theorem 5.3.2 made no use of the specifics of the Hotelling game described above. In particular, the same proof shows that any regret minimizing player achieves expected payoff approaching $n/(2k-2)$ *regardless* of how other players behave, and so we are able to guarantee good payoff among regret-minimizing players players even in the presence of Byzantine players making arbitrary (or adversarial) decisions.

Theorem 5.3.5. *Any player who minimizes regret in the Hotelling game achieves payoff approaching $n/(2k-2)$, regardless of how the other players play.*

The same proof also holds when the buyers use much more general rules for choosing which stand to patronize.[3] Neither do we use the fact that players' utilities are linear. In fact, our proof only makes use of three properties of the Hotelling game:

1. **Constant Sum**: The individual utilities of the players in the game always sum to the same value, regardless of play.
2. **Symmetric**: All players have the same action set, and the payoff vector is a function of the action vector that is invariant to a permutation of the names of the players.
3. **Monotone**: The game is defined for any number of players, and removing players from the game (while keeping the strategies of the remaining players fixed) does not decrease the payoff for any remaining player. If multiple players employ the same pure strategy, their total utility is at least the utility that would be achieved by a single player among them employing that strategy while the others among them play the empty strategy.

We call such games with the "fairness" social utility function $\bar{\gamma}(S) = \min_i \alpha_i(S)$ *generalized Hotelling games* and get the following theorem:

Theorem 5.3.6. *In any k-player, generalized Hotelling game, the price of total anarchy among regret minimizing players is $(2k-2)/k$ even in the presence of arbitrarily many Byzantine players.*

One slight generalization of the Hotelling game that fits this model is as follows: buyers each have different distributions over permutations on the nodes in the graph; every day they sample from that distribution and visit the nodes in the given order, buying from the first seller they encounter. Note that in this setting, if the number of buyers is super-constant, and we only have oracle access to the buyers, it is not clear how to solve for a Nash equilibrium in polynomial time, but sellers may efficiently run regret minimizing algorithms.

Models for understanding how customers select among sellers are are understandably a hot topic in operations research. One of the primary approaches was introduced by Huff [76, 77],

[3] One caveat is that customers may not in general base their selection rules on the actions of the players—for instance by patronizing the *second* closest souvenir stand. If we were to allow rules such as this, removing players from the game could decrease the payoff of some of the remaining players, and we rely on this not being the case.

and proposes that customers will buy from a particular seller with a probability that depends inversely on the distance to it, and also on some measure of the seller "attractiveness." Such a *gravity based* model of buyer choices also fits into the generalized Hotelling framework. While one line of the subsequent work in this area has been on techniques for evaluating attractiveness, a second line of work focuses on mathematical programming techniques for selecting facility locations to maximize market share in this model (see, for example, [39, 41]).

Another generalization of the Hotelling game for which these theorems hold is the c-*franchise game*, wherein each seller must choose locations on the graph for c businesses. Megiddo et al. [93] give an $O(cn^2)$ algorithm for the offline version of this problem, the *m*aximum coverage location problem (when customers simply patronize the nearest store), when the game graphs are restricted to trees; they also observe that the general problem is closely related to the NP-hard problem of minimum dominating set. The c-franchise model, in combination with the gravity model, has also received attention in the operations research literature, with Drezner et al. [40] proposing a complex multi-step heuristic procedure for solving even the offline problem, when the locations of the other player's franchises are known ahead of time.

Building on the idea of the gravity-based model, we can further generalize the class of Hotelling games to remove the constant sum assumption; the resulting class of *generalized location games* encompasses Hotelling games, but also similar games where the buyers have a maximum distance they are willing to travel, other models of buyer behavior, and Hotelling games on disconnected graphs. We can build on our results for generalized Hotelling games to bound the price of anarchy and price of total anarchy for these games as well:

Theorem 5.3.7. *The price of anarchy for generalized location games is at most 2.*

Proof. Let v be the social welfare of an optimal solution (that is, the number of customers served by the worst seller). Consider a Nash equilibrium strategy set S. If there exists a player j with payoff $\bar{\alpha}_j(S) \geq v$, any other player in S would prefer to defect to action S_j and get payoff at least $v/2$, were she not already achieving at least this utility.

Otherwise, no such player j exists. In this case, a player i considering defecting from S could consider each of the $k-1$ strategies taken by other players in S, plus the k actions taken by players in $\mathbf{OPT}$. The union of these $2k-1$ strategies (note that there may be duplicates) covers at least n' customers in expectation, and so the expected value achieved by the best strategy among them (were all $2k-1$ strategies played simultaneously) is at least $\frac{n'}{2k-1} \geq \frac{vk}{2k-1} \geq v/2$. Among these $2k-1$ actions the one that achieves the best performance when played against the $k-1$ actions in $(S \oplus \emptyset_i)$ then achieves expected value more than $v/2$, and so this best strategy is one of the actions in $\mathbf{OPT}$. If there exists a player i in S with payoff $\bar{\alpha}_i(S) \leq v/2$, she would improve her expected payoff by defecting to this action in $\mathbf{OPT}$. □

The proof of the price of total anarchy for generalized Hotelling games is based on the idea of copying an action of a random opponent at a random timestep in history, and showing that the expected regret of this fixed action is low, *e*ven when not making any assumptions about the guarantees achieved by your opponents. Unfortunately, this is not true for generalized location games, since arbitrary opponents could choose actions so that the total number of customers they serve is much less than the optimal solution serves. Instead, we can make an argument similar to that above for the price of anarchy for generalized location games, that *either* copying a random action in history *or* playing an action from an optimal strategy profile will have low regret.

Theorem 5.3.8. *The price of total anarchy in generalized location games is at most* 3*.*

Proof. Again, let v be the social welfare of an optimal solution. Our analysis considers two cases with respect to a regret-minimizing player i.

In the first case, suppose that the history $S^1, S^2, \ldots, S^T$ is such that the average number of customers serviced on each timestep by $S^1 \oplus \emptyset_i, \ldots, S^T \oplus \emptyset_i$ is at least $2kv/3$. In this case, we can use an analysis similar to that in the proof of Theorem 5.3.2 to show that a randomly-selected opponent action from a random time step has expected average payoff at least $v/3$ in hindsight.

Otherwise, consider the expected payoff in hindsight of the group of actions that make up an optimal solution; since these actions by themselves cover at least kv customers, even in competition with the action history, they cover at least $kv/3$ on average. Then at least one of the k actions in $\mathbf{OPT}$ achieves average payoff at least $v/3$ when played against $S^1 \oplus \emptyset_i, \ldots, S^T \oplus \emptyset_i$.

Since there always exists a fixed action with average payoff at least $v/3$, regret-minimizing algorithms converge to achieve at least this payoff, as well. □

Note that the above proof did not use the assumption that the opponents are regret-minimizing (or any other assumption about their actions), and so this result holds even against Byzantine opponents.

5.3.4 Regret minimization need not converge

Since players may efficiently minimize regret in Hotelling games, but may not necessarily be able to compute Nash equilibria, it is notable that we are able to match standard price-of-anarchy guarantees. In fact, it is possible that regret-minimizing players in Hotelling games never converge to a Nash equilibrium:

Theorem 5.3.9. *Even if all players in the Hotelling game are regret minimizing, stage game play need not converge to Nash equilibrium.*

Proof. Consider k players $\{0, \ldots, k-1\}$ on a graph with $k-1$ identical $(n-1)/(k-1)$-vertex stars with centers $v_0, \ldots, v_{k-2}$ and an isolated vertex v_{k-1}. At time period t, player i plays on vertex $v_{t+i \mod k}$. Each player has expected payoff $\left(\frac{k-1}{k}\right)\left(\frac{n-1}{k-1}\right)+\left(\frac{1}{k}\right) = n/k$, but no fixed vertex has expected payoff more than $\left(\frac{k-2}{k}\right)\left(\frac{n-1}{2(k-1)}\right)+\left(\frac{1}{k}\right)\left(\frac{n-1}{k-1}\right)+\frac{1}{k}$, so no player has positive regret. However, at each time period, the player at the isolated vertex v_{k-1} has incentive to deviate, so this is not a Nash equilibrium. □

In addition, we observe that the uncoupled empirical distribution of play does not constitute a mixed Nash equilibrium, nor does the *joint* empirical distribution of play constitute a mixed Nash. This highlights the fact that a sequence of play can be low regret in hindsight, but still place nonzero probability on a strictly dominated action.

