Algorithms and Complexity of Problems Arising from Strategic Settings

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Themistoklis Melissourgos

November 2019

Abstract

Different strategic environments enforce different rules and incentives to the interacting entities, and as a result, they need to be analysed through environment-specific models. For each environment, suitable concepts of stability that capture the required properties have been defined over the years. For example, an appropriate solution concept in an evolutionary environment is the Evolutionarily Stable Strategy (ESS); in an environment modelled as a normal-form game is the Nash Equilibrium (NE); and for some environments where we require fair division of an object is a Consensus Halving.

In this thesis we study such solution concepts from a computational point of view. Given a pair of environment and its solution concept, we are interested in answering if every instance of this environment admits a solution. If the answer is no, we investigate the computational complexity of deciding the existence of a solution for a given instance. If the answer is yes, we try to determine the complexity of computing a solution. Most of the problems we study are intractable. In this case, either we further explore the space of the problem's instances to identify in which of them lies the computational hardness, or we consider a meaningful relaxation of the problem for which we seek an efficient algorithm.

This thesis extends the current state of the art on both computational complexity, and approximation algorithms of various problems in strategic settings. First, we deal with an evolutionary setting where we show that for a wide range of symmetric bimatrix games, deciding ESS existence is intractable. Then, we consider a setting where numerous entities compete repeatedly over a common resource. We present NEs and further categorize them in terms of desirable efficiency qualities. Next, we study a network security game. We characterize the NEs, study their complexity, and measure how effective they are in securing the network using the Price of Defense notion, analogous to the Price of Anarchy. After that, we consider an important fair division problem, namely the Consensus Halving problem. We bound the complexity of computing an exact solution and en route define a new complexity class which has interesting relations with already existing ones. Finally, we present a general framework for constructing approximation schemes for problems that can be written as an Existential Theory of the Reals formula with variables constrained in a bounded convex set. Using this framework, we provide new quasi-polynomial and polynomial time approximation schemes for optimisation problems, variations of problems on normal form games, Consensus Halving, and computational geometry.

Acknowledgements

I am extremely fortunate to have Paul Spirakis as my supervisor. If I had to keep only one of the uncountably many beautiful things I learned from him, it would be the passion for solving problems. Paul, this is a gift that I will never be able to thank you enough for. Also, your mentoring and support throughout my Ph.D. studies, in research and personal level, have not only helped me during difficult times, but most importantly, taught me how to be a good friend and mentor myself. I want to further thank you for introducing to me Computer Science, an event that deeply affected my life.

A huge thanks goes to my second supervisor Giorgos Christodoulou for being there whenever I needed his help in research and in other non-research related situations. I am thankful to my academic advisors, Piotr Krysta and Martin Gairing for assessing my progress and helping me improve my research skills. Also, I am grateful to the members of my thesis committee, Piotr Krysta and Elias Koutsoupias, for their feedback during the viva examination and for their constructive comments on the thesis. I would like to thank the people I had the privilege of collaborating with: Paul, Giorgos, Sotiris Nikoletseas, Christoforos Raptopoulos, John Fearnley, Argyrios Deligkas, and Eleni Akrida. Also, I want to thank Frans A. Oliehoek for our inspiring discussions.

I cannot omit thanking all the great people in the Economics and Computation group of the department, for inspiring me and creating such a nice atmosphere. Being part of this group was a truly stimulating experience. Further, I would like to thank the University of Liverpool for the financial support that made my Ph.D. studies possible and gave me the privilege to meet and work with all these great people.

My days in Liverpool would not have been the same without the friends and office-mates I was fortunate to meet during these years: Eleni, Argyris, Xenophon, Alkmini, Dimitris, Ifigenia, Pratibha, Peter, Matoula, Daniyah, Lefteris, Christos, Bart, Austin, Yiannis, Nicos, Eleni, Michalis, Manos, Elektra, George, Katerina, Katerina, Katerina, Katerina, Aris, Kostas, and all those that I might have forgotten. Thank you guys for all the laughs we had together. Last but not least, I want to thank my friends in Greece and, most importantly, my family: Ioannis, Anastasia, Dimitris and Flora. Without your love, understanding and support I would not be able to follow this path in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

How are choices made? This is a fundamental philosophical question which defines and differentiates theories about the interaction of living entities. A natural and very popular assumption in Biology and Economics is that an individual has a sense of what is in her best interest and tries to achieve it. But what will happen if in an environment of co-existence everyone acts in a selfish manner? Game Theory models mathematically settings where selfish entities interact, and tries to derive meaningful answers to that question. Given some assumptions about this system of interactions, its behaviour can seem chaotic since conflicting interests constantly change its state without reaching stability, or as we say, a *solution*.

Various solution concepts have been defined depending on the system of interaction. In an evolutionary setting, a population of non-rational and non-intelligent entities compete, and the solution concept is the Evolutionarily Stable Strategy (ESS); that is a mixture of actions of the population that cannot be overtaken by any other mixture that intrudes into it. In a setting with rational (i.e. seeking to maximize their own utility) and intelligent (i.e. fully aware of the environment and implications of their own actions) entities freely competing over some finite resource, the solution concept is the well-known Nash Equilibrium (NE). For another setting with entities that wish to feel equally treated when a finite resource is split among them, we desire a Consensus Halving solution.

Given a setting and a solution concept, the natural questions that arises are: *Does a solution exist for every instance?* If not, *how difficult is it to decide existence?*, and if yes, *how difficult is to compute a solution?* Both the latter questions translate to well defined problems in Computer Science and are studied using computational complexity theory. If

the computational problem in hand turns out to be intractable, then two typical courses of action are the following: (a) Explore the space of the problem's instances to pinpoint in which subset the computational hardness lies, or (b) make a meaningful relaxation of the problem which hopefully admits an *efficient algorithm*.

In this thesis we study various strategic settings and in each of them we are interested in extending the results regarding the aforementioned problems. In Part I we deal with evolutionary games. We are interested in exploring the instances for which deciding existence of an ESS is intractable. In Part II we study two games between rational and intelligent entities; one that models competition over time for access to a common resource, and another that models conflict of forces that try to harm and protect a network. In both games, we search for NEs and try to measure how efficient they are in terms of quality and computational complexity. In Part III we consider a setting where an object needs to be split in a way that seems fair to the interested individuals, and we study the complexity of computing such a solution. Finally, in Part IV we provide a general framework for constructing approximation schemes for a significantly wide range of computationally hard problems.

In the following sections of the Introduction we give a brief overview of the strategic settings and the results of each part of the thesis. The detailed model, related work and results for each setting are presented in the corresponding chapter.

1.1 Evolutionary Games

In Part I we study the behaviour of an unstructured population of entities that interact pairwise, by modelling it as a game. The motivation for this kind of games comes from questions in Biology such as *How do attributes of a species evolve over time?* Similar questions gave rise to an entire field of study known as Evolutionary Game Theory in the early 60's by Lewontin [94]. Evolutionary settings of structured populations have been studied towards understanding ESS [90] and also the spread of diseases [95,107,109,128]. For settings of unstructured populations whose solution concept is the ESS, models of both finite [126] and infinite [104] populations have been studied. The latter has attracted significantly more attention since it has been shown in [126] to approximate really well the finite case, and it is significantly easier to analyse. In this thesis only the infinite case is considered.

In Evolutionary Game Theory a game, as defined in the seminal work of Smith and Price

[104], involves animal species and not only humans, therefore it does not require individual rationality or intelligence from the involved entities, contrary to a classical game. Surprisingly enough, this interaction between non-rational and non-intelligent entities reduces to a two-player symmetric game under the classical definition, whose only players are two copies of Nature that try to reach a symmetric equilibrium. But how does individual rationality and intelligence emerge out of nothing in this model? The answer is that since in real life individuals that fit better than others in their environment tend to have more chances of passing on their genes, nature "picks" them as strategies to be played in this game of life in a way that seems "intelligent" but it is not. The same mechanism makes Natural Selection seem an intelligent process even though there is no requirement for intelligence.

A starting point to visualize an evolutionary game of an unstructured population is to be thought of as an infinitely repeated, multi-player, normal-form game of non-rational players. This game models the interaction between the members of a particular species, and not interspecies interaction. Every player plays sequentially against every other player and gets a payoff prescribed by some payoff matrix A that quantifies her attribute's fitness against any other attribute. Since the population is infinite, the game is repeated for infinitely many rounds. Each player has a single action which is an attribute (e.g. shape, size, behaviour, etc.) hardwired in her genome and her expected payoff is the average of payoffs she received over the infinite games played. A reasonable assumption is that the possible attributes n of the population are finite, thus there are fractions $s_i \geq 0$, $i \in \{1, 2, \ldots, n\}$ of individuals of the same type i, where $\sum_i s_i = 1$, and also matrix A is finite $n \times n$. The whole population's payoff is then the quantity $s^T As$, where $s = (s_1, s_2, \ldots, s_n)$.

The current state of the species can be captured by the aforementioned s which is a probability distribution on the action set and is also called a *(mixed) strategy*. The most popular solution concept in such evolutionary games is the *Evolutionarily Stable Strategy (ESS)*, introduced in [104], which captures a notion of stability in biological systems. Suppose that in the current population with strategy s another small population with strategy t is inserted, resulting in a mix of populations. If the expected payoff of the population playing strategy s is strictly greater than the expected payoff of the population playing any strategy $t \neq s$, when t is played by an arbitrarily small fraction of the total population, then s is an ESS. In other words, an ESS is a strategy (a mixture of genes in a population) that cannot be invaded by a small group of any other strategy.

Via a simple algebraic manipulation it can be shown that the constraints for the existence of an ESS in the aforementioned infinitely repeated game in the infinite population can

be reduced to the problem of finding a symmetric Nash equilibrium (s, s) in the one-shot, symmetric, two-player game defined by payoff matrix A with the following extra property: for any $t \neq s$, if $t^T As = s^T As$ then $s^T At > t^T At$. The latter shows that the ESS notion is a refinement of a strategy in a symmetric Nash equilibrium in two-player symmetric games. That is, if an ESS is played by both players in a given symmetric game, then this is a Nash equilibrium. However, contrary to a symmetric Nash equilibrium [113], not all two-player symmetric games admit an ESS; an example of a game with no ESS is the well-known Rock-Paper-Scissors game. This fact naturally leads to questions in Computer Science, such as *How hard is it to decide whether a given game possesses an ESS?* In Chapter 3 we study this question and explore the computational complexity of the decision problem for various instances.

1.2 Games between Rational and Intelligent Entities

In Part II we study two games that involve rational and intelligent *players*. The first game proceeds in discrete, *unbounded* time and the players have to access a common resource as fast as they can using a given strategy but being free at the same time to deviate from it if there is a better one. The actions of each player in this game are infinite, therefore the NE existence theorem of Nash [113] does not apply here. We prove constructively the existence of a common equilibrium strategy for the players that is also time-efficient. The second is a finite, one-shot game, played on a network with two conflicting forces; the attackers and the defender of the network. We are interested in the form of the equilibria in this game, the complexity of finding one, and the efficiency of the equilibria from the point of view of the defender.

1.2.1 Strategic Contention Resolution

Consider the case where many selfish users try to access a multi-access channel in discrete time, and each user chooses a strategy (transmission protocol) that prescribes how to access the channel. If at some time-step more than one user tries to access it, then no user is successful and has to try again in the future. If a user at some time-step is successful, then she no longer participates in the competition. We would like to provide the users with a *protocol*, i.e. a prescription of actions for every time-step, that is fair, time-efficient, and stable. The main question that is raised in this setting is *Is there a combination of protocols*

for the users which is fair, time-efficient, and stable?

For example, putting the users in a priority queue and prescribing to them to access the channel one after the other is optimal in terms of time-efficiency, but it seems unfair from the perspective of the users last in the queue, since they do not trust the protocol provider that is not corrupt. Even if the queue is created using randomization, again the same lack of trust would be justified since the user has no proof of the way her priority number is produced, therefore she might think that her expected time until success is greater than that of the other players. As another example consider the case of $n \geq 3$ users and a protocol common to all users that prescribes to each one to attempt access at every time-step, which results to infinite time until success for everyone. This protocol satisfies the fairness and stability properties since every user is given the same protocol, and no one can unilaterally deviate from it and lower her expected time until success, but it is clearly not time-efficient.