A similar example shows that even if all players minimize internal regret (so that play is guaranteed to converge to the set of correlated equilibria), play can cycle forever and so need not converge to Nash equilibrium.[4]

[4] k players play on a set of $k/2+1$ vertices. Players are divided into two equal sized groups, L and R. Every

5.4 Valid games

5.4.1 Definitions and price of anarchy

Valid games, introduced by Vetta [118], are a broad class of games that includes the market sharing game studied by Goemans et al. [65], the facility location problem, a version of the traffic routing problem of Roughgarden and Tardos [108], and multiple-item auctions [118]. When describing valid games, we slightly adapt the notation of [118]. Consider a k-player maximization game, where each player i has a groundset of actions $\mathcal{V}_i$ from which she can play some subset. Not every subset of actions is necessarily allowed. Let $\mathcal{V} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_k$, and let $\mathcal{A}_i = \{a_i \subseteq \mathcal{V}_i : \ a_i \text{ is a feasible action}\}$. Let the game have some social utility function $\gamma : 2^{\mathcal{V}} \to \mathbb{R}$, and let each player have a private utility function $\alpha_i : 2^{\mathcal{V}} \to \mathbb{R}$. The discrete derivative of f at $X \subseteq V$ in the direction $D \subseteq V - X$ is $f'_D(X) = f(X \cup D) - f(X)$.

Definition 5.4.1. *A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular if for $A \subseteq B$, $f'_i(A) \geq f'_i(B)$ $\forall i \in \mathcal{V} - B$.*

Note that submodular utility functions represent the economic concept of decreasing marginal utility, reflecting economies of scale.

Definition 5.4.2. *A game with private utility functions $\alpha_i : 2^{\mathcal{V}} \to \mathbb{R}$ and social utility function $\gamma : 2^{\mathcal{V}} \to \mathbb{R}$ is valid if γ is submodular and*

$$\bar{\alpha}_i(S) \quad \geq \quad \bar{\gamma}'_{s_i}(S \oplus \emptyset_i) \tag{5.1}$$

$$\sum_{i=1}^{k} \bar{\alpha}_i(S) \quad \leq \quad \bar{\gamma}(S) \tag{5.2}$$

Condition 5.1 states that each agent's payoff is at least her *Vickrey utility*—the change in social utility that would occur if agent i did not participate in the game. Condition 5.2 states that the social utility of the game is at least the sum of the agents' private utilities.

For example, consider the market sharing game studied by Goemans et al. [65]. The game is played on a bipartite graph $G = ((V, U), E)$. Each vertex in V is a player, and each vertex in U is a market. Each market has a value and a cost to service it, and each player has a budget. A player may enter a set of markets to which she has edges, if the sum of their costs is at most her budget. For each market that a player enters, she receives payoff equal to the value of that market divided by the number of players that chose to enter it. The social utility function is the sum of the individual player utilities, or equivalently, the sum of the values of the markets that have been entered by any player. This valid game models a situation in which cable internet providers enter different cities with values proportional to their populations and share the market equally with other local providers; the social utility is the number of people with access to high speed internet.

Vetta [118] analyzes the price of anarchy of valid games and shows that if S is a Nash equilibrium strategy and $\Omega = \{\sigma_1, \ldots, \sigma_k\}$ is a strategy profile optimizing the social utility function

turn, there is exactly one player on $k/2$ vertices, and $k/2$ players on the remaining vertex. Players in L and R get their own vertices on alternate turns, and the crowded vertex rotates, so that each player is equally often on every vertex, and on any particular vertex she is equally often alone and crowded. Therefore no player has any incentive to swap any vertex with any other.

so that $\bar{\gamma}(\Omega) = \mathbf{OPT}$, then

$$\begin{aligned}\mathbf{OPT} \leq 2\bar{\gamma}(S) \quad &- \sum_{i:\sigma_i = s_i} \bar{\gamma}'_{s_i}(S \oplus \emptyset_i) \\ &- \sum_{i:\sigma_i \neq s_i} \bar{\gamma}'_{s_i}(\Omega \cup (S \oplus \emptyset_i \oplus \ldots \oplus \emptyset_k)).\end{aligned}$$

Thus, if γ is nondecreasing, then for any Nash equilibrium strategy S, $\gamma(S) \geq \mathbf{OPT}/2$, giving a price of anarchy of 2. In contrast, Goemans et al. [63] show that the price of sinking for valid games is larger than n.

5.4.2 Price of total anarchy

In this section, we show that the price of total anarchy for valid games matches the price of anarchy exactly:

Theorem 5.4.3. *If all players play regret-minimizing strategies for T rounds, with strategy profile S^i at time i, then*

$$\mathbf{OPT} \leq \frac{1}{T}\sum_{i=1}^{T}\Bigg(2\bar{\gamma}(S^t) - \sum_{i:\sigma_i = s_i^t} \bar{\gamma}'_{s_i^t}(S^t \oplus \emptyset_i) \\ - \sum_{i:\sigma_i \neq s_i^t} \bar{\gamma}'_{s_i^t}(\Omega \cup (S^t \oplus \emptyset_i \oplus \ldots \oplus \emptyset_k))\Bigg) + \epsilon k.$$

Proof. Suppose all players use low regret strategies, so that for any player i,

$$T\epsilon + \sum_{t=1}^{T} \bar{\alpha}_i(S^t) \geq \sum_{t=1}^{T} \bar{\alpha}_i(S^t \oplus \sigma_i).$$

Expanding terms, we can rewrite this as

$$\begin{aligned}T\epsilon + \sum_{t:s_i^t = \sigma_i} \bar{\alpha}_i(S^t) + \sum_{t:s_i^t \neq \sigma_i} \bar{\alpha}_i(S^t) \\ \geq \sum_{t:s_i^t = \sigma_i} \bar{\alpha}_i(S^t \oplus \sigma_i) + \sum_{t:s_i^t \neq \sigma_i} \bar{\alpha}_i(S^t \oplus \sigma_i).\end{aligned}$$

We note that when $s_i^t = \sigma_i$, $\bar{\alpha}_i(S^t) = \bar{\alpha}_i(S^t \oplus \sigma_i)$, so this yields

$$\epsilon T + \sum_{t:s_i^t \neq \sigma_i} \bar{\alpha}_i(S^t) \geq \sum_{t:s_i^t \neq \sigma_i} \bar{\alpha}_i(S^t \oplus \sigma_i).$$

Summing over all players, we get

$$\begin{aligned}k\epsilon T + \sum_{i=1}^{k} \sum_{t:s_i^t \neq \sigma_i} \bar{\alpha}_i(S^t) \quad &\geq \quad \sum_{i=1}^{k} \sum_{t:s_i^t \neq \sigma_i} \bar{\alpha}_i(S^t \oplus \sigma_i) \\ &\geq \quad \sum_{i=1}^{k} \sum_{t:s_i^t \neq \sigma_i} \bar{\gamma}'_{\sigma_i}(S^t \oplus \emptyset_i),\end{aligned}$$

where the second equation holds by assumption 5.1. Now note that

$$\begin{aligned}
\sum_{t=1}^{T} \bar{\gamma}(S^t) &\geq \sum_{t=1}^{T}\sum_{i=1}^{k} \bar{\alpha}_i(S^t) \\
&= \sum_{t=1}^{T} \sum_{i:\sigma_i = s_i^t} \bar{\alpha}_i(S^t) + \sum_{t=1}^{T} \sum_{i:\sigma_i \neq s_i^t} \bar{\alpha}_i(S^t) \\
&\geq \sum_{t=1}^{T} \sum_{i:\sigma_i \neq s_i^t} \bar{\alpha}_i(S^t) + \sum_{t=1}^{T} \sum_{i:\sigma_i = s_i^t} \bar{\gamma}'_{s_i^t}(S^t \oplus \emptyset_i) \\
&= \sum_{i=1}^{k} \sum_{t:\sigma_i \neq s_i^t} \bar{\alpha}_i(S^t) + \sum_{t=1}^{T} \sum_{i:\sigma_i = s_i^t} \bar{\gamma}'_{s_i^t}(S^t \oplus \emptyset_i),
\end{aligned}$$

where the third line holds by assumption 5.2 and the fourth line is a reordering of the summations. This gives us

$$\begin{aligned}
\sum_{i=1}^{k} \sum_{t:\sigma_i \neq s_i^t} \bar{\gamma}'_{\sigma_i^t}(S^t \oplus \emptyset_i) \leq T\epsilon k + \sum_{i=1}^{k} \sum_{t:\sigma_i \neq s_i^t} \bar{\alpha}_i(S^t) \\
\leq T\epsilon k + \sum_{t=1}^{T} \bar{\gamma}(S^t) - \sum_{t=1}^{T} \sum_{i:\sigma_i = s_i^t} \bar{\gamma}'_{s_i^t}(S^t \oplus \emptyset_i).
\end{aligned}$$