In Chapter 4 we extend the aforementioned single channel setting to one with $n \geq 1$ users and $k \geq 1$ channels. We model it as an infinitely repeated game with n players each of whose individual cost is the expected time until success. The protocols we seek are *anonymous*, in order to guarantee fairness. An anonymous protocol does not depend on the identity of the user, and also is run locally by her. We wish also to find *efficient* protocols, meaning that the time until everyone is successful is $\Theta(n/k)$ with probability tending to 1 as $n/k \to \infty$. The last property we require for our protocols is to be *equilibrium* protocols. The equilibrium notion here is the Nash equilibrium [113] and means that no user can unilaterally deviate from the provided protocol and strictly lower her own expected time until success.

Our extension of the model to multiple channels is motivated by the trend of the last roughly sixteen years in the Electrical and Electronics Engineering community to design multiple-channel medium access protocols (MAC). This deviation from the single channel MAC protocol aimed in higher throughput and robustness against failure of a channel. Among our results there are protocols that guarantee these properties even in the setting with strategic/selfish users.

1.2.2 Connected Subgraph Defense Games

In Chapter 5 we turn to a game between individuals whose actions depend on a graph structure. We consider a game on a network with $k+1$ players: a *defender* and $k \geq 1$

attackers. Each attacker can choose any node of the network to "infect". The defender can choose and "clean" any connected induced subgraph of the network with λ nodes. As usual, the players are able to choose a probability distribution over their choices. The attackers try to maximize the infection success, i.e. the expected number of infected nodes, by avoiding the defender, while the defender tries to minimize the infection success by catching as many attackers as possible in expectation.

This game models the situation where opposite forces compete over a computer network. There are many harmful softwares independently attacking a network of n nodes with limited resources $k \leq n$, since maybe the network is vast. A security software is used from the side of the network to eliminate the threats by putting a "seed" in a computer and from there spreading it to λ computers in total. The limitation on the size $\lambda \leq n$ again may be due to the fact that the network is vast and the security software only manages to cover λ nodes before a new set of attackers appears, or it could be because of budget limitation that does not allow purchase of a global security software.

The main questions that arise in such a setting are: *Given available resources λ for the defender but no knowledge of the graph, what is the worst fraction of the attackers she can catch if she uses her resources optimally?*, and *Given available resources λ for the defender, what are the graphs that maximize the fraction of the attackers she can catch if she uses her resources optimally?* In order to analyse resource optimality in this strategic environment one has to use tools from Game Theory and constrain the solutions to those that constitute a Nash equilibrium; no player regrets having chosen her (mixed) strategy that attacks or defends.

1.3 Fair Division

In Part III we turn from game-theoretic settings to a fair division setting. Fair division problems such as "cake cutting" [17,18,31], "rent division" [65,79] and "Consensus Halving" [133] involve selfish agents who want to maximize their own utility. However, contrary to game-theoretic settings, the agents are not able to deviate on their own. They are given a way for dividing an object, and they have to obey it. Even though the freedom of the agents in this setting is removed, the hardness in such problems lies in the fairness notion that has to be satisfied. In many fair division problems the existence of a solution is guaranteed via a theorem from algebraic topology such as Brouwer's fixed point theorem, Sperner's lemma, or Kakutani's fixed point theorem. Therefore, a most interesting question that is

raised regarding computational complexity is *How hard is it to compute a solution that divides fairly the object?*

We deal with the Consensus Halving problem which can be described via the following example: Suppose there is a land owned $50-50$ by the two founding members of a real-estate company. The company consists of departments each of which has different customers who desire that land. Therefore each department has set a (in general) different price function on the land, where the price function determines the price for any area of the land. Now suppose that the company wants to split into two companies, and wants to do it in a fair manner, so that the equally co-owned land will produce for both parts the same profit after it is sold. How can the land be split in two parts such that, no matter how the departments are allocated in the new companies, the new companies will have the same profit?

More formally, in the Consensus Halving problem an object A represented by $[0, 1]$ is to be divided into two halves A_+ and A_-, so that n agents agree that A_+ and A_- have the same value, given that each agent has her personal valuation function over A. Provided that every valuation function of the agents is bounded and continuous over A, this can always be achieved using at most n *cuts*, and this fact can be proved via the Borsuk-Ulam theorem from algebraic topology [133]. The computational problem we study is to compute a solution to the Consensus Halving problem.

1.4 Approximation Algorithms

In Part IV we study approximation algorithms for various computationally hard problems. We develop a general framework for constructing approximation schemes for any problem that can be written down in an ETR formula or even an ETR augmented with expressions beyond its grammar. ETR stands for the Existential Theory of the Reals which consists of sentences containing only existentially quantified formulae using the connectives $\{\wedge, \vee, \neg\}$ over polynomials compared with the operators $\{<, \leq, =, \geq, >\}$. For example, each of the following is a formula in ETR.

$$\exists x \exists y \exists z \cdot (x^4 \cdot y \cdot z^2 = 2) \qquad \exists x \exists y \cdot ((x^2 = y) \wedge (x > y)) \vee (x < y^3)$$

The class `ETR` consists of all "yes" instances of ETR sentences, and equivalently contains every problem that can be reduced in polynomial time to a formula in the existential theory of the reals (ETR formula). The typical problem in `ETR` is to decide whether the formula

is *true*, that is, whether there exist real values for the variables that satisfy the formula. It is easy to see that the Boolean satisfiability problem can be formulated as an ETR problem, therefore NP $\subseteq$ ETR. Also, due to Canny [35] it is known that ETR $\subseteq$ PSPACE and ETR is suspected to be closer to PSPACE than to NP.

Many natural problems have been shown to be ETR-complete, mainly in Computational Geometry (for a great survey, see [38]), but also in Game Theory regarding constrained Nash equilibria [25–27,75,125]. In the side of approximation algorithms, Lipton, Markakis and Mehta [96] derive an algorithm (LMM) which is a quasi-polynomial time approximation scheme (QPTAS) for the problem of finding an ϵ-NE (with no additional constraints). The LMM algorithm is based on proving that if a solution exists to a conjunction of multilinear inequalities, then there is a proper discrete solution to the conjunction of the ϵ-relaxed multilinear inequalities. Since an exact solution to the NE problem can be expressed as such a conjunction, and because there is always a solution to it, there is always a proper discrete solution to the conjunction of relaxed inequalities. Since the latter conjunction corresponds to a solution to ϵ-NE, it suffices to find a solution to that conjunction. The properness of the discrete solution means that it is so simple that can be found in quasi-polynomial time for constant ϵ.

The aforementioned results regarding NE problems raise the following question: *Is there a broader class of problems to which the sampling technique can be applied?* We answer this in the positive by providing a sampling theorem for ETR in the case where the domain of the variables can be described by a convex set.

1.5 Thesis Contribution

1.5.1 Evolutionary games

In Chapter 3 we are interested in exploring to what extend the problem of deciding existence of an ESS is hard. Conitzer [49] showed that this problem is Σ_2^P-complete, while by then it had been shown that the problem is NP-hard and coNP-hard in [67], and coDP-hard in [115]. On the other hand, [127] showed that when the payoffs of the game are drawn from some common distributions, then an ESS of support size 2 exists with high probability. So the natural question that arises is whether the hard games constructed in [67] become easy by perturbing (randomly or deterministically) its payoff values. In [108] we answer this question in the negative. By extending the reduction of [67] we show that the problem

remains coNP-hard even for arbitrary deterministic perturbations of these values.

En route, we introduce the notion of a robust reduction, for reductions from a problem to another one that involves a real matrix. If it is shown that a reduction is robust, then the reduction holds for a wide range of payoff values of the resulting problem. Starting with the reduction of [67] that shows coNP-harness for the existence of ESS, we construct a robust reduction and thus prove that there is a large family of symmetric bimatrix games for which existence of ESS remains coNP-hard to decide. The robustness notion might be of independent interest and we believe it could prove itself a useful tool towards shedding light on the instances of game-theoretic problems that are hard for some class.

1.5.2 Games between rational and intelligent entities

1.5.2.1 Strategic contention resolution

In Chapter 4 we try to answer if there exists an anonymous, efficient, equilibrium protocol for given numbers $n \geq 1$ and $k \geq 2$ of players and channels respectively. The difficulty in the analysis and also the results themselves regarding the above question depend heavily on the information that the players get back from the channel throughout the game. That is because the techniques for optimizing the expected time until success for each player, depending on what information she has, can be achieved by reducing to a Markov Chain, a Markov Decision Process (MDP), or a Partially Observable Markov Decision Process (POMDP). The solution of each of these three models requires increasing computational difficulty, with the latter having no method for solution in the current literature. We study this question in two feedback settings whose analyses correspond to POMDPs and MDPs respectively: the *acknowledgement-based feedback* and the *ternary feedback*.

In the acknowledgement-based feedback the player gets just the information of whether she had a successful attempt or not, only when she attempts to have access. In the ternary feedback the user is informed about the number of pending players in each time-step regardless of whether she attempted transmission or not. The only theoretical results on these feedback settings are for a single transmission channel by Christodoulou et al. [43] and by Fiat et al. [69].

While for the single-channel case a ternary feedback protocol with the required properties for $n \geq 1$ players has been found in [69], in the acknowledgement-based feedback even for $n = 3$ there is no result on the existence of an equilibrium protocol. As it turns out, in the single-channel case with acknowledgement-based feedback, the analysis for finding

equilibria (if any) with finite expected time until success per player becomes extremely hard for $n = 3$. This is the reason why in [44] the study is restricted to finding an anonymous protocol that prevents some pathological behaviour, but is not necessarily an equilibrium protocol. If there are no better characterizations of equilibria than the ones we provide in Chapter 4, then there are no known techniques to find an equilibrium for more than two players in the single-channel setting.

Since we wish to have equilibrium protocols for more than two players, we increase the number of channels in hope of making the proof of equilibrium existence possible and the search for equilibria easier. In Chapter 4 for the acknowledgement-based feedback and $k \in \{2, 3\}$ channels we present simple, anonymous equilibrium protocols for specific fixed number of players. For the ternary feedback and $k = 2$ channels we extend the result of [69] on history-independent protocols by finding the unique anonymous, equilibrium protocol which is also not efficient though.

Finally, for any $k \geq 1$ and $n \geq 2k + 1$ we present an anonymous, efficient, equilibrium protocol for both feedback classes. These results extend those of [43] in the acknowledgement-based feedback that considered only $k = 1$ and provide a new general protocol for the ternary feedback. Our results indicate that there is a trade-off between efficiency and the property of having finite expected time until success for a player. Proving the existence of a protocol with the latter property for $k \geq 2$ remains an open problem.

1.5.2.2 Connected subgraph defense games

In Chapter 5 we extend the line of work of Mavronicolas et al. in their seminal paper [100] on defense games in graphs. We term the type of games we consider *Connected Subgraph Defense (CSD) games*. In [100] the defender had the power to defend only two adjacent nodes of the network, i.e. that work considered only the special case $\lambda = 2$. In our model we have the available resources $\lambda \in \{1, \ldots, n\}$ of the defender as a parameter and study the behaviour of equilibria depending on that parameter.

We study many questions related to CSD games, all of which regard equilibria. As a first important step, we precisely characterize the Nash equilibria and defense-optimal graphs; that is graphs that allow the best defense over all graphs. We provide an LP-based algorithm that computes an exact equilibrium of any given CSD game, whose running time is polynomial in $\binom{n}{\lambda}$.

We then study tree-graphs, and show a special characterization of defense-optimality

in such graphs. This characterization is strong enough to provide us with the arguments to prove a polynomial-time algorithm that decides whether a tree is defense-optimal, and if it is, to output a defense-optimal Nash equilibrium. On the other hand, we prove that it is NP-hard to find an optimal strategy if the tree is not defense-optimal.

Following the terminology of [100], we also extend the notion of *Price of Defense* for any λ. The Price of Defense is defined as the minimum ratio of the total number of attackers over the expected number of attackers that the defender catches in equilibrium over all graphs. This notion is a measure of how bad the equilibria of the CSD game are over all graphs. Through approximation algorithms that induce upper bounds and graph constructions that yield lower bounds we conclude that the PoD is roughly $2n/\lambda$ meaning that in the worst equilibrium case over all graphs the expected number of attackers caught by the defender is $k\lambda/2n$, where k is the total number of attackers.

1.5.3 Fair division

In Chapter 6 we are interested in pinning down the complexity of computing an exact solution to the Consensus Halving problem. Recent work has shown that the approximate version of this problem is PPA-complete [71,72]. Here we show that the exact version is much harder (under standard complexity assumptions). In particular, finding a solution with n agents and n cuts is FIXP-hard, and deciding whether there exists a solution with fewer than n cuts is ETR-complete. ETR stands for the "Existential Theory of the Reals" class, which is between NP and PSPACE and suspected to be closer to PSPACE.

In order to capture the precise complexity of the problem we define a new complexity class, named BU, which captures all problems that can be reduced to solving an instance of the typical problem of BU, namely Borsuk-Ulam, exactly. The Borsuk-Ulam problem is a search problem that asks for antipodal points on the surface of an $(n+1)$-dimensional sphere, that are mapped to the same point on $\mathbb{R}^n$ via a function represented by polynomial size algebraic circuits over basis $\{+, *, -, \max, \min\}$ with rational constants. The existence of such antipodal points is guaranteed by the Borsuk-Ulam theorem, a well known result in algebraic topology [133].

Furthermore, we show that FIXP $\subseteq$ BU $\subseteq$ TFETR and that LinearBU = PPA, where LinearBU is the subclass of BU in which the Borsuk-Ulam instance is specified by a linear arithmetic circuit. The latter is analogous to the result that LinearFIXP = PPAD by Etessami and Yannakakis in [66] establishing a surprising relation between the class FIXP

of exact solutions via Brouwer's fixed point theorem and the class PPAD of solutions proven via the parity argument on directed graphs. Our results indicate that BU is indeed a class on its own, which has more encoding power than FIXP. Such a fact in the computational world would agree with the mathematical fact that the Borsuk-Ulam theorem implies the fixed-point theorem of Brouwer [140], the theorem that guarantees the existence of fixed-points in the typical problem of FIXP.

1.5.4 Approximation algorithms

Finally, in Chapter 7 we provide efficient algorithms that output approximate solutions to computationally hard problems. We first show that the general ETR problem whose variables are not constrained in a bounded domain, even if the formula is relaxed by ϵ, is ETR-complete. Since the completeness proof heavily relies on the fact that the variables are not constrained, we consider the general ETR problem whose variables are constrained in a bounded convex set, namely "constrained ϵ-ETR". The main result of Part IV is an extension of the sampling technique of Lipton-Markakis-Mehta (LMM) [96] which applies to constrained ϵ-ETR. This yields an algorithm that, depending on the parameters of the problem under examination, can be a PTAS or a QPTAS.

In particular, we initiate our set of results by proving a sampling LMM-like theorem for the existence of ϵ-close-to-optimal solutions under some objective of a degree-d polynomial function: the degree of every of this function's term is exactly $d \geq 1$. The proof of LMM has a bottleneck at $d = 1$, meaning that it works only for multilinear polynomial functions, but not for higher than degree-1 functions. Extending this proof is the crucial result which opens the door for consecutive results that finally lead to our main theorem. These consecutive results, one by one add a parameter to the solution that affects its quality; and the quality, in turn, affects the running time of the proposed algorithm that finds the solution. We make sure in each step that the solution quality remains in desired bounds, so that the solution of our final theorem, parameterized by all the required parameters, has quality high enough to yield an algorithm as efficient as possible.

There is a plethora of search problems that can be captured by an ETR formula. Hopefully, when this formula is relaxed, a solution to it corresponds to the approximate version of the initial problem. In Part IV we present problems from a variety of fields as examples of what our method can do: unconstrained or constrained approximate Nash equilibrium, Shapley games, approximate Consensus Halving with polynomial valuation functions, op-

timization problems with polynomial functions, tensor problems, and several problems in computational geometry. We use our result to create several new PTAS and QPTAS algorithms for the aforementioned problems.

1.6 Author's Publications Presented in this Thesis

Chapter	***Title***	***Authors***	***Proceedings***
3	Existence of Evolutionarily Stable Strategies Remains Hard to Decide for a Wide Range of Payoff Values [108]	T. Melissourgos, P.G. Spirakis	CIAC 2017[1]
4	Strategic Contention Resolution [47] in Multiple Channels	G. Christodoulou, T. Melissourgos, P.G. Spirakis	WAOA 2018[2]
	Short Paper: Strategic Contention Resolution in Multiple Channels with Limited Feedback [46]		SAGT 2018[3]
5	Connected Subgraph Defense Games [4]	E.C. Akrida, A. Deligkas, T. Melissourgos, P.G. Spirakis	SAGT 2019[4]
6	Computing Exact Solutions of Consensus Halving and the Borsuk-Ulam Theorem [58]	A. Deligkas, J. Fearnley, T. Melissourgos, P.G. Spirakis	ICALP 2019[5]
7	Approximating the Existential Theory of the Reals [57]	A. Deligkas, J. Fearnley, T. Melissourgos, P.G. Spirakis	WINE 2018[6]

Table 1.1: List of author's publications presented in this thesis

[1]CIAC 2017: 10th International Conference on Algorithms and Complexity.
[2]WAOA 2018: 16th Workshop on Approximation and Online Algorithms.
[3]SAGT 2018: 11th International Symposium on Algorithmic Game Theory.
[4]SAGT 2019: 12th International Symposium on Algorithmic Game Theory.
[5]ICALP 2019: 46th International Colloquium on Automata, Languages and Programming.
[6]WINE 2018: 14th International Conference on Web and Internet Economics.

Chapter 2

Preliminaries

In this chapter we set the notation that will be typically used throughout the thesis. Unless it is stated otherwise in the respective chapter, the notation established here is followed. We also provide some basic definitions of Algorithmic Game Theory and Graph Theory together with some preliminary results from bibliography.

2.1 Sets

Given a set C with cardinality $c := |C|$, if it is $C = \{1, 2, \ldots, c\}$, we denote the set also by $[c]$. Conversely, if we denote a set by $[c]$, we imply that $[c] = \{1, 2, \ldots, c\}$. $\mathbb{R}^n_{\geq 0}$ denotes the set of non-negative real number vectors $(x_1, x_2, .., x_n)$. We denote the standard $(n-1)$-simplex by $\Delta_n := \left\{(x_1, x_2, .., x_n) \in \mathbb{R}^n_{\geq 0} : \sum_{i=1}^n x_i = 1\right\}$.

2.2 Bimatrix Games

A *bimatrix game* is a two-player strategic form game $\Gamma = (S_1, S_2, u_1, u_2)$ defined by two finite sets S_1, S_2 of *pure strategies* (also called *actions*) and utility (or *payoff*) functions $u_1 : S_1 \times S_2 \mapsto \mathbb{R}$ and $u_2 : S_1 \times S_2 \mapsto \mathbb{R}$ for the row-player and the column-player, respectively. These payoff functions define *payoff matrix* $A_\Gamma = (a_{i,j})$ and $B_\Gamma = (b_{i,j})$ for players 1 and 2 respectively, where $a_{i,j} = u_1(i, j)$ and $b_{i,j} = u_2(i, j)$ for $i \in S_1$, $j \in S_2$. For simplicity, assume $S_1 = [n]$ and $S_2 = [m]$, i.e., pure strategies are identified with integers $i \in [n]$ and $j \in [m]$ for each player respectively.

A bimatrix game is called *symmetric* if $S_1 = S_2 =: S$ and $u_1(i, j) = u_2(j, i)$ for all

$i, j \in S$. If we are only concerned with symmetric two-player strategic form games, we write (S, u_1) as shorthand for (S, S, u_1, u_2), since $u_2(j, i) = u_1(i, j)$ for all $i, j \in S$. In a symmetric bimatrix game, the column-player's payoff matrix is the transpose of the row-player's payoff matrix, i.e. $B_\Gamma = A_\Gamma^T$. Note that A_Γ is not necessarily symmetric, even if Γ is a symmetric game.

A *(mixed) strategy* $s = (s(1), ..., s(n))^T$ over some set S (with $|S| = n$) of pure strategies is a vector that defines a probability distribution on S and we will denote by $s(i)$ the probability assigned by strategy s on the pure strategy $i \in S$. Thus, $s \in \Delta_n$. Strategy s is called *pure* if and only if $s(i) = 1$ for some $i \in S$. In that case we identify s with i. For brevity, we generally use the term *strategy* to refer to a mixed strategy s, and indicate otherwise when the strategy is pure.

Consider the strategies $s \in \Delta_n$, $t \in \Delta_m$. The *expected payoff* function $U_k : \Delta_n \times \Delta_m \mapsto \mathbb{R}$ for player $k \in 1, 2$ is given by $U_k(s, t) = \sum_{i,j \in S} s(i)t(j)u_k(i, j)$, for all strategies $s \in \Delta_n$, $t \in \Delta_m$. Note that when the game is symmetric, then $U_1(s, t) = s^T A_\Gamma t$ and $U_2(s, t) = s^T A_\Gamma^T t = t^T A_\Gamma s$.

For player $k \in \{1, 2\}$, strategy $t \in \Delta_n$ is a *best response* to s if $U_k(t, s) = \max_{t' \in \Delta_n} U_k(t', s)$. The *support* $\text{supp}(s)$ of s is the set $\{i \in S : s(i) > 0\}$ of pure strategies which are assigned non-zero probability in s. A pair of strategies (s, t) is a *Nash equilibrium* (NE) for game Γ if s is a best response to t and t is a best response to s. A Nash equilibrium is guaranteed to exist in every finite bimatrix game [113]. Its formal definition is the following.

Definition 1 (Nash equilibrium)**.** *A strategy profile (s, t) is a Nash equilibrium for the bimatrix game $\Gamma = (S_1, S_2, u_1, u_2)$ if $s^T A_\Gamma t \geq s'^T A_\Gamma t$ for every $s' \in \Delta_n$ and $s^T B_\Gamma t \geq s^T B_\Gamma t'$ for every $t' \in \Delta_m$.*

In a finite symmetric bimatrix game a symmetric Nash equilibrium always exists [113]. Its formal definition is the following.

Definition 2 (Symmetric Nash equilibrium)**.** *A strategy profile (s, s) is a symmetric Nash equilibrium for the symmetric bimatrix game $\Gamma = (S, u_1)$ if $s^T A_\Gamma s \geq s'^T A_\Gamma s$ for every $s' \in \Delta_n$.*

2.3 Graph Theory

An *undirected graph* G is an ordered pair (V, E) consisting of a set V of *vertices* and a set E of *edges*, which consists of unordered pairs of elements of V. We usually denote $n := |V|$.

Definition 3 (Adjacency matrix)**.** *The adjacency matrix of the above undirected graph G is the $n \times n$ matrix $A_G := (a_{uv})$, where a_{uv} is 1 if $(uv) \in E$, and 0 otherwise.*

Definition 4 (Clique)**.** *A clique of an undirected graph G is a complete subgraph of G, i.e. one whose vertices are joined with each other by edges.*

The following is a very important theorem by Motzkin and Straus that establishes an interesting relation between the maximum-size clique in a graph and quadratic optimization over the simplex.

Theorem 1 ([111])**.** *Let $G = (V, E)$ be an undirected graph with maximum clique size d. Then $\max_{x \in \Delta_n} x^T A_G x = \frac{d-1}{d}$.*

Etessami and Lochbihler used the above theorem to prove the following, more general result.

Corollary 1 ([67])**.** *Let $G = (V, E)$ be an undirected graph with maximum clique size d and let $\ell \in \mathbb{R}_{\geq 0}$. Let $\Lambda_\ell = \left\{ x \in \mathbb{R}^n_{\geq 0} : \sum_{i=1}^n x_i = \ell \right\}$. Then $\max_{x \in \Lambda_\ell} x^T A_G x = \frac{d-1}{d} \ell^2$.*

2.4 Complexity Classes

In the standard Turing machine model, a *(computational) problem* is defined as a function f that needs to be computed by a Turing Machine (TM). The *input* of a problem is the arguments of function f, and the result of f is the problem's *output*. An *algorithm* for computing f is a TM which encodes a set of finite instructions whose execution terminates in finite time. The input to an algorithm is a particular *instance* of the problem and the output is the computed function f. Given a problem and an algorithm for it, the *(worst-case running) time* of the algorithm is the maximum time over all instances of the problem needed for the computation of f. When f is Boolean, meaning that its output can only be a single bit representing the answers "yes" or "no", the problem is called *decision problem*. If f can output a computed value greater than a single bit, the problem is called *function*

problem (or *search problem*). A *language* corresponding to a decision problem P is the set containing all bit strings encoding an input of P for which the answer is "yes".