We use the following lemma proved by Vetta [118]:

Lemma 5.4.4. *If $\Omega = \{\sigma_1, \ldots, \sigma_k\}$ is a strategy profile optimizing the social utility function γ, then for any strategy profile S*

$$\bar{\gamma}(\Omega) \leq \bar{\gamma}(S) + \sum_{i:\sigma_i \neq s_i} \bar{\gamma}'_{\sigma_i}(S \oplus \emptyset_i) - \sum_{i:\sigma_i \neq s_i} \bar{\gamma}'_{s_i}(\Omega \cup (S \oplus \emptyset_i \oplus \ldots \oplus \emptyset_k)).$$

From Lemma 5.4.4, for any sequence of plays $S_1, \ldots, S^t$,

$$\begin{aligned}
T\bar{\gamma}(\Omega) \leq \sum_{t=1}^{T} \Bigg(\bar{\gamma}(S^t) + \sum_{i:\sigma_i \neq s_i^t} \bar{\gamma}'_{\sigma_i}(S^t \oplus \emptyset_i) \\
- \sum_{i:\sigma_i \neq s_i^t} \bar{\gamma}'_{s_i^t}(\Omega \cup (S^t \oplus \emptyset_i \oplus \ldots \oplus \emptyset_k)) \Bigg).
\end{aligned}$$

Substituting, we get

$$\begin{aligned}
T \cdot \mathbf{OPT} \leq \sum_{i=1}^{T} \Bigg(2\bar{\gamma}(S^t) + \epsilon k - \sum_{i:\sigma_i = s_i^t} \bar{\gamma}'_{s_i^t}(S^t \oplus \emptyset_i) \\
- \sum_{i:\sigma_i \neq s_i^t} \bar{\gamma}'_{s_i^t}(\Omega \cup (S^t \oplus \emptyset_i \oplus \ldots \oplus \emptyset_k)) \Bigg),
\end{aligned}$$

which completes the proof. □

For nondecreasing γ, we get the following corollary:

Corollary 5.4.5. *If γ is nondecreasing, the price of total anarchy for valid games is asymptotically 2.*

The price of anarchy and the price of sinking are both brittle to the addition of Byzantine players. In contrast, for nondecreasing social welfare functions γ, our price of total anarchy result holds even in the presence of arbitrarily many Byzantine players. In any valid game, suppose players $1, \ldots, k$ are regret minimizing. Let $\mathbf{OPT} = \gamma(\Omega)$ be the optimal value for these players playing alone. Suppose there is some additional set of Byzantine players $\mathcal{B}$ that behave arbitrarily.

Theorem 5.4.6. *Consider a valid game with nondecreasing social welfare function γ, where the k regret minimizing players play $S^1, \ldots, S^T$ over T time steps while the Byzantine players play $B^1, \ldots, B^T$. Then the average social welfare $1/T \sum_{t=1}^{T} \gamma(S^t \cup B^t) \geq \mathbf{OPT}/2$.*

Proof. We observe that

$$\begin{aligned}
&\gamma(\Omega \cup B^t) \\
&\qquad \leq \gamma(\Omega \cup S^t \cup B^t) \\
&\qquad = \gamma(S^t \cup B^t) + \sum_{i:\sigma_i \neq s_i^t} \gamma'_{\sigma_i}(S^t \cup B^t \cup (\Omega \oplus \emptyset_i \oplus \ldots \oplus \emptyset_k)) \\
&\qquad \leq \gamma(S^t \cup B^t) + \sum_{i:\sigma_i \neq s_i^t} \gamma'_{\sigma_i}(S^t \oplus \emptyset_i \cup B^t),
\end{aligned}$$

where the first inequality follows because γ is nondecreasing, and the third follows from submodularity. We then have

$$\begin{aligned}
\mathbf{OPT} &\leq \gamma(\Omega \cup B^t) \\
&\leq \gamma(S^t \cup B^t) + \sum_{i:s_i \neq \sigma_i} \gamma'_{\sigma_i}(S^t \oplus \emptyset_i \cup B^t) \\
&\leq \gamma(S^t \cup B^t) + \sum_{i:s_i \neq \sigma_i} \alpha_i(S^t \oplus \sigma_i \cup B^t)
\end{aligned}$$

with the first line following because γ is nondecreasing, and the second from the Vickrey condition. Summing over T, this yields

$$T \cdot \mathbf{OPT} \leq \sum_{t=1}^{T} \gamma(S^t \cup B^t) + \sum_{t=1}^{T} \sum_{i:s_i \neq \sigma_i} \alpha_i(S^t \oplus \sigma_i \cup B^t).$$

Suppose $\sum_{t=1}^{T} \gamma(S^t \cup B^t) < T \cdot \mathbf{OPT}/2$. Since

$$\sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_i(S^t \cup B^t) \leq \sum_{t=1}^{T} \sum_{i=1}^{k+|\mathcal{B}|} \alpha_i(S^t \cup B^t) \leq \sum_{t=1}^{T} \gamma(S^t \cup B^t),$$

it must be that

$$\sum_{i=1}^{k} \sum_{t=1}^{T} \alpha_i(S^t \oplus \sigma_i \cup B^t) > \sum_{i=1}^{k} \sum_{t=1}^{T} \alpha_i(S^t \cup B^t),$$

and so there is some regret minimizing player i for whom $\sum_{t=1}^{T} \alpha_i(S^t \oplus \sigma_i \cup B^t) > \sum_{t=1}^{T} \alpha_i(S^t \cup B^t)$, violating the condition that he is regret minimizing. □

Note that here we have shown that in a valid game with a nondecreasing social utility function, if k players minimize regret and an arbitrary number of Byzantine players are *added* to the system, the resulting social welfare is no worse than half the optimal social welfare for k players. This is a slightly different result than we showed for Hotelling games, where we were able to guarantee that each regret-minimizing player obtains at least half of her fair share of the entire game, regardless of what the other $k - 1$ players do. On the other hand, for valid games one clearly cannot obtain half of the optimum social welfare for $k + |\mathcal{B}|$ players since the Byzantine players need not be acting in even their own interest.

Valid games and Hotelling games can both be used to model competition in markets; the main difference between them is the social utility function, with Hotelling games considering “fairness”, or minimum player utility, and with valid games explicitly constrained to have social utility at least the sum of the player utilities (and so unable to depend solely on the utility of the worst-off player). Another difference is the inherent symmetry of Hotelling games. One can, however, construct a Hotelling game quite similar to the market sharing game described at the beginning of this section: represent each market as a star graph. The size of the star corresponds to the the value of the market. Any player can play at the center of any star, and we can model budgets by allowing player i to play at the centers of c_i stars. With the fairness social utility function, this is a minor modification of a c-Hotelling game (one can also connect the stars but stipulate that the buyers will only travel at most distance one, to get a slightly modified generalized location game). Using, for example, a sum social utility function, this is a valid utility game. Because the player utility functions are the same in each, the techniques and outcomes of regret minimization are the same; only the analysis of *quality* of the outcomes differs.

5.5 Atomic Congestion Games

In this section, we show price of total anarchy results matching existing price of anarchy results for atomic, unweighted congestion games with social utility equal to the sum of the player utilities [6, 25]. We also consider the atomic congestion game of weighted load balancing with social utility equal to the makespan [29, 87, 88], and show matching results for two links, but demonstrate that for n links, the price of total anarchy is exponentially worse than the price of anarchy. Finally, we consider weighted load balancing with social utility equal to the sum of the player utilities [115], and show that for $k >> n$, the price of total anarchy is $1 + o(1)$. In the case of load balancing with sum social utility, our price of total anarchy results also imply previously unknown price of anarchy results for mixed strategies.

A congestion game is a minimization game consisting of a set of k players and, for each player i, a set $\mathcal{V}_i$ of facilities. Player i plays subsets of facilities from some feasible set $\mathcal{A}_i = \{a_i \subseteq \mathcal{V}_i : a_i \text{ is a feasible action}\}$. In *weighted* games, each player i has an associated weight w_i; in *unweighted* games, each player weight is 1. Each facility e has an associated latency function ℓ_e. A player i playing a_i experiences cost $\alpha_i = \sum_{e \in a_i} \ell_e(f_e)$ where f_e is the load on facility e: $f_e = \sum_{j : e \in a_i} w_j$.

5.5.1 Atomic congestion games with sum social utility

In this section, we consider unsplittable atomic selfish routing with unweighted players. The social utility function we consider in this section is the sum of the player costs, or $\gamma(A) = \sum_i \alpha_i(A)$. We write $\Omega = \{\sigma_1, \dots, \sigma_k\}$ for a strategy profile optimizing the social utility function γ. We write f_e^t for the load on edge e at time t, and f_e^* for the load on edge e in Ω.

We first consider linear edge costs of the form $\ell_e(f_e) = c_e f_e + b_e$ for edge e. In this setting, Christodoulou and Koutsoupias [25] and Awerbuch et al. [6] independently showed that the price of anarchy for pure strategies is 2.5. We show a matching bound for the price of total anarchy, which also implies the matching bound shown by Christodoulou and Koutsoupias [26] for the price of anarchy for mixed strategies and for correlated equilibria.

Theorem 5.5.1. *The price of total anarchy of atomic congestion games with unweighted players, sum social utility function, and linear cost functions is 2.5.*

Proof of Theorem 5.5.1. Let $\Omega = \{\sigma_1, \dots, \sigma_k\}$ be a strategy profile optimizing the social utility function so that $\bar{\gamma}(\Omega) = \mathbf{OPT}$ By the assumption of regret minimization, each player's time average cost is no more than the cost of her best fixed action in hindsight. In particular, it is no more than if she had played her part in the optimal strategy on every timestep: For all i,

$$\begin{aligned}\sum_{t=1}^{T} \bar{\alpha}_i(S^t) &= \sum_{t=1}^{T} \sum_{e \in s_i^t} c_e f_e^t + b_e \\ &\leq \sum_{t=1}^{T} \bar{\alpha}_i(S^t \oplus \sigma_i) \\ &\leq \sum_{t=1}^{T} \sum_{e \in \sigma_i} c_e(f_e^t + 1) + b_e.\end{aligned}$$

Summing over each player and rearranging the sum:

$$\begin{aligned}\sum_{t=1}^{T} \sum_{e \in E} \sum_{i \text{ s.t. } e \in s_i^t} c_e f_e^t + b_e &\leq \sum_{t=1}^{T} \sum_{e \in E} \sum_{i \text{ s.t. } e \in \sigma_i} c_e(f_e^t + 1) + b_e \\ &= \sum_{t=1}^{T} \sum_{e \in E} c_e f_e^t f_e^* + c_e f_e^* + b_e f_e^*.\end{aligned}$$

We now use a lemma also used by Awerbuch et al. [6]:

Lemma 5.5.2. *For $i, j > 0$ integers:*

1. $ij = \frac{1}{3}j^2 + \frac{3}{4}i^2 - \frac{1}{3}(j - \frac{3}{2}i)^2$
2. $\frac{9}{8}i^2 + \frac{3}{2}i - \frac{1}{2}(j - \frac{3}{2}i)^2 \leq \frac{5}{2}i^2$

We can apply part 1 of the lemma to get

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{e\in E}(c_e f_e^t + b_e) f_e^t\\ &\qquad \leq \sum_{t=1}^{T}\sum_{e\in E} c_e\left(\frac{1}{3}(f_e^t)^2 + \frac{3}{4}(f_e^*)^2 - \frac{1}{3}(f_e^t - \frac{3}{2}f_e^*)^2 + f_e^*\right) + b_e f_e^*.\end{aligned}$$

This is equivalent to

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{e\in E}(c_e f_e^t + \frac{3}{2}b_e) f_e^t\\ &\qquad \leq \sum_{t=1}^{T}\sum_{e\in E} c_e\left(\frac{9}{8}(f_e^*)^2 + \frac{3}{2}f_e^* - \frac{1}{2}(f_e^t - \frac{3}{2}f_e^*)^2\right) + \frac{3}{2}b_e f_e^*.\end{aligned}$$

This allows us to apply property 2 of the lemma to obtain

$$\begin{aligned}\sum_{t=1}^{T}\sum_{e\in E}(c_e f_e^t + b_e) f_e^t &\leq \sum_{t=1}^{T}\sum_{e\in E}\frac{5}{2}c_e (f_e^*)^2 + \frac{3}{2}b_e f_e^*\\ &\leq \frac{5}{2}\sum_{t=1}^{T}\sum_{e\in E}(c_e f_e^* + b_e) f_e^*,\end{aligned}$$

which proves the claim. □

Corollary 5.5.3 (Christodoulou and Koutsoupias [26]). *The price of anarchy of atomic congestion games with unweighted players, sum social utility function, and linear cost functions is 2.5, even for mixed strategies. The same bound also holds for correlated equilibria in this setting.*

We next consider polynomial latency functions and show a bound matching the price of anarchy shown by Christodoulou and Koutsoupias [25] and Awerbuch et al. [6] for mixed strategies.

Theorem 5.5.4. *The price of total anarchy of atomic congestion games with unweighted players, sum social utility function, and polynomial latency functions of degree d is at most $d^{d^{1-o(1)}}$.*

Proof. By the no-regret property we have for each player i:

$$\sum_{t=1}^{T}\sum_{e\in a_i^t}\ell_e(f_e^t) \leq \sum_{t=1}^{T}\sum_{e\in\sigma_i}\ell_e(f_e^t+1).$$

We may sum over each player:

$$\sum_{t=1}^{T}\sum_{e\in E}\sum_{i \text{ s.t. } e\in a_i^t}\ell_e(f_e^t) \leq \sum_{t=1}^{T}\sum_{e\in E}\sum_{i \text{ s.t. } e\in\sigma_i}\ell_e(f_e^t+1)$$

and rearrange the sums:

$$\sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t)f_e^t \leq \sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t+1)f_e^*.$$

We now apply a lemma used by Christodoulou and Koutsoupias [25]:

Lemma 5.5.5. *For $f(x)$ a polynomial with non-negative coefficients of degree d, and for every $x, y \geq 0$:*

$$y\cdot f(x+1) \leq \frac{x\cdot f(x)}{2} + \frac{C_0(d)\cdot y\cdot f(y)}{2},$$

where $C_0(d) = p^{p^{1-o(1)}}$.

Applying the lemma, we get

$$\begin{aligned}\sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t)f_e^t &\leq \sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t+1)f_e^* \\ &\leq \sum_{t=1}^{T}(\sum_{e\in E}\frac{f_e^t f(f_e^t)}{2} + \sum_{e\in E}\frac{C_0(d)f_e^* f(f_e^*)}{2}).\end{aligned}$$

Rearranging, we then get

$$\sum_{t=1}^{T}\sum_{e\in E}\ell_e(f_e^t)f_e^t \leq C_0(d)\sum_{t=1}^{T}\sum_{e\in E}f(f_e^*)f_e^*,$$

which completes the proof. □

5.5.2 Parallel link congestion game with makespan social utility

The parallel link congestion game models n identical links and k weighted players (jobs) who must choose which link to use. Each player pays the sum of the weights of the jobs on the link she chose. The social cost for this game is defined as the total weight on the worst-loaded link. This game was the main focus of the Koutsoupias and Papadimitriou paper that introduced the concept of the price of anarchy [87].

More formally, this is a minimization game where for each player i, the feasible actions are $\mathcal{A}_i = \{1, \ldots n\}$. The social utility function is $\gamma(A) = \max_{j\in\{1,\ldots n\}}\sum_{i:a_i=j} w_i$.

Koutsoupias and Papadimitriou [87] proved that the price of anarchy of the parallel link congestion game with two links is $3/2$. Two groups of researchers [31, 88] later proved that the price of anarchy when there are n links is $\Theta(\log n/\log\log n)$.

In this section, we show a matching bound on the price of total anarchy for 2 links. We also show that for n links, the price of total anarchy *does not* match the price of anarchy.

Theorem 5.5.6. *The price of total anarchy of the parallel link congestion game with makespan social utility and two links is $3/2$, exactly matching the price of anarchy.*

The proof parallels that in the original Koutsoupias and Papadimitriou paper [87]. It is subtler because regret-minimizing algorithms only give a guarantee in expectation, on average, and make no guarantees about the performance on any given day.

Proof of Theorem 5.5.6. Denote by q_i the expected probability that player i is on the maximally loaded machine (breaking ties between equally loaded machines at random). Note that the expected social cost is then $\bar{\gamma}(S) = \sum_{i=1}^{k} q_i w_i$. By the regret-minimizing property, for all players, $\frac{1}{T}\sum_{t=1}^{T} \bar{\alpha}_i(S^t) \leq w_i + \epsilon + \frac{1}{T}\sum_{t=1}^{T} \frac{\sum_{h\,:\,h\neq i} w_h}{2}$.