A *complexity class* is a set of problems (functions) whose computation can be done by some algorithm within specific running time. Equivalently, a complexity class of decision problems can be defined as the set of languages corresponding to the decision problems of the class. In this section we describe the complexity classes that are mentioned in this thesis.

P. Contains all decision problems for which there are TMs that decide them in time polynomial in the input size.

NP. Contains all decision problems for which, any "yes" instance can be verified by a TM in time polynomial in the input size using a certificate of size polynomial in the input size. Equivalently, this class contains all decision problems that can be decided in time polynomial in the input size by a non-deterministic TM.

coNP. Contains all decision problems for which, any "no" instance can be verified by a TM in time polynomial in the input size using a disqualification of size polynomial to the input size.

DP. Defined in [118], consists of all languages L where $L = L_1 \cap L_2$ and $L_1 \in \texttt{NP}$ and $L_2 \in \texttt{coNP}$. DP is a syntactic class, not to be confused with $\texttt{NP} \cap \texttt{coNP}$.

coDP. Consists of all the complement languages $\bar{L}$ of $L \in \texttt{DP}$, where $\bar{L} = \bar{L_1} \cup \bar{L_2}$ and $\bar{L_1} \in \texttt{coNP}$ and $\bar{L_2} \in \texttt{NP}$. Clearly, $\texttt{NP} \subseteq \texttt{coDP}$, $\texttt{coNP} \subseteq \texttt{coDP}$ and $\texttt{coDP} \subseteq \Sigma_2^P$.

FNP. Called Function NP, this class contains all *binary relations* $R(x, y)$ for which the following hold:

- bit string y is at most polynomially larger than bit string x, and
- given x and y, there is a TM that decides whether $R(x, y)$ is a "yes" instance in time polynomial in the input size.

Informally, FNP contains all function problems whose corresponding decision problems are in NP. When the answer to the corresponding decision problem is "yes", then we require value y to be the output of the problem in FNP.

TFNP. Called Total Function NP, this class was defined in [105], and it is the subclass of FNP that contains all *binary relations* $R(x, y)$ of FNP for which, additionally, there is at least one y for every x that makes $R(x, y)$ a "yes" instance. Informally, TFNP contains all function problems of FNP which additionally are total, i.e. any instance of their corresponding decision version is a "yes" instance. Subsets of TFNP have been defined in an attempt to capture the exact complexity of each individual problem based on the attribute that makes them total. PPAD and PPA (see below) are such classes that contain problems whose solution is guaranteed by specific theorems on graphs.

PPAD. Named after the Polynomial Parity Argument in Directed graphs, this class is a subset of TFNP, and contains all problems whose underlying proof of totality is via the argument indicated by the class' name. The argument is that since the sum of all degrees of a directed graph is even, given an odd-degree vertex, another odd-degree vertex exists. The typical problem of the class, namely End-of-a-Line, gives an odd-degree vertex and asks to find another one. If the graph is given explicitely, e.g. by an adjacency matrix, then an easy polynomial-time algorithm can find the required odd-degree vertex. However, in this class, what is given as input is two *circuits* S and P that return for any vertex, its successor and predecessor in the graph. Then the size of the graph can be exponential in the input size of the problem.

Formally, the class contains all polynomial-time reducible problems to the End-of-a-Line problem:

Definition 5 ([50])**.** *Given two circuits S and P, with n input bits and n output bits each, such that $P(0^n) = 0^n \neq S(0^n)$, find an input $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.*

PPA. Named after the Polynomial Parity Argument in undirected graphs, this class is also a subset of TFNP, and contains all problems whose underlying proof of totality is via the same argument as the one of PPAD, but for undirected graphs. Clearly, it is PPAD $\subseteq$ PPA $\subseteq$ TFNP $\subseteq$ FNP.

ETR. Defined in [125], it is also denoted $\exists\mathbb{R}$. This class is named after the Existential Theory of the Reals (ETR), which is the set of true sentences of the form

$$(\exists x_1, \ldots, x_n)\phi(x_1, \ldots, x_n),$$

where ϕ is a quantifier-free $(\vee, \wedge, \neg)$-Boolean formula over the signature $(0, 1, +, *, <, \leq, =)$, and the sentence is interpreted over the universe of real numbers. ETR is the corresponding class of ETR whose typical problem is to decide whether a system of multivariate polynomial (with rational coefficients) equalities/inequalities over the reals is satisfiable. It is NP $\subseteq$ ETR $\subseteq$ PSPACE due to [35] and the fact that the NP-complete problem SATISFIABILITY can be written in ETR form. ETR can be thought of as the analogue of NP in the Blum-Shub-Smale model of computation [28].

FETR, TFETR. We define FETR and TFETR in a way analogous to that of FNP and TFNP. In FETR are all function problems whose corresponding decision problem is in ETR. When the answer to the corresponding decision problem is "yes" for some input x, then in the function problem we require as output the computed function y. In TFETR are all function problems of FETR, which additionally are total, meaning that any instance of their corresponding decision version is a "yes" instance. Just like TFNP, we believe that TFETR contains subclasses which are characterized by the theorem of existence that guarantees a solution in the problems of each class. Such classes are FIXP (see below) which seems to be an analogue of PPAD, and a new class we define in Chapter 6, called BU, which seems to be an analogue of PPA, but so far lacks a complete problem.

FIXP. Defined in [66], this class captures search problems that can be cast as Fixed Point computation problems for functions represented by polynomial size algebraic circuits (straight line programs) over basis $\{+, *, -, /, \max, \min\}$ with rational constants. The typical problem of FIXP is, given a function F as described above whose domain is compact convex and mapped to itself, to compute a fixed-point, i.e. a point for which $F(x) = x$. The existence of such a point is guaranteed by Brouwer's fixed-point theorem [34]. It has been shown in [66] that the linear subclass of FIXP, called LinearFIXP, in which the $*$ of the basis is replaced by multiplication with a rational constant (and therefore the input circuit is linear), coincides with PPAD.

For a more detailed presentation of the landscape of complexity classes, the reader is referred to the excellent books [13,119]

Part I

Evolutionary Games

Chapter 3

Evolutionarily Stable Strategies in Infinite Populations

The concept of an *evolutionarily stable strategy* (ESS) is a refinement of Nash equilibrium in 2-player symmetric games introduced by Smith and Price [104], in order to explain behaviours of living entities to which nature has converged through evolution. The existence of ESSs in a finite 2-player symmetric game is not guaranteed, a fact that gives rise to the computational problem of deciding whether a game possesses an ESS. The problem has been shown to be Σ_2^P-complete by Conitzer [49], following the preceding important works by Nisan [115] and by Etessami and Lochbihler [67]. The latter, among other results, proved that deciding the existence of an ESS is both NP-hard and coNP-hard. In this work we introduce a *reduction robustness* notion and we show that deciding the existence of an ESS remains coNP-hard for a wide range of games even if we arbitrarily perturb within some intervals the payoff values of the game under consideration.

On the other hand, Hart and Rinott [127] showed that when the payoffs of the game are drawn from some common distributions, then an ESS of support size 2 exists with high probability. So the natural question that arises is whether the hard games constructed in [67] become easy by perturbing (randomly or deterministically) its payoff values. We answer this question in the negative. By generalizing the reduction of the latter work we show that the problem remains coNP-hard even for arbitrary deterministic perturbations of these values.

The results of this chapter have been published in the Proceedings of the 10th International Conference on Algorithms and Complexity (CIAC 2017) [108] (co-authored with

Spirakis).

3.1 Overview

3.1.1 Concepts of evolutionary games and stable strategies

Evolutionary game theory has proven itself to be invaluable when it comes to analysing complex natural phenomena. A first attempt to apply game theoretic tools to evolution was made by Lewontin [94] who saw the evolution of genetic mechanisms as a game played between a species and nature. He argued that a species would adopt the "maximin" strategy, i.e. the strategy which gives it the best chance of survival if nature does its worst. Subsequently, his ideas where improved by the seminal work of Smith and Price in [104] and Smith in [134] where the study of natural selection's processes through game theory was triggered. They proposed a game-theoretic model in order to decide the outcome of groups consisting of living individuals, conflicting in a specific environment.

The key insight of evolutionary game theory is that a set of behaviours depends on the interaction among multiple individuals in a population, and the prosperity of any one of these individuals depends on that interaction of its own behaviour with that of the others. An *evolutionarily stable strategy (ESS)* is defined as follows: An infinite population consists of two types of infinite groups with the same set of pure strategies; the *incumbents*, that play the (mixed) strategy s and the *mutants*, that play the (mixed) strategy $t \neq s$. The ratio of mutants over the total population is ϵ. A pair of members of the total population is picked uniformly at random to play a finite symmetric bimatrix game Γ with payoff matrix A_Γ. Strategy s is an ESS if for every $t \neq s$ there exists a constant ratio ϵ_t of mutants over the total population, such that, if $\epsilon < \epsilon_t$ the expected payoff of an incumbent versus a mutant is strictly greater than the expected payoff of a mutant versus a mutant. For convenience, we say that "s is an ESS of the game Γ".

The concept of ESS tries to capture the resistance of a population against invaders. This concept has been studied in two main categories: infinite, and finite population groups. The former was the one where this Nash equilibrium refinement was first defined and presented by Smith and Price [104]. The latter was studied by Schaffer [126] who showed that the finite population case is a generalization of the infinite population case. The current work deals with the infinite population case which can be mathematically modelled in an easier way, and in addition, its results may provide useful insight for the finite population case.

3.1.1.1 A useful example

In order for the reader to conceive the notion of the evolutionarily stable strategy, we give a simple example of the infinite population case. Let us consider a particular species of crab and suppose that each crab's fitness in a specific environment is mainly decided by its capability to find food and use the nutrients from the food in an efficient way. In our crab population a particular mutation makes its appearance, so the crabs born with the mutation grow a significantly larger body size. We can picture the population now, consisting of two distinct kinds of crabs; ϵ fraction of the population being the large ones and $1-\epsilon$ being the small ones. The large crabs, in fact, have difficulty maintaining the metabolic requirements of their larger body structure, meaning that they need to divert more nutrients from the food they eat and as a consequence, they experience a negative effect on fitness. However, the large crabs have an advantage when it comes to conflicting with the small ones, so they claim an above-average share of the food. To make our framework simple, we will assume that food competition involves pairs of crabs, drawn at random, interacting with each other once, but the reasoning of the analysis is equivalent to interactions that occur (simultaneously or not) between every possible pair, with each individual receiving the mean of the total fitness. When two crabs compete for food, we have the following "rules" that apply: (1) When crabs of the same body size compete, they get equal shares of the food. (2) When a large crab competes with a small crab, the large one gets the majority of the food. (3) In all cases, large crabs experience less of a fitness benefit from a given quantity of food, since some of it is diverted into maintaining their expensive metabolism. (4) When two large crabs compete, they experience even less of a fitness benefit since they put considerable effort in fighting. The following bimatrix encloses the aforementioned rules in the context of a game.

		Crab 2	
		Small	*Large*
Crab 1	*Small*	7 , 7	1 , 9
	Large	9 , 1	4 , 4

In this setting, we call a given strategy evolutionarily stable if, when the whole population is using this strategy, any small enough group of invaders using a different strategy will eventually die off over multiple generations. This idea is captured in terms of numerical payoffs by saying that, when the entire population is using a strategy s, then an arbitrarily

small ratio of invaders over the new (blended) population will have strictly lower fitness than the initial population has in the new population. Since fitness translates into reproductive success, and consequently passing one's genes on to future generations at higher frequencies, it is assumed by evolutionary principles [104] that strictly lower fitness is the reason for a subpopulation (like the users of strategy t) to shrink over time through multiple generations and eventually become extinct.