Define p_{ij} to be the expected probability that player i selects machine j; c_{ih} is the expected probability that players i and h select the same machine. Then for any fixed i,

$$\begin{aligned}
\sum_{h:h\neq i} (q_i + q_h) w_h &\leq \sum_{h:h\neq i} (1 + c_{ih}) w_h \\
&\leq \sum_{h:h\neq i} w_h + \sum_{h:h\neq i} c_{ih} w_h \\
&= \sum_{h:h\neq i} w_h + \sum_{h:h\neq i} (p_{i1}p_{h1}w_h + p_{i2}p_{h2}w_h).
\end{aligned}$$

Note that for any player i, regardless of her strategy, her cost is

$$\bar{\alpha}_i(S) = w_i + p_{i1} \sum_{h:h\neq i} p_{h1} w_h + p_{i2} \sum_{h:h\neq i} p_{h2} w_h$$

by definition. This relationship is essentially Lemma 1 of [87]; however they only note that it holds for Nash equilibrium strategies. This gives us $\sum_{h:h\neq i}(q_i + q_h) w_h \leq \sum_{h:h\neq i} w_h + \bar{\alpha}_i(S) - w_i$. Averaging over time, this is

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{h:h\neq i} (q_i^t + q_h^t) w_h \leq \frac{1}{T}\sum_{t=1}^{T} \sum_{h:h\neq i} w_h + \frac{1}{T}\sum_{t=1}^{T} \bar{\alpha}_i(S^t) - w_i.$$

Using the fact that player i obtains low regret, we then have

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{h:h\neq i} (q_i^t + q_h^t) w_h \leq \frac{1}{T}\sum_{i=1}^{T} \sum_{h:h\neq i} w_h + \epsilon + \frac{1}{T}\sum_{t=1}^{T} \frac{\sum_{h\,:\,h\neq i} w_h}{2}.$$

Rearranging, this yields for any fixed i

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\bar{\gamma}(S^t) \\
&\quad = \frac{1}{T}\sum_{t=1}^{T}\sum_{h=1}^{k} q_h^t w_h \\
&\quad \leq \frac{1}{T}\sum_{t=1}^{T}\left(\frac{3}{2}\sum_{h:h\neq i} w_h + q_i^t w_i - \sum_{h\neq i} q_i^t w_h\right) + \epsilon \\
&\quad = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{3}{2}\sum_{h=1}^{k} w_h - \frac{3w_i}{2} + q_i^t w_i - q_i^t \sum_{h=1}^{k} w_h + q_i^t w_i\right) + \epsilon \\
&\quad = \frac{1}{T}\sum_{t=1}^{T}\left(\left(\frac{3}{2} - q_i^t\right)\sum_{h=1}^{k} w_h + \left(2q_i^t - \frac{3}{2}\right) w_i\right) + \epsilon.
\end{aligned}$$

Note that $\mathbf{OPT} \geq \max\{\frac{1}{2}\sum_{h=1}^{k} w_h, w_i\}$ for any i. If for all agents i, $\frac{1}{T}\sum_{t=1}^{T} q_i^t \leq \frac{3}{4}$, then

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\bar{\gamma}(S^t) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{h=1}^{k} q_h^t w_h \\
&= \sum_{h}\left(w_h \frac{1}{T}\sum_{t} q_h^t\right) \\
&\leq \frac{3}{4}\sum_{h} w_h \\
&\leq \frac{3}{2}\mathbf{OPT}.
\end{aligned}$$

Otherwise, there exists some agent i such that $\frac{1}{T}\sum_{t=1}^{T} q_i^t > \frac{3}{4}$ and thus

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\bar{\gamma}(S^t) \\
&\quad \leq 2\mathbf{OPT}\frac{1}{T}\sum_{t}(\frac{3}{2} - q_i^t) + \mathbf{OPT}\frac{1}{T}\sum_{t}(2q_i^t - \frac{3}{2}) + \epsilon \\
&\quad = \frac{3}{2}\mathbf{OPT} + \epsilon,
\end{aligned}$$

as desired. □

For the parallel link congestion game with n links, the price of total anarchy diverges from the price of anarchy. This divergence stems from the fact that in the parallel links game, the social cost function γ is defined in terms of expected maximum *link* latency, whereas individual utility is a function of average *job* latency.[5] In the single stage Nash equilibrium analyzed for price of anarchy results, the two values are related: expected job latency for player i is equal to the average link latency of every link in the support of i's mixed strategy. In a Nash equilibrium, therefore, maximum expected link latency must be low, and with tail bounds, it is straightforward to argue that the expected maximum link latency cannot be too high [31]. Over an arbitrary sequence of regret-minimizing plays, however, average job latency no longer necessarily corresponds to the average latency of any link. This is demonstrated by a cycling example we use in the proof of the following theorem:

Theorem 5.5.7. *The price of total anarchy in the parallel link game with makespan social utility and n links is $\Omega(\sqrt{n})$.*

Proof of Theorem 5.5.7. Consider n parallel links $1, \ldots, n$, and n players all with unit weights $w_i = 1$. Clearly, $\mathbf{OPT} = 1$. Define a sequence of plays $A^1, \ldots, A^T$ as follows: Divide the players into $2\sqrt{n}$ groups $G_0, \ldots, G_{2\sqrt{n}-1}$, each of size $\sqrt{n}/2$. At time t, all players in $G_{(t \mod 2\sqrt{n})}$ play on link 1, and all other players play over links $2 + (t \mod n-1), 2 + (t+1 \mod n-1), \ldots, 2 + (t + n - \sqrt{n}/2 - 1 \mod n-1)$ so that there is exactly one player on each link (ordering may be arbitrary). Then each player experiences average latency $\frac{1}{T}\sum_{t=1}^{T} \alpha_i(A^t) = \frac{1}{2\sqrt{n}} \cdot \frac{\sqrt{n}}{2} + \frac{2\sqrt{n}-1}{2\sqrt{n}} \cdot 1 = \frac{5}{4} - \frac{1}{2\sqrt{n}}$. Consider the latency experienced by player i if she were to play at any fixed node. Given the sequence of plays described above, every node $v \geq 2$ is occupied by some player $h \neq i$ on an $(n - \sqrt{n}/2 - 1)/(n-1)$ fraction of time steps. Since player i always pays for her own weight, she expects to experience latency $2 \cdot \frac{n - \frac{\sqrt{n}}{2} - 1}{n-1} + 1 \cdot \frac{\sqrt{n}}{2(n-1)} = 2 - \frac{\sqrt{n}}{2(n-1)}$. Therefore, for sufficiently large n, all players experience negative regret. Nevertheless, at every time step, the maximum latency is $\Omega(\sqrt{n})$. □

5.5.3 Parallel links congestion game with sum social utility

We have just shown that the price of total anarchy does not match the $O(\log n / \log\log n)$ price of anarchy for the load balancing game with the makespan social utility function. The results in Section 5.5.1, however, imply a price of total anarchy ≤ 2.5 for the load balancing game with the *sum* social utility function (since load balancing is a special case of routing), even for mixed strategies and different server speeds. In fact, we can show more: in this section, we show that so long as $k >> n$ and the server speeds are relatively bounded, the price of total anarchy is $1 + o(1)$. This matches a price of anarchy result shown by Suri et al. [115] for pure strategy equilibria. Our theorem below implies an equivalent price of anarchy result even for mixed strategy equilibria.

Theorem 5.5.8. *In the load balancing game with sum social cost and linear latency functions, the price of total anarchy is $1 + o(1)$ provided that $k >> n$ and server speeds are relatively bounded.*

[5] Note that if we were to redefine the social cost function γ for the parallel links game to be the maximum expected *job* latency, it is simple to verify that the resulting price of total anarchy is 2: Rescale the weights so that $\mathbf{OPT} = 1$. Total weight is $\leq n$, and $w_i \leq 1$ for all players. Over any sequence of plays, there must be some link with average latency $l \leq 1$. Therefore, every player i is guaranteed to experience average latency in expectation at most $l + w_i + \epsilon \leq 2 + \epsilon$.