Let us see if any of the two pure strategies is evolutionarily stable. Suppose a population of small crabs gets invaded by a group of large ones (of ratio ϵ over the whole population). The expected payoff (fitness) of a small crab is:

$$7(1-\epsilon)+1\epsilon = 7-6\epsilon \qquad \text{because it meets a small crab with probability } 1-\epsilon \text{ and a large one with probability } \epsilon.$$

The expected payoff of a large crab is:

$$9(1-\epsilon)+4\epsilon = 9-5\epsilon \qquad \text{because it meets a small crab with probability } 1-\epsilon \text{ and a large one with probability } \epsilon.$$

Clearly, no ϵ can make the payoff of the small crabs greater than that of the large ones. So, the pure strategy *Small* is not an ESS. Now suppose a population of large crabs gets invaded by a group of small ones (of ratio ϵ over the whole population). The expected payoff (fitness) of a large crab is:

$$4(1-\epsilon)+9\epsilon = 4+5\epsilon \qquad \text{because it meets a large crab with probability } 1-\epsilon \text{ and a small one with probability } \epsilon.$$

The expected payoff of a small crab is:

$$1(1-\epsilon)+7\epsilon = 1+6\epsilon \qquad \text{because it meets a large crab with probability } 1-\epsilon \text{ and a small one with probability } \epsilon.$$

In this case, for every $\epsilon \in (0,1)$ the payoff of the large crabs is greater than that of the small ones. So, the pure strategy *Large* is an ESS.

3.1.1.2 The big picture

The concept of ESSs can also be extended to mixed strategies. We can think of three natural ways to interpret the notion of probability assignment on the pure strategies of a population. One is, each individual is preprogrammed (through its DNA) to play just a specific pure strategy from a set of strategies and we say that individuals with the same pure strategy are of the same *type*. The group of individuals can be considered to behave as a player with a mixed strategy, defined as a probability vector over the pure strategies used by the group. Each pure strategy's probability equals the ratio of its type's members over the total population (type's *frequency*), because of the simple assumption made, that when two groups conflict one individual from each group is drawn equiprobably to play a bimatrix game. Another one is, each individual is preprogrammed to play a particular mixed strategy. Thus, whoever is drawn will play the specific mixed strategy. The last one is the most general way to think of it, as a blend of the former cases. A group's mixed strategy is defined by its probabilities over the available pure strategies. As soon as one individual is equiprobably picked from each group, the probability over a pure strategy of a group is determined by the sum of the probability each type is picked times the probability this type plays the specific pure strategy. Referring to our previous example (Section 3.1.1.1), the following three infinite populations of crabs are equivalent: (i) One with 2/3 of type *Small* and 1/3 of type *Large*. (ii) One with every crab playing the mixed strategy [2/3: *Small*, 1/3: *Large*]. (iii) One with 1/4 of type *Small*, 1/4 playing the mixed strategy [1/6: *Small*, 5/6: *Large*] and 1/2 playing the mixed strategy [3/4: *Small*, 1/4: *Large*]. Of course in the particular example the individuals cannot have mixed strategies, each one is committed to have a body size for life, but the reasoning holds for other games with strategies that do not exclude each other such as in the Stag-Hunt game. We should mention here, that some games such as Hawk-Dove do not have a pure ESS, but they have a mixed ESS. Other games do not have either.

3.1.2 Previous work

Searching for the exact complexity of deciding if a bimatrix game possesses an ESS, Etessami and Lochbihler [67] invent a nice reduction from the complement of the CLIQUE problem to a specific game with an appointed ESS, showing that the ESS problem is coNP-hard. They also show a polynomial time reduction from the SAT problem to ESS, thus proving that ESS is NP-hard too. This makes impossible for the ESS to be NP-complete, unless NP

=coNP. Furthermore, they provide a proof for the general ESS being contained in Σ_2^P, the second level of the polynomial-time hierarchy, leaving open the question of what is the complexity class in which the problem is complete.

A further improvement of those results was made by Nisan [115], showing that, given a payoff matrix, the existence of a mixed ESS is coDP-hard. The hardness result is due to a relatively simple reduction from the coDP-complete problem co-EXACT-CLIQUE(for the definition see [118]), to ESS. A notable consequence of both [67] and [115] is that the problem of *recognizing* a mixed ESS, once given along with the payoff matrix, is coNP-complete. However, the question of the exact complexity of ESS existence, given the payoff matrix, remained open. A few years later, Conitzer finally settled this question in [49], showing that ESS is actually Σ_2^P-complete.

On the contrary, Hart et al. [127] showed that if the symmetric bimatrix game defined by a $n \times n$ payoff matrix with elements independently randomly chosen according to a distribution F with exponential or faster decreasing tail, such as exponential, normal or uniform, then the probability of having an ESS with just 2 pure strategies in the support tends to 1 as n tends to infinity. In view of this result, and since the basic reduction of [67] used only 3 payoff values, it is interesting to consider whether ESS existence remains hard for arbitrary payoffs in some intervals.

3.1.3 Contribution and a roadmap for the chapter

In the reduction of Etessami and Lochbihler that proves coNP-hardness of ESS the values of the payoffs used, are 0, $\frac{k-1}{k}$ and 1, for $k \in \mathbb{N}$. A natural question is if the hardness results hold when we *arbitrarily* perturb the payoff values within respective intervals (in the spirit of smoothed analysis [138]). In our work we extend the aforementioned reduction and show that the specific reduction remains valid even after significant changes of the payoff values.

We can easily prove that the evolutionarily stable strategies of a symmetric bimatrix game remain the exact same if we add, subtract or multiply (or do all of the aforementioned) with a positive value its payoff matrix. However, that kind of value modification forces the entries of the payoff matrix to change in an entirely correlated manner, hence it does not provide an answer to our question. In this work, we prove that if we partition the payoff matrix into parts of entries with the same value, arbitrary independent perturbations of those values within certain intervals do not affect the validity of our reduction. In other words, we prove that deciding ESS existence remains hard even if we perturb the payoff

values associated with the reduction. En route we give a definition of "reduction robustness under arbitrary perturbations" and show how the reduction under examination adheres to this definition.

In contrast, it was shown in [127] that if the payoffs of a symmetric game are random and independently chosen from the same distribution F with "exponential or faster decreasing tail" (e.g. exponential, normal or uniform), then an ESS (with support of size 2) exists with probability that tends to 1 when n tends to infinity.

One could superficially get a non-tight version of our result by saying that (under supposed continuity assumptions in the ESS definition) any small perturbation of the payoff values will not destroy the reduction. However, in such a case (a) the continuity assumptions have to be precisely stated and (b) this does not explain why the ESS problem becomes easy when the payoffs are random [127].

In fact, the value of our technique is, firstly, to get as tight as possible ranges of the perturbation that preserve the reduction (and the ESS hardness) without any continuity assumptions, and secondly, to indicate the basic difference from random payoff values (which is exactly the notion of partitioning the payoffs into groups in our definition of robustness, and the allowance of arbitrary perturbation within some interval in each group). For the reduction to be preserved when we independently perturb the values (in each of the resulting parts arbitrarily), one must show that a system of inequalities has always a feasible solution, and we manage to show this in our final theorem. Our result seems to indicate that existence of an ESS remains hard despite a smoothed analysis [138].

An outline of the chapter is as follows: In Section 3.2 we define the robust reduction notion and we provide a first extension of the aforementioned reduction by [67]. In Section 3.3 we provide another extended reduction, based on the one from [67], that is essentially modified in order to be robust. In Section 3.4 we give our main result.

3.1.4 Definitions and preliminary results

A central problem of this chapter is the CLIQUE problem.

Definition 6 (CLIQUE)**.** *Given an undirected graph G and a number k, we are asked whether there is a clique of size k.*

A definition of ESS equivalent to that presented in Section 3.1.1 is:

Definition 7 (Evolutionarily stable strategy)**.** *A (mixed) strategy $s \in X$ is an evolutionarily stable strategy (ESS) of a two-player symmetric game Γ if:*

1. (s, s) *is a symmetric NE of* Γ*, and*

2. *if* $t \in X$ *is any best response to* s *and* $t \neq s$*, then* $U_1(s, t) > U_1(t, t)$*.*

Due to [113], we know that every symmetric game has a symmetric Nash equilibrium. The same does not hold for evolutionarily stable strategies (for example "rock-paper-scissors" does not have any pure or mixed ESS). The ESS problem is the following.

Definition 8 (ESS)**.** *Given a symmetric two-player normal-form game* Γ*, we are asked whether there exists an evolutionarily stable strategy of* Γ*.*

Theorem 1 of Motzkin and Straus (see Chapter 2) gives us the following corollary.

Corollary 2. *Let* $G = (V, E)$ *be an undirected graph with maximum clique size* d*. Let* $A_G^{\tau,\rho}$ *be a modified adjacency matrix of graph* G *where its entries with value 0 are replaced by* $\tau \in \mathbb{R}$ *and its entries with value 1 are replaced by* $\rho \in \mathbb{R}$*. Let* $\Delta_1 = \left\{x \in \mathbb{R}^n_{\geq 0} : \sum_{i=1}^n x_i = 1\right\}$*. Then* $\max_{x \in \Delta_1} x^T A_G^{\tau,\rho} x = \tau + (\rho - \tau)\frac{d-1}{d}$*.*

Proof.

$$x^T A_G^{\tau,\rho} x = x^T \left[\tau \cdot \mathbf{1} + (\rho - \tau) \cdot A_G\right] x \quad \text{, where } \mathbf{1} \text{ is the } n \times n \text{ matrix with value 1 in every entry.}$$

$$= \tau + (\rho - \tau) \cdot x^T A_G x \quad \text{, and by Theorem 1 the result follows.}$$

□

3.2 Robust Reductions

Definition 9 (Neighbourhood)**.** *Let* $v \in \mathbb{R}$*. An (open) interval* $I(v) = [a, b]$ *(*$I(v) = (a, b)$*) with* $a < b$ *where* $a \leq v \leq b$*, is called a neighbourhood of* v *of range* $|b - a|$*.*

Definition 10 (Robust reduction under arbitrary perturbations of values)**.** *We are given a valid reduction of a problem to a strategic game that involves a real matrix* A *of payoffs as entries* a_{ij}*.* A *is partitioned into* m *parts, with each part's entries having the same value* $v(t)$*, for* $t \in \{1, 2, ...m\}$*. Let* $I(v(t)) \neq \emptyset$ *be a neighbourhood of* $v(t)$ *and* $w(t) \in I(v(t))$ *be an arbitrary value in that neighbourhood. The reduction is called robust under arbitrary perturbations of values if it is valid for all the possible matrices* W *with entries* $w(t)$*.*

3.2.1 A first extension of the reduction from the complement of the CLIQUE problem to ESS

In the sequel we extend the idea of K. Etessami and A. Lochbihler [67] by giving sufficient conditions in order for the reduction to hold. We replace the zeros and ones of their reduction with rational numbers τ and ρ respectively, and determine their acceptable values so that the reduction still goes through.

Given an undirected graph $G = (V, E)$ we construct the following game $\Gamma_{k,\tau,\rho}(G) = (S, u_1)$ for $\lambda(k) = \frac{k-1}{k}$, where $k \in \mathbb{N}$, and suitable $0 < \tau < \rho < 1$ whose values are determined in Theorem 2. Note that from now on we will only consider rational τ and ρ so that every payoff value of the game is rational.

$S = V \cup \{a, b, c\}$ are the strategies for the players where $a, b, c \notin V$.

$n = |V|$ is the number of nodes.

- $u_1(i, j) = \rho$ for all $i, j \in V$ with $(i, j) \in E$.
- $u_1(i, j) = \tau$ for all $i, j \in V$ with $(i, j) \notin E$.
- $u_1(z, a) = \rho$ for all $z \in S \setminus \{b, c\}$.
- $u_1(a, i) = \lambda(k)$ for all $i \in V$.
- $u_1(y, i) = \rho$ for all $y \in \{b, c\}$ and $i \in V$.
- $u_1(y, a) = \tau$ for all $y \in \{b, c\}$.
- $u_1(z, y) = \tau$ for all $z \in S$ and $y \in \{b, c\}$.

An example of a graph G with 3 nodes is shown in Figure 3.1. The payoff matrix of the strategic game derived from G is shown in Table 3.1. The transpose of it is the payoff matrix of the column-player.

In the sequel we shall use two corollaries of the Motzkin and Strauss theorem, namely, Corollary 2 and Corollary 1.