Proof of Theorem 5.5.8. By the no regret property, for each player i,

$$\sum_{t=1}^{T} \bar{\alpha}_i(A^t) = \sum_{t=1}^{T} \frac{l_{a_i^t}}{\pi_{a_i^t}} \leq \sum_{t=1}^{T} \bar{\alpha}_i(A^t \oplus \sigma_i) \leq \sum_{t=1}^{T} \frac{l_{\sigma_i}^t + 1}{\pi_{\sigma_i}}.$$

Summing over all players and reordering the sum, we get

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^t)^2}{\pi_e} &\leq \sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^t + 1) \cdot f_e^*}{\pi_e} \\
&\leq \sum_{t=1}^{T} \sum_{e \in E} \frac{1}{\pi_e} \left(\frac{(f_e^t)^2 + (f_e^*)^2}{2} + f_e^* \right),
\end{aligned}$$

where the second inequality follows from the fact that $a \cdot b \leq \frac{a^2+b^2}{2}$. Subtracting, we get

$$\begin{aligned}
\frac{1}{2} \sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^t)^2}{\pi_e} &\leq \sum_{t=1}^{T} \sum_{e \in E} \frac{1}{\pi_e} \left(\frac{(f_e^*)^2}{2} + f_e^* \right) \\
\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^t)^2}{\pi_e} &\leq \sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^*)^2 + 2f_e^*}{\pi_e}.
\end{aligned}$$

Combining these inequalities, we can bound the price of total anarchy:

$$\begin{aligned}
\frac{\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^t)^2}{\pi_e}}{\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^*)^2}{\pi_e}} &\leq \frac{\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^*)^2 + 2f_e^*}{\pi_e}}{\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^*)^2}{\pi_e}} \\
&= 1 + 2 \frac{\sum_{t=1}^{T} \sum_{e \in E} \frac{f_e^*}{\pi_e}}{\sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^*)^2}{\pi_e}} \\
&\leq 1 + 2 \sum_{t=1}^{T} \frac{\sum_{e \in E} \frac{f_e^*}{\pi_e}}{\sum_{e \in E} \frac{(f_e^*)^2}{\pi_e}}.
\end{aligned}$$

We then use the following technical lemma of Suri et al. [115]

Lemma 5.5.9. *Let n, k be positive integers and $f_e \geq 0, \pi_e > 0$ be reals such that $\sum_{e \in E} f_e = k$. Then*

$$\frac{\sum_{e \in E} f_e / \pi_e}{\sum_{e \in E} f_e^2 / \pi_e} \leq (1 + \sqrt{\max_{1 \leq i,j \leq n} \frac{\pi_i}{\pi_j}}) \frac{n}{2k}$$

.

This gives us

$$\frac{1}{2} \sum_{t=1}^{T} \sum_{e \in E} \frac{(f_e^t)^2}{\pi_e} \leq 1 + 2 \left(1 + \sqrt{\max_{1 \leq i,j \leq n} \frac{\pi_i}{\sigma_j}} \right) \frac{n}{2k}.$$

This is $1 + o(1)$ in k when $k >> n$, which completes the proof. □

Corollary 5.5.10. *In the load balancing game with sum social cost and linear latency functions, the price of anarchy is $1 + o(1)$ provided that $k >> n$ and server speeds are relatively bounded, even for mixed strategies.*

5.6 Algorithmic efficiency

In the Hotelling games we analyzed in Section 5.3, each player has only n strategies—the n nodes in the graph. In such settings, the weighted majority algorithm [91] runs in polynomial time and minimizes regret. Similarly, in the parallel links congestion game, there are n strategies—the n links—and thus minimizing regret is relatively straightforward.

In valid games, if the set of actions available to a player is polynomial in $|\mathcal{V}_i|$, the action groundset, then once again, weighted majority can be used to minimize regret. However, in arbitrary valid games, the action space for player i could be as large as $2^{|\mathcal{V}_i|}$. In such situations, if the player's private utility is a linear function of the elements of the groundset she obtains and she can compute exact best responses in polynomial time (such as in the market sharing game of Goemans et al. [65]), then she can use results of Kalai and Vempala [81] to minimize regret in polynomial time. If her utility function is linear, but she can only compute approximate best responses, results of Kakade et al. [80] allow her to *approximately* minimize regret; that is, she obtains expected average cost close to β times the cost of the best fixed solution in hindsight, where β is the approximation ratio of her optimizer. We can modify our proof of the price of total anarchy to carry this β through and show:

Theorem 5.6.1. *The price of β-minimizing regret in valid games is $1 + \beta$.*

If the player's utility function is convex and well-defined over the convex hull of her pure strategies and she furthermore has the ability to project points in space onto that convex hull, then she can use an algorithm developed by Zinkevich [122] to minimize her regret. In situations where no existing techniques are a perfect fit, more specialized regret-minimizing algorithms for specific games may also be developed.

5.7 Conclusions

In this chapter, we propose regret minimization as a definition of selfish behavior in repeated games. We consider four general classes of games—generalized Hotelling games, valid games, and atomic congestion games with two different social utility functions—and show that the price of total anarchy exactly matches the price of anarchy in most cases, but there is a gap of $\Omega(\sqrt{n})$ versus $O(\frac{\log n}{\log \log n})$ in the case of n parallel links. Our results hold even in games where regret-minimizing algorithms can cycle and fail to converge to an equilibrium. We also prove results in Byzantine settings when only some of the players achieve regret minimization and the other players are allowed to act in an arbitrary fashion. In addition, our results for weighted load balancing with player-summed social utility functions imply new price of anarchy results for mixed strategies.

Chapter 6

Conclusions and Directions for Future Work

The goal of the work presented in this thesis is to advance our understanding of the outcomes of selfish behavior in games. In support of that aim, we propose regret minimization as a descriptive criterion for selfishness (as opposed to prescriptive characterizations such as the direct study of static notions of equilibrium), and present new regret minimizing algorithms.

The Price of Anarchy has been proposed as a tool for understanding selfish behavior. However, in 2-player, n-action games, Nash equilibria are PPAD-hard to compute [23]. In *any* game with a polynomial number of actions, though, one can run regret-minimizing algorithms. (One can also do so efficiently in many settings with even an exponential number of actions.) Many games only admit mixed Nash equilibria, and there is no immediate incentive for players to play their given mixed strategy as opposed to any one of the pure strategies in the support of the mixed strategy. In addition, there is no reason to assume in general games that agents demonstrating selfish behavior should *converge* to a Nash equilibrium.

Another line of work seeks to develop algorithms that, when played against each other, approach equilibria. Many such results require a centralized authority, and nearly all of them require that all (or nearly all) of the players play particular algorithms that prescribe particular choices at every step in time. In this thesis we sidestep the issue of equilibria and instead analyze the performance of strategies that may or may not reach equilibrium and are able to show results even in situations where such strategies may cycle. In addition, the results we present hold whenever players choose strategies that in hindsight achieve low regret. We do not require that players all use the same algorithm, or that they employ particular algorithms to achieve this guarantee. In fact, our results hold even in situations where players do not use regret-minimizing algorithms, but where the strategies they employ happen to have yielded low regret in hindsight for the particular sequence of events they experienced.

Because we place a weaker assumption on the agents' algorithms, there are more algorithms, simpler algorithms, and more efficient algorithms for regret minimization than for more demanding guarantees such as internal regret minimization. In particular, efficient internal regret minimizing procedures are not known for many of the game settings we consider, such as routing. Further, we are able to prove guarantees even in Byzantine settings, where not all players behave rationally; such settings need not correspond to correlated equilibria.

In addition to the open problems mentioned in each chapter of the thesis, we sketch some additional directions for future work here.

One direction for future work is the analysis of the outcomes of regret-minimizing play in additional classes of games. For example, auction settings are a natural application for this approach, and the results of such inquiry could have consequences for the design of bidding strategies and of auction mechanisms.

As we show in this thesis, in some natural classes of games, regret minimization is such a minimal assumption that it cannot prevent agents from colluding to do poorly, causing social costs much worse than those of the worst Nash equilibrium. How can we categorize and study the classes of games in which this occurs? Are there additional simple assumptions one can make on the agent algorithms or on the underlying game that prevent this sort of collusion? One approach to this set of questions is the exploration of various models of noise and perturbation in games. Noisy models may be useful for smoothing out pathological game outcomes to yield simple models of behavior with even better social utility guarantees. Such an approach is motivated by the view that noise is not only a useful theoretical tool, but a necessary component of any realistic model of large, real-world games.

Finally, this work motivates further study of the interactions between adversarial and selfish agents and resulting impacts on social welfare. When studying distributed systems of heterogeneous agents, it is vital to consider players who behave unpredictably, and yet previous models of selfishness are often quite brittle to such behavior.