Theorem 2. *Let $G = (V, E)$ be an undirected graph. The game $\Gamma_{k,\tau,\rho}(G)$ with $\lambda(k) = \frac{k-1}{k}$ and*

- $\rho \in \left(1 - \frac{4}{(n+1)^2}, \quad 1 - \frac{1}{(n+1)^2}\right]$ *and* $\tau \in \left[(1-\rho)(n-1), \quad \rho - (1 - \sqrt{1-\rho})^2\right)$ *or*

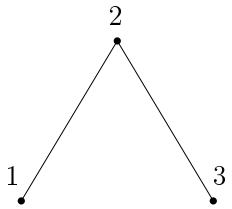


Figure 3.1: Example: the graph G.

	a	b	c	1	2	3
a	ρ	τ	τ	$(k-1)/k$	$(k-1)/k$	$(k-1)/k$
b	τ	τ	τ	ρ	ρ	ρ
c	τ	τ	τ	ρ	ρ	ρ
1	ρ	τ	τ	τ	ρ	τ
2	ρ	τ	τ	ρ	τ	ρ
3	ρ	τ	τ	τ	ρ	τ

Table 3.1: The payoff matrix of the row player in which we have encoded graph G.

- $\rho \in \left(1 - \frac{1}{(n+1)^2}, \quad 1\right)$ *and* $\tau \in \left[(1-\rho)(n-1), \quad (1-\rho)(n-1) + \frac{1}{n+1}\right)$

has an ESS if and only if G has no clique of size k.

Proof. Let $G = (V, E)$ be an undirected graph with maximum clique size d. We consider the game $\Gamma_{k,\tau,\rho}(G)$ above. Suppose s is an ESS of $\Gamma_{k,\tau,\rho}(G)$.

For the reduction we will prove by contradiction three claims, whose combined statements imply that the only possible ESS s of $\Gamma_{k,\tau,\rho}(G)$ is the pure strategy a. Here we should note that these three claims hold not only for the aforementioned intervals of τ and ρ, but for any $\tau, \rho \in \mathbb{R}$ for which $\tau < \rho$.

Claim 1. *The support of any possible ESS s of $\Gamma_{k,\tau,\rho}(G)$ does not contain b or c ($supp(s) \cap \{b, c\} = \emptyset$).*

Proof. Suppose $supp(s) \cap \{b, c\} \neq \emptyset$.

Let $t \neq s$ be a strategy with $t(i) = s(i)$ for $i \in V$, $t(y) = s(b) + s(c)$ and $t(y') = 0$ where $y, y' \in \{b, c\}$ such that $y \neq y'$ and $s(y) = min\{s(b), s(c)\}$. Since $u_1(b, z) = u_1(c, z)$ for all

$z \in S$,

$$U_1(t,s) = \sum_{i \in V} t(i)U_1(i,s) + (t(b)+t(c))U_1(b,s) + t(a)U_1(a,s) \ ,$$
$$U_1(s,s) = \sum_{i \in V} s(i)U_1(i,s) + (s(b)+s(c))U_1(b,s) + s(a)U_1(a,s) \ ,$$

which yields $U_1(t,s) = U_1(s,s)$ and so t is a best response to s. Also,

$$U_1(s,t) = \sum_{i \in V} s(i)U_1(i,t) + (s(b)+s(c))U_1(b,t) + s(a)U_1(a,t) \ ,$$
$$U_1(t,t) = \sum_{i \in V} t(i)U_1(i,t) + (t(b)+t(c))U_1(b,t) + t(a)U_1(a,t) \ ,$$

which yields $U_1(s,t) = U_1(t,t)$. But this is a contradiction since it should be $U_1(s,t) > U_1(t,t)$ as s is an ESS. □

Claim 2. *The support of any possible ESS s of $\Gamma_{k,\tau,\rho}(G)$ contains a.*

Proof. Suppose $supp(s) \subseteq V$.

Let us denote by A_G the adjacency matrix of the graph G. Then,

$$\begin{aligned} U_1(s,s) = \sum_{i,j \in V} s(i)s(j)u_1(i,j) = x^T A_{G,\tau,\rho} x \\ &\leq \tau + (\rho - \tau)\frac{d-1}{d} \quad \text{(by Corollary 2)} \\ &< \rho = U_1(b,s) \qquad \text{for every } \rho > \tau. \end{aligned}$$

But this is a contradiction since s is an ESS and therefore a NE. From Claim 1 and Claim 2, it follows that $a \in supp(s)$, i.e. $s(a) > 0$. □

Claim 3. $s(a) = 1$.

Proof. Suppose $s(a) < 1$.

Since (s,s) is a NE, a is a best response to s and $a \neq s$. Then

$$U_1(s,a) = \sum_{z \in supp(s)} s(z)u_1(s,a) = \rho = U_1(a,a).$$

But this is also a contradiction since it should be $U_1(s,a) > U_1(a,a)$ as s is an ESS. Therefore, the only possible ESS of $\Gamma_{k,\tau,\rho}(G)$ is the pure strategy a. □

Now we show the following lemma, which concludes also the proof of Theorem 2.

Lemma 1. *The game $\Gamma_{k,\tau,\rho}(G)$ with the requirements of Theorem 2 has an ESS (strategy a) if and only if there is no clique of size k in graph G.*

Proof. We consider two cases for k:

Case 1: $d < k$

Let $t \neq a$ be a best response to a. Then $supp(t) \subseteq V \cup \{a\}$.

Let $r = \sum_{i \in V} t(i)$. So $r > 0$ $(t \neq a)$ and $t(a) = 1 - r$. Combining Corollary 2 and 1 we get,

$$\begin{aligned}
U_1(t,t) - U_1(a,t) = &\sum_{i,j \in V} t(i)t(j)u_1(i,j) + r \cdot t(a) \cdot \rho + \\
&+ t(a) \cdot r \cdot \frac{k-1}{k} + t(a)^2 \cdot \rho - \Big[r \cdot \frac{k-1}{k} + t(a) \cdot \rho \Big] \\
\leq &\Big[\tau + (\rho - \tau)\frac{d-1}{d} \Big] r^2 + r(1-r) \cdot \rho + \\
&+ (1-r)r\frac{k-1}{k} + (1-r)^2 \cdot \rho - r\frac{k-1}{k} - (1-r) \cdot \rho \\
= &\Big[\tau + (\rho - \tau)\frac{d-1}{d} \Big] r^2 - \frac{k-1}{k} r^2 \\
= &\frac{r^2}{d} \Big[\tau + \rho(d-1) - d\frac{k-1}{k} \Big] \\
= &\frac{r^2}{d} E \qquad \text{, where } E = \tau + \rho(d-1) - d\frac{k-1}{k} \text{ .}
\end{aligned}$$

If we can show that $E < 0$ then strategy a is an ESS. We now show why $E < 0$:
Let us define the following function,

$$f(k,d,\rho) = d\frac{k-1}{k} - \rho(d-1) \quad \text{, with the constraints: } k \geq d+1, 1 \leq d \leq n \text{ and } \rho \in (0,1) \text{ .}$$

Then we define the function $g(d, \rho)$:

$$g(d,\rho) = \min_k f(k,d,\rho) = d\frac{d}{d+1} - \rho(d-1) = (1-\rho)(d-1) + \frac{1}{d+1} \ . \tag{3.1}$$

By examining the first and second partial derivative of g with respect to variable d, we find the minimum of function $g(d, \rho)$:

$$h(\rho) = \min_d g(d,\rho) = \rho - (1-\sqrt{1-\rho})^2 \qquad \text{, for} \quad d^* = \frac{1}{\sqrt{1-\rho}} - 1 \ . \tag{3.2}$$

Now there are two subcases. The maximum clique size may be impossible to reach the value of d^*, or it could reach it, depending on the size of $n = |V|$.

Subcase i) $n < \frac{1}{\sqrt{1-\rho}} - 1$ or equivalently: $\rho > 1 - \frac{1}{(n+1)^2}$.

From the partial derivatives of function $g(d, \rho)$ with respect to variable d we know that it is a strictly decreasing function for $d < d^*$. And given that $d \leq n$, from (3.1) we get:

$$h(\rho) = (1-\rho)(n-1) + \frac{1}{n+1} \qquad \text{, for} \quad 1 - \frac{1}{(n+1)^2} < \rho < 1 \ . \tag{3.3}$$

Subcase ii) $n \geq \frac{1}{\sqrt{1-\rho}} - 1$ or equivalently: $\rho \leq 1 - \frac{1}{(n+1)^2}$.

By examining the first and second partial derivative with respect to variable ρ, we find the plot of function $h(\rho)$ to be the one shown in Figure 3.2.

As we can see, the maximum of $h(\rho)$ is $\frac{1}{2}$ and it is achieved when $\rho = \frac{3}{4}$.

Interval a) $\frac{3}{4} < \rho \leq 1 - \frac{1}{(n+1)^2}$.

The monotonicity of $h(\rho)$ in this interval implies that its minimum is achieved for $\rho^* = 1 - \frac{1}{(n+1)^2}$. Thus if we want a minimum of h over all ρ, from (3.2) we get:

$$\min_\rho h(\rho) = 1 - \frac{1}{(n+1)^2} - \left(1 - \sqrt{1 - \left(1 - \frac{1}{(n+1)^2}\right)}\right)^2 = \frac{2n}{(n+1)^2} \ . \tag{3.4}$$

Interval b) $0 < \rho \leq \frac{3}{4}$.

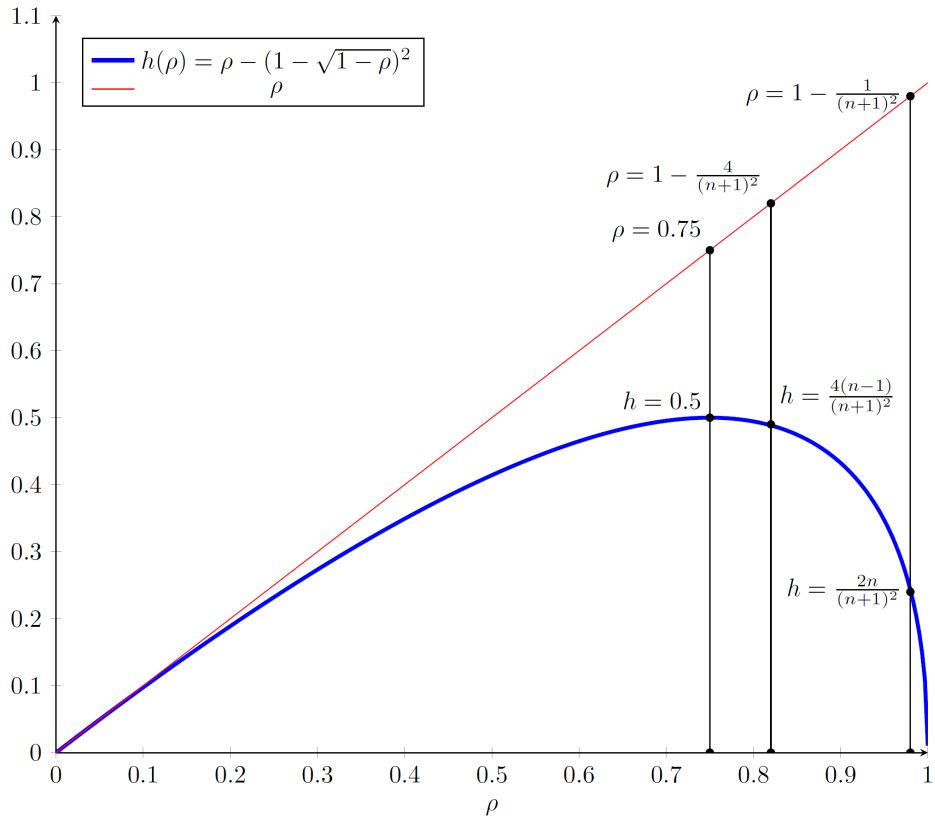


Figure 3.2: The function $h(\rho)$ (thick graph in blue colour).

The monotonicity of $h(\rho)$ in this interval implies that there is no minimum point, but when ρ gets arbitrarily close to zero then $h(\rho)$ goes arbitrarily close to zero as well, i.e. $\lim_{\rho\to 0^+} h(\rho) = 0$.

To sum up:

$$\tau^* = \min_{k,d} f(k,d,\rho) = \begin{cases} \rho - (1-\sqrt{1-\rho})^2 & \text{, if } 0 < \rho \leq 1 - \frac{1}{(n+1)^2} \text{ , from (3.2)} \\ (1-\rho)(n-1) + \frac{1}{n+1} & \text{, if } 1 - \frac{1}{(n+1)^2} < \rho < 1 \text{ , from (3.3)} \end{cases}$$

or if we want the minima to be independent of ρ when possible:

$$\tau^* = \min_{k,d,\rho} f(k,d,\rho) = \begin{cases} \rho - (1-\sqrt{1-\rho})^2 & \text{, if } \quad 0 < \rho \leq \frac{3}{4} \\ \frac{2n}{(n+1)^2} & \text{, if } \quad \frac{3}{4} < \rho \leq 1 - \frac{1}{(n+1)^2} \text{ , from (3.4)} \\ \frac{1}{n+1} & \text{, if } \quad 1 - \frac{1}{(n+1)^2} < \rho < 1 \text{ , from (3.3).} \end{cases}$$

Therefore, depending on the interval that ρ belongs to, we can demand τ to be strictly less than τ^* , making $U_1(t,t) - U_1(a,t)$ negative. We conclude that when $d < k$ then strategy a is an ESS.

Case 2: $d \geq k$

Let $C \subseteq V$ be the vertex set of a clique of G, where $|C| = k$. Then t with $t(i) = \frac{1}{k}$ for $i \in C$ and $t(j) = 0$ for $j \in S \setminus C$ is a best response to a and $t \neq a$, and

$$U_1(t,t) = \sum_{i,j\in C} t(i)t(j)u_1(i,j) = \frac{1}{k^2} \cdot (k-1)k \cdot \rho + \frac{1}{k^2} k \cdot \tau = \frac{(k-1)\rho + \tau}{k} \text{ ,}$$
$$U_1(a,t) = \frac{k-1}{k} \text{ .}$$

Then,

$$U_1(t,t) - U_1(a,t) = \frac{1}{k}\Big[\tau - (1-\rho)(k-1)\Big] \\ = \frac{1}{k}E' \qquad \text{, where } E' = \tau - (1-\rho)(k-1)\ .$$

If $E' \geq 0$ then a cannot be an ESS. We explain why $E' \geq 0$:
Let's define the following function:

$$y(k,\rho) = (1-\rho)(k-1) \qquad \text{, with the constraints: } k \leq d \text{ and } \rho \in (0,1)\ .$$

Then we define the function $z(d,\rho)$:

$$z(d,\rho) = \max_k y(k,\rho) = (1-\rho)(d-1)\ ,$$

so,

$$\tau^{**} = \max_d z(d,\rho) = (1-\rho)(n-1)\ .$$

Now, given that τ needs to be at least τ^{**} but strictly less than τ^* the following should hold:

$$(1-\rho)(n-1) < \rho - (1-\sqrt{1-\rho})^2 \quad \text{, or equivalently, } \quad \rho > 1 - \frac{4}{(n+1)^2}\ .$$

So we conclude that when $d \geq k$ then strategy a is not an ESS. This completes the proof of Lemma 1. □

This completes the proof of Theorem 2. □

Corollary 3. *The* ESS *problem with payoff values in the domains given in Theorem 2 is* coNP*-hard.*

3.3 Extending the Reduction with Respect to $\lambda(k)$

We now prove a generalization of the latter reduction for $\lambda(k) = 1 - \frac{1}{k^x}$, with $x \geq 3$. In this section one can see that τ, ρ are non-negative but they are not always strictly smaller than 1. It is easy to show that adding, subtracting or multiplying with a positive number the

payoff matrix of an ESS instance, the set of ESSs remains the same. The proof is similar to the one which shows that for the aforementioned operations on the payoff matrices of a general strategic game the set of Nash equilibria remains the same. Therefore, we can always scale down the payoffs in our ESS instances and have a normalized payoff matrix with payoffs in $[0, 1]$.

Theorem 3. *Let $G = (V, E)$ be an undirected graph. The game $\Gamma^x_{k,\tau,\rho}(G)$ with $\lambda(k) = 1 - \frac{1}{k^x}$, for $x \geq 3$ and*

- $\rho \in \left(1 + \frac{n^{x-1}-2^x}{2^x n^{x-1}(n-1)}, \quad 1 + \frac{(n+1)^x - n2^x}{2^x (n+1)^x (n-1)}\right]$ *and*
  $\tau \in \left[(1-\rho)(n-1) + 1 - \frac{1}{n^{x-1}}, \quad 1 - \frac{1}{2^x}\right)$

  or

- $\rho \in \left(1 + \frac{(n+1)^x - n2^x}{2^x (n+1)^x (n-1)}, \quad +\infty\right)$ *and*
  $\tau \in \left[(1-\rho)(n-1) + 1 - \frac{1}{n^{x-1}}, \quad (1-\rho)(n-1) + 1 - \frac{n}{(n+1)^x}\right)$

has an ESS if and only if G has no clique of size k.

Proof. Let $G = (V, E)$ be an undirected graph with maximum clique size d. We consider the game $\Gamma_{k,\tau,\rho}(G)$ defined in Section 3.2.1, with the only difference that now, we substitute payoffs of value $\frac{k-1}{k}$ with new payoffs $\frac{k^x-1}{k^x}$, meaning we make the change $k \leftarrow k^x$. Suppose s is an ESS of $\Gamma^x_{k,\tau,\rho}(G)$.

In this case, the same analysis as in Section 3.2.1 is similarly applied up to the point where we prove that the only possible ESS of $\Gamma^x_{k,\tau,\rho}(G)$ is the pure strategy a. Now we proceed to show the following lemma, which concludes also the proof of Theorem 3.

Lemma 2. *The game $\Gamma^x_{k,\tau,\rho}(G)$ with the requirements of Theorem 3 has an ESS (strategy a) if and only if there is no clique of size k in graph G.*

Proof. We consider again two cases for k:

Case 1: $d < k$

Let $t \neq a$ be a best response to a. Then $supp(t) \subseteq V \cup \{a\}$.

Let $r = \sum_{i \in V} t(i)$. So $r > 0, (t \neq a)$ and $t(a) = 1 - r$. Combining Corollary 2 and 1 we get,

$$
\begin{aligned}
U_1(t,t) - U_1(a,t) = &\sum_{i,j \in V} t(i)t(j)u_1(i,j) + r \cdot t(a) \cdot \rho + \\
&+ t(a) \cdot r \cdot \frac{k^x - 1}{k^x} + t(a)^2 \cdot \rho - \Big[r \cdot \frac{k^x - 1}{k^x} + t(a) \cdot \rho \Big] \\
\leq &\Big[\tau + (\rho - \tau)\frac{d-1}{d} \Big] r^2 + r(1-r) \cdot \rho + \\
&+ (1-r) r \frac{k^x - 1}{k^x} + (1-r)^2 \cdot \rho - r\frac{k^x - 1}{k^x} - (1-r) \cdot \rho \\
= &\Big[\tau + (\rho - \tau)\frac{d-1}{d} \Big] r^2 - \frac{k^x - 1}{k^x} r^2 \\
= &\frac{r^2}{d} \Big[\tau - (1-\rho)(d-1) - (1 - \frac{d}{k^x}) \Big] \\
= &\frac{r^2}{d} E \qquad \text{, where } E = \tau - (1-\rho)(d-1) - (1 - \frac{d}{k^x}) \ .
\end{aligned}
$$

If we can show that $E < 0$ then strategy a is an ESS. We show why $E < 0$:

Let's define the following function:

$$f(k,d,\rho) = (1-\rho)(d-1) + 1 - \frac{d}{k^x} \text{ , with the constraints: } k \geq d+1, 1 \leq d \leq n, x \geq 3 \ .$$

Then we define the function $g(d,\rho)$:

$$g(d,\rho) = \min_k f(k,d,r) = (1-\rho)(d-1) + 1 - \frac{d}{(d+1)^x} \ .$$

Now, the first two partial derivatives of $g(d,\rho)$ with respect to variable d, are:

$$
\begin{aligned}
&\frac{\partial g(d,\rho)}{\partial d} = (1-\rho) + \frac{(x-1)d - 1}{(d+1)^{x+1}} \\
&\frac{\partial^2 g(d,\rho)}{\partial d^2} = \frac{-x[(x-1)d - 2]}{(d+1)^{x+2}}, \qquad \text{which is non-positive for } d \geq 1, x \geq 3 \ .
\end{aligned}
$$

This means that function g has its minimum either at $d = 1$ or $d = n$:

$$
\begin{aligned}
&g(1,\rho) = 1 - \frac{1}{2^x} \\
&g(n,\rho) = (1-\rho)(n-1) + 1 - \frac{n}{(n+1)^x}
\end{aligned}
$$

If the minimum is $g(1,\rho)$:

$$g(1,\rho) \leq g(n,\rho), \text{ or equivalently, } \rho \leq 1 + \frac{(n+1)^x - n2^x}{2^x(n+1)^x(n-1)} .$$

Then,

$$h(\rho) = \min_d g(d,\rho) = 1 - \frac{1}{2^x} .$$

If the minimum is $g(n,\rho)$:

$$g(n,\rho) < g(1,\rho), \text{ or equivalently, } \rho > 1 + \frac{(n+1)^x - n2^x}{2^x(n+1)^x(n-1)} .$$

Then,

$$h(\rho) = \min_d g(d,\rho) = (1-\rho)(n-1) + 1 - \frac{n}{(n+1)^x} .$$

So, following the notation we used in Section 3.2.1:

$$\tau^* = \min_{k,d} f(k,d,\rho) = \begin{cases} 1 - \frac{1}{2^x} & \text{, if } \rho \leq 1 + \frac{(n+1)^x - n2^x}{2^x(n+1)^x(n-1)} \\ (1-\rho)(n-1) + 1 - \frac{n}{(n+1)^x} & \text{, if } \rho > 1 + \frac{(n+1)^x - n2^x}{2^x(n+1)^x(n-1)}. \end{cases}$$

Therefore, we can demand τ to be strictly less than τ^*, making $U_1(t,t) - U_1(a,t)$ negative. We conclude that when $d < k$ then strategy a is an ESS.

Case 2: $d \geq k$

Let $C \subseteq V$ be a clique of G of size k. Then t with $t(i) = \frac{1}{k}$ for $i \in C$ and $t(j) = 0$ for $j \in S \setminus C$ is a best response to a and $t \neq a$, and

$$U_1(t,t) = \sum_{i,j \in C} t(i)t(j)u_1(i,j) = \frac{1}{k^2} \cdot (k-1)k \cdot \rho + \frac{1}{k^2} k \cdot \tau = \frac{(k-1)\rho + \tau}{k} ,$$
$$U_1(a,t) = \frac{k^x - 1}{k^x} .$$

Then,

$$U_1(t,t) - U_1(a,t) = \frac{1}{k}\Big[\tau - (1-\rho)(k-1) - (1 - \frac{1}{k^{x-1}})\Big]$$
$$= \frac{1}{k}E' \qquad \text{, where } E' = \tau - (1-\rho)(k-1) - (1 - \frac{1}{k^{x-1}}) \ .$$

If $E' \geq 0$ then a cannot be an ESS. We explain why $E' \geq 0$:
Let's define the following function:

$$y(k,\rho) = (1-\rho)(k-1) + 1 - \frac{1}{k^{x-1}} \quad \text{, with the constraints: } k \leq d \ .$$

Then we define the function $z(d,\rho)$:

$$z(d,\rho) = \max_k y(k,\rho) = (1-\rho)(d-1) + 1 - \frac{1}{d^{x-1}} \ ,$$

so,

$$\tau^{**} = \max_d z(d,\rho) = (1-\rho)(n-1) + 1 - \frac{1}{n^{x-1}} \ .$$

Now, given that τ needs to be at least τ^{**} but strictly less than τ^* the following should hold:

$$(1-\rho)(n-1) + 1 - \frac{1}{n^{x-1}} < 1 - \frac{1}{2^x} \ \text{, or equivalently, } \rho > 1 + \frac{n^{x-1} - 2^x}{2^x n^{x-1}(n-1)} \ .$$

So we conclude that when $d \geq k$ then strategy a is not an ESS. This completes the proof of Lemma 2. □

This concludes the proof of Theorem 3. □

Corollary 4. *The* ESS *problem with payoff values in the domains given in Theorem 3 is* coNP*-hard.*

3.4 The Main Result

Now we can prove our main theorem:

Theorem 4. *Consider the numbers* $x_0 \geq 3$, $x_1 \in (x_0, x_0 \log_n(n+1))$, $C = \frac{(n+1)^{x_0} - n^{x_1}}{n^{x_1-1}(n+1)^{x_0}(n-1)}$, $D = C(n-1)$, $E = \frac{(n+1)^{x_0} - n2^{x_0}}{2^{x_0}(n+1)^{x_0}(n-1)}$. *Any reduction as in Theorem 3 for* $x = x_0$ *from the complement of the* CLIQUE *problem to the* ESS *problem is robust under arbitrary perturbations of values in the intervals:*

$$\tau \in \left[1 - \frac{1}{2^{x_0}} - D, \quad 1 - \frac{1}{2^{x_0}} - D + B\right),$$
$$\rho \in (1 + E, \quad 1 + E + A),$$
$$\lambda \in \left[1 - \frac{1}{k^{x_0}}, \quad 1 - \frac{1}{k^{x_1}}\right],$$

for any $A \in (0, C)$ *and* $B = (C - A)(n-1)$.