Bibliography

[1] J. Abernathy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2008. 3.1.3

[2] H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM*, 2008. 2.1.1

[3] Nir Andelman, Michal Feldman, and Yishay Mansour. Strong price of anarchy. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 189–198. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2007. 2.4

[4] B. Awerbuch and R. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, 2004. 1.1.1, 2.2.4, 2.2.4, 3.1, 3.1.3, 3.3, 3.4, 3.4, 4.1, 4.2.2

[5] B. Awerbuch, Y. Azar, A. Epstein, V.S. Mirrokni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 264–273. ACM New York, NY, USA, 2008. 2.2.4, 2.3

[6] Baruch Awerbuch, Yossi Azar, and Amir Epstein. The price of routing unsplittable flow. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 57–66. ACM New York, NY, USA, 2005. 1.1.2, 5.1, 5.1.1, 5.5, 5.5.1, 5.5.1, 5.5.1

[7] Baruch Awerbuch, Yossi Azar, Yossi Richter, and Dekel Tsur. Tradeoffs in worst-case equilibria. *Theor. Comput. Sci.*, 361(2):200–209, 2006. ISSN 0304-3975. 2.4

[8] Y. Azar. *On-line Load Balancing Online Algorithms - The State of the Art*, chapter 8, pages 178–195. Springer, 1998. 4.1

[9] Moshe Babaioff, Robert Kleinberg, and Christos H. Papadimitriou. Congestion games with malicious players. In *EC*, 2007. 1

[10] Maria-Florina Balcan and Avrim Blum. Approximation algorithms and online mechanisms for item pricing. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*, 2006. 3.1, 3.1.1

[11] P. Bartlett, V. Dani, T. P. Hayes, S. M. Kakade, A. Rakhlin, and A. Tewari. High probability regret bounds for bandit online optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2008. 3.1.3

[12] M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Trans-*

portation. Yale University Press, 1956. 4.2.2, 4.2.3

[13] Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul Goldberg, Zengjian Hu, and Russell Martin. Distributed selfish load balancing. In *SODA*, 2006. 2.2.4

[14] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956. 2.2.4

[15] Avrim Blum, Eyal Even-Dar, and Katrina Ligett. Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 45–52. ACM New York, NY, USA, 2006. ISBN 1-59593-384-0. 1.1.2

[16] Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. Regret minimization and the price of total anarchy. In *Proceedings of the fourtieth annual ACM symposium on Theory of computing*, pages 373–382. ACM New York, NY, USA, 2008. 1.1.2, 4.8

[17] Lawrence E. Blume. The statistical mechanics of best-response strategy revision. *Games and Economic Behavior*, 11(2):111–145, November 1995. 2.2.1

[18] G.W. Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13:374–376, 1951. 2.2.2

[19] N. Cesa-Bianchi, Y. Freund, D.P. Helmbold, D. Haussler, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997. 2.2.4

[20] Deeparnab Chakrabarty, Aranyak Mehta, and Vijay Vazirani. Design is as easy as optimization. In *33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2006. 6

[21] X. Chen and X. Deng. 3-Nash is PPAD-complete. Technical Report 134, Electronic Colloquium on Computational Complexity, 2005. 2.1.1

[22] X. Chen, X. Deng, and S.H. Teng. Computing Nash Equilibria: Approximation and Smoothed Complexity. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 603–612, 2006. 2.1.1

[23] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS06)*, pages 261–272, 2006. 1, 2.1.1, 6

[24] S. Chien and A. Sinclair. Convergence to approximate nash equilibria in congestion games. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 169–178. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2007. 2.2.4, 5.3.2

[25] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, pages 67–73. ACM New York, NY, USA, 2005. 1.1.2, 5.1, 5.1.1, 5.5, 5.5.1, 5.5.1, 5.5.1

[26] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. *ESA*, pages 59–70, 2005. 5.5.1, 5.5.3

[27] C. Chung, K. Ligett, K. Pruhs, and A. Roth. The Price of Stochastic Anarchy. In *Proceedings of the First International Symposium on Algorithmic Game Theory*, pages 303–314. Springer, 2008. 2.4

[28] J.R. Current and J.E. Storbeck. A multiobjective approach to design franchise outlet networks. *Journal of the Operational Research Society*, pages 71–81, 1994. 5.3.1

[29] A. Czumaj and B. Vöcking. Tight bounds on worse case equilibria. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 413–420, 2002. 2.1.1, 6, 5.5

[30] A. Czumaj and B. Vöcking. Tight bounds on worse case equilibria. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 413–420, 2002. 1.1.2

[31] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms*, 3(1), 2007. 5.5.2, 5.5.2

[32] A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proceedings of the 34th Symposium on Theory of Computing*, pages 287–296, 2002. 2.1.1

[33] A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proceedings of the 34th Symposium on Theory of Computing*, pages 287–296, 2002. 1.1.2

[34] V. Dani, T. P. Hayes, and S. M. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 20 (NIPS)*, 2007. 3.1.3

[35] Varsha Dani and Thomas P. Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 937–943. ACM New York, NY, USA, 2006. 1.1.1, 2.2.4, 3.1, 3.1.3, 3.4, 3.4

[36] C. Daskalakis and C.H. Papadimitriou. Three-player games are hard. Technical Report 139, Electronic Colloquium on Computational Complexity, 2005. 2.1.1

[37] C. Daskalakis and C.H. Papadimitriou. Discretized multinomial distributions and nash equilibria in anonymous games. In *Proceedings of FOCS*, 2008. 2.1.1

[38] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *STOC '06*, 2006. 2.1.1

[39] T. Drezner. Location of multiple retail facilities with limited budget constraintsin continuous space. *Journal of Retailing and Consumer Services*, 5(3):173–184, 1998. 5.3.3

[40] T. Drezner, Z. Drezner, and S. Salhi. Solving the multiple competitive facilities location problem. *European Journal of Operational Research*, 142(1):138–151, 2002. 5.3.3

[41] Z. Drezner. Competitive location strategies for two facilities. *Regional Science and Urban Economics*, 12(4):485–493, 1982. 5.3.3

[42] J. Dunagan and S. Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. *Mathematical Programming*, 114(1):101–114, 2008. 3.3.4

[43] Glenn Ellison. Basins of attraction, long-run stochastic stability, and the speed of step-by-

step evolution. *Review of Economic Studies*, 67(1):17–45, January 2000. 2.2.1

[44] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–781, 2005. 2.2.4

[45] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *30th International Conference on Automata, Languages and Programming (ICALP)*, pages 502–513, 2003. 2.2.4

[46] E. Even-Dar, Y. Mansour, and U. Nadav. On the Convergence of Regret Minimization Dynamics in Concave Games. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, 2009. 2.3

[47] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the twenty-second annual symposium on Principles of Distributed Computing (PODC)*, pages 347–351. ACM Press, 2003. 1.1.2, 2.1.1

[48] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *STOC*, 2004. 2.1.1

[49] Amos Fiat, Haim Kaplan, Meital Levy, and Svetlana Olonetsky. Strong price of anarchy for machine load balancing. In *ICALP*, 2007. 2.4

[50] Simon Fischer and Berthold Vöcking. On the evolution of selfish routing. In *Proceedings of the 12th European Symposium on Algorithms (ESA)*, pages 323–334, 2004. 2.2.1, 2.2.4

[51] Simon Fischer and Berthold Vöcking. Adaptive routing with stale information. In *Proceedings of the 24th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, 2005. 2.2.4

[52] Simon Fischer, Harald Raecke, and Berthold Vöcking. Fast convergence to wardrop equilibria by adaptive sampling methods. In *Proceedings of 38th ACM Symposium on Theory of Computing (STOC)*, 2006. 2.2.1, 2.2.4, 4.2.2

[53] A.D. Flaxman, A.T. Kalai, and H.B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005. 3.4

[54] D. Foster and P. Young. Stochastic evolutionary game dynamics. *Theoret. Population Biol.*, 38:229–232, 1990. 2.2.1

[55] Dean P. Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 1997. 2.2.4

[56] D.P. Foster and H.P. Young. On the impossibility of predicting the behavior of rational agents. *Proceedings of the National Academy of Sciences*, 98(22):12848, 2001. 2.2.2

[57] D.P. Foster and H.P. Young. Learning, hypothesis testing, and Nash equilibrium. *Games and Economic Behavior*, 45(1):73–96, 2003. 2.2.2

[58] D.P. Foster and H.P. Young. Regret testing: learning to play Nash equilibrium without knowing you have an opponent. *Theoretical Economics*, 1(3):341–367, 2006. 2.2.3

[59] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. 1.1.1, 2.2.4

[60] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999. 2.2.4, 4.1.1

[61] Jean J. Gabszewicz and Jacques-Francois Thisse. Location. In R.J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 1, chapter 9. Elsevier Science Publishers (North-Holland), 1992. 5.3

[62] F. Germano and G. Lugosi. Global Nash convergence of Foster and Young's regret testing. *Games and Economic Behavior*, 60(1):135–154, 2007. 2.2.3

[63] Michel Goemans, Vahab Mirrokni, and Adrian Vetta. Sink equilibria and convergence. In *FOCS*, pages 142–154, 2005. ISBN 0-7695-2468-0. 1.1.2, 2.2.4, 5.1, 5.4.1

[64] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42 (6):1115–1145, 1995. ISSN 0004-5411. 3.1.1

[65] MX Goemans, L. Li, VS Mirrokni, and M. Thottan. Market sharing games applied to content distribution in ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24(5):1020–1033, 2006. 1.1.2, 5.1, 5.4.1, 5.4.1, 5.6

[66] Paul W. Goldberg. Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In *PODC*, 2004. 2.2.4

[67] P.W. Goldberg and C.H. Papadimitriou. Reducibility among equilibrium problems. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 61–70. ACM New York, NY, USA, 2006. 2.1.1

[68] A. Greenwald, Z. Li, and C. Marks. Bounds for regret-matching algorithms. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2005. 2.2.4

[69] J.F. Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A.W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume III, pages 97–139. Princeton University Press, 1957. 1.1.1, 2.2.4, 3.1

[70] J.C. Harsanyi. Games with incomplete information played by" Bayesian" players, I-III. Part I. The basic model. *Management science*, pages 159–182, 1967. 2.2.2

[71] S. Hart and Y. Mansour. The communication complexity of uncoupled nash equilibrium procedures. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 345–353. ACM New York, NY, USA, 2007. 2.1.1

[72] S. Hart and A. Mas-Colell. A General Class of Adaptive Strategies. *Journal of Economic Theory*, 98:2654, 2001. 2.2.4

[73] S. Hart and A. Mas-Colell. Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, 93(5):1830–1836, 2003. 2.2.3

[74] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated

equilibrium. *Econometrica*, 2000. 2.2.4

[75] Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929. ISSN 0013-0133. 1.1.2, 5.3

[76] D.L. Huff. Defining and estimating a trading area. *The Journal of Marketing*, pages 34–38, 1964. 5.3.3

[77] D.L. Huff. A programmed solution for approximating an optimum retail location. *Land Economics*, pages 293–303, 1966. 5.3.3

[78] A. Jafari, A. Greenwald, D. Gondek, and G. Ercal. On no-regret learning, fictitious play, and nash equilibrium. In *ICML*, pages 226–223, 2001. 2.2.4

[79] Jens Josephson and Alexander Matros. Stochastic imitation in finite games. *Games and Economic Behavior*, 49(2):244–259, November 2004. 2.2.1, 2.4

[80] Sham Kakade, Adam Tauman Kalai, and Katrina Ligett. Playing games with approximation algorithms. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 546–555. ACM Press New York, NY, USA, 2007. 1.1.1, 5.6

[81] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005. ISSN 0022-0000. 1.1.1, 2.2.4, 3.1, 3.1.1, 3, 4.1, 4.3.2, 4.4.4, 4.8, 5.6

[82] E. Kalai and E. Lehrer. Rational learning leads to Nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 1019–1045, 1993. 2.2.2

[83] Michihiro Kandori, George J. Mailath, and Rafael Rob. Learning, mutation, and long run equilibria in games. *Econometrica*, 61(1):29–56, January 1993. 2.2.1

[84] M. Kilkenny and J.F. Thisse. Economics of Location: A Selective Survey. *Computers and Operations Research*, 26(14):1369–1394, 1999. 5.3

[85] R. Kleinberg. Anytime algorithms for multi-armed bandit problems. In *Proceedings of the seventeenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 928–936. ACM New York, NY, USA, 2006. 1.1.1, 2.2.4

[86] Robert Kleinberg, Georgios Pillouras, and Eva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, 2009. 2.3

[87] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413. Springer, 1999. 1, 1.1.2, 2.1.1, 5.1, 5.1.1, 5.5, 5.5.2, 5.5.2

[88] E. Koutsoupias, M. Mavronikolas, and P. Spirakis. Approximate equilibria and ball fusion. *ACM Transactions on Computer Systems*, 36(6):683–693, 2003. 5.5, 5.5.2

[89] H. W. Kuhn. Extensive games and the problem of information. In H.W. Kuhn and A. W. Tucker, editors, *Contributions to the Theor of Games*, number 28 in Annals of Mathematics Studies II, pages 193–216. Princeton University Press, 1953. 2.2.2

[90] Samuelson Larry. Stochastic stability in games with alternative best replies. *Journal of Economic Theory*, 64(1):35–65, October 1994. 2.2.1

[91] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994. 1.1.1, 2.2.4, 5.6

[92] H.B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, pages 109–123, 2004. 1.1.1, 2.2.4, 2.2.4, 3.1, 3.1.3, 3.4, 4.1, 4.2.2

[93] N. Megiddo, E. Zemel, and S.L. Hakimi. The maximum coverage location problem. *SIAM Journal on Algebraic and Discrete Methods*, 4:253, 1983. 5.3.3

[94] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, 1996. 2.2.4

[95] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and economic behavior*, 13(1):111–124, 1996. 2.1

[96] I. Milchtaich. Generic uniqueness of equilibrium in large crowding games. *Mathematics of Operations Research*, 25(3):349–364, 2000. 4.2.3

[97] Vahab S. Mirrokni and Adrian Vetta. Convergence issues in competitive games. In *APPROX-RANDOM*, 2004. 2.2.4

[98] K. Miyasawa. On the Convergence of the Learning Process in a 2 X 2 Non-Zero-sum Two-person Game. Technical Report 33, Princeton University, 1961. 2.2.2

[99] D. Monderer and L.S. Shapley. Potential games. *Games and economic behavior*, 14(1): 124–143, 1996. 2.1

[100] D. Monderer and L.S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68(1):258–265, 1996. 2.2.2

[101] John Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. 2.1

[102] Abraham Neyman. Correlated equilibrium and potential games. *International Journal of Game Theory*, 26:223–227, 1997. 2.2.4

[103] Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *J. ACM*, 55(3):1–29, 2008. ISSN 0004-5411. 2.1.1

[104] H. Robbins. Some aspects of the sequential design of experiments. In *Bulletin of the American Mathematical Society*, volume 55, 1952. 3.1

[105] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951. 2.2.2

[106] Arthur J. Robson and Fernando Vega-Redondo. Efficient equilibrium selection in evolutionary games with random matching. *Journal of Economic Theory*, 70(1):65–92, July 1996. 2.2.1

[107] R.W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973. 2.1

[108] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2): 236–259, 2002. 1.1.2, 2.1.1, 2.2.1, 5.4.1

[109] T. Roughgarden and E. Tardos. Bounding the inefficiency of equilibria in nonatomic

congestion games. *Games and Economic Behavior*, 47(2):389–403, 2004. 4.6, 4.6, 4.6.3

[110] Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, 2009. 2.3

[111] W. Sandholm. Potential games with continuous player sets. *Journal of Economic Theory*, 97:81–108, 2001. 2.2.1

[112] D. Schmeidler. Equilibrium points of nonatomic games. *Journal of Statistical Physics*, 7 (4):295–300, 1973. 4.2.2

[113] L.S. Shapley et al. Some topics in two-person games. *Advances in game theory*, 52:1–29, 1964. 2.2.2

[114] A. Skopalik and B. Vöcking. Inapproximability of pure Nash equilibria. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 355–364. ACM New York, NY, USA, 2008. 2.1.1, 2.2.4

[115] S. Suri, C.D. Toth, and Y. Zhou. Selfish Load Balancing and Atomic Congestion Games. *Algorithmica*, 47(1):79–96, 2007. 5.5, 5.5.3, 5.5.3

[116] Siddarth Suri. Computational evolutionary game theory. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007. 2.2.1

[117] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, Lecture Notes in Artificial Intelligence. Springer, 2002. 2.2.4

[118] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *FOCS*, page 416, 2002. ISBN 0-7695-1822-2. 1.1.2, 5.1, 5.1.1, 5.4.1, 5.4.1, 5.4.2

[119] H Peyton Young. The evolution of conventions. *Econometrica*, 61(1):57–84, January 1993. 2.2.1

[120] H.P. Young. *Strategic learning and its limits*. Oxford University Press, 2004. 2.2.4

[121] H.P. Young. The possible and the impossible in multi-agent learning. *Artificial Intelligence*, 171(7):429–433, 2007. 2.2.2, 2.2.3

[122] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003. 1.1.1, 2.2.4, 3.1, 3.1.3, 3.3.1, 4.1, 5.6

[123] Martin Zinkevich. Theoretical guarantees for algorithms in multi-agent settings. Technical Report CMU-CS-04-161, Carnegie Mellon University, 2004. 4.1.1