Proof. We partition the game's payoff matrix U in three disjoint sets: $U_\tau, U_\rho, U_\lambda$ with $U_\tau \cup U_\rho \cup U_\lambda = U$ and values τ, ρ, λ of their entries respectively. Each set's entries have the same value. For every $\lambda \in \left[1 - \frac{1}{k^{x_0}}, 1 - \frac{1}{k^{x_1}}\right]$ there is a $x = -\log_k(1-\lambda)$ in the interval $[x_0, x_1]$ such that $\lambda = 1 - \frac{1}{k^x}$, where $x_0 \geq 3$ and $x_1 \in (x_0, x_0 \log_n(n+1))$. We will show that, for this x, any reduction with the values of τ, ρ in the respective intervals stated in Theorem 3, is valid.

In Figure 3.3, we show the validity area of τ depending on ρ with parameter x, due to Theorem 3. The thin and thick plots bound the validity area (shaded) for $x = x_0$ and $x = x_1$ respectively.

While x increases, the parallel lines of the lower and upper bound of τ move to the right, the horizontal line of the upper bound of τ moves up, and the left acute angle as well as the top obtuse angle of the plot move to the left (by examination of the monotonicity of those bounds with respect to x).

The lower bound of τ for an $x = x' > x_0$ equals the upper bound of τ for $x = x_0$, when $x' = x_0 \log_n(n+1)$. Thus, for all $x \in (x_0, x_0 \log_n(n+1))$ there is a non-empty intersection between the validity areas. We have picked an $x = x_1 \in (x_0, x_0 \log_n(n+1))$.

In Figure 3.4, we show a zoom-in of the intersection of the validity areas of Figure 3.3. Let the intersection of lines: $1 - \frac{1}{2^{x_0}}$, $(1-\rho)(n-1) + 1 - \frac{1}{n^{x_1-1}}$ be at point $\rho = \rho_C$.

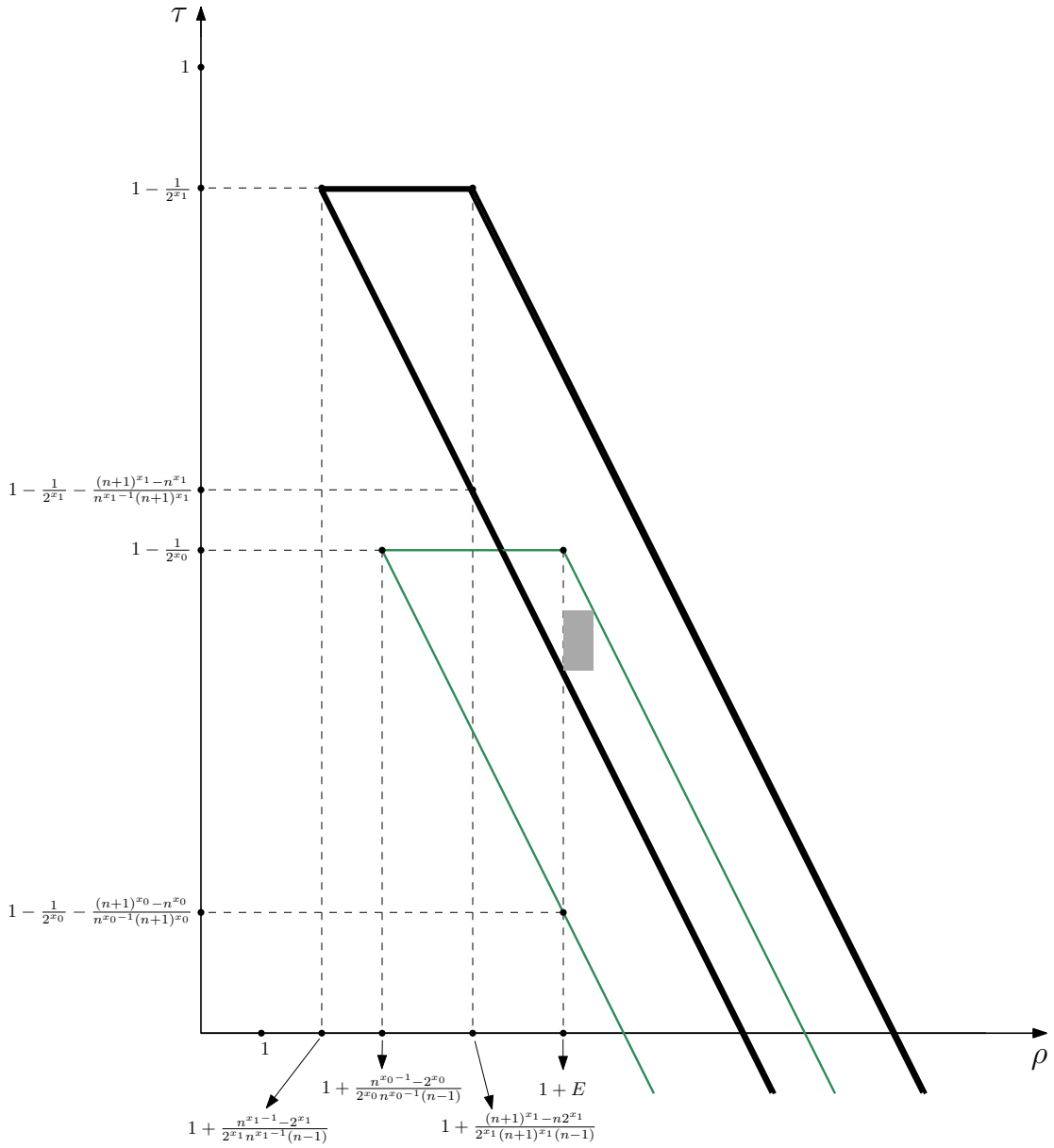


Figure 3.3: The validity area of τ and ρ with parameter x.

Then,

$$(1-\rho_C)(n-1)+1-\frac{1}{n^{x_1-1}}=1-\frac{1}{2^{x_0}}$$

$$\text{or equivalently, } \rho_C=1-\frac{1}{2^{x_0}(n-1)}-\frac{1}{n^{x_1-1}(n-1)}\,.$$

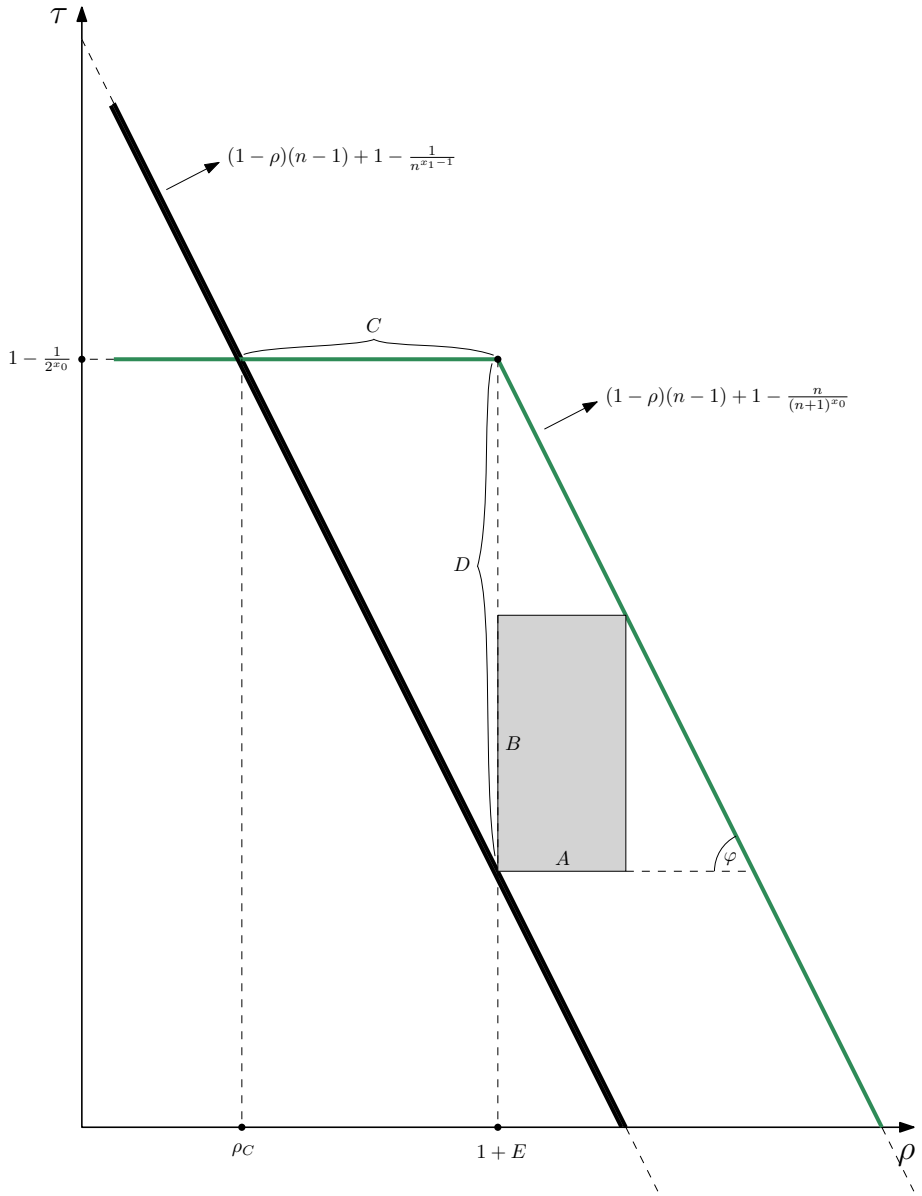


Figure 3.4: Detail of the validity areas' intersection and the ρ, τ robust area (shaded).

So,

$$\begin{aligned} C &= 1 + \frac{(n+1)^{x_0} - n2^{x_0}}{2^{x_0}(n+1)^{x_0}(n-1)} - \rho_C \\ &= \frac{(n+1)^{x_0} - n^{x_1}}{n^{x_1-1}(n+1)^{x_0}(n-1)}. \end{aligned}$$

From the upper bound of τ as a function of ρ we can see that $\tan\varphi = n-1$. Thus,

$$D = C\tan\varphi, \text{ or equivalently, } D = \frac{(n+1)^{x_0} - n^{x_1}}{n^{x_1-1}(n+1)^{x_0}}.$$

Now we can pick any $A \in (0, C)$. So, it must be

$$B = (C-A)\tan\varphi, \text{ or equivalently, } B = (n-1)(C-A).$$

For the rectangle with sides A, B shown in Figure 3.4, the reduction is valid for all $x \in [x_0, x_1]$, thus for all $\lambda \in \left[1 - \frac{1}{k^{x_0}}, 1 - \frac{1}{k^{x_1}}\right]$. This completes the proof of Theorem 4. $\square$

Part II

Games between Rational and Intelligent Entities

Chapter 4

Strategic Contention Resolution

In this chapter we study a game in an unstructured, finite population. The problem under examination is to resolve the contention in a communication network with selfish users. In a *contention game* each of $n \geq 2$ identical players has a single information packet that she wants to transmit using one of $k \geq 1$ multiple-access channels. To do that, a player chooses a slotted-time protocol that prescribes the probabilities with which at a given time-step she will attempt transmission at each channel. If more than one players try to transmit over the same channel (collision) then no transmission happens on that channel. Each player tries to minimize her own expected *latency*, i.e. her expected time until successful transmission, by choosing her protocol. The natural problem that arises in such a setting is, given n and k, to provide the players with a common, anonymous protocol (if it exists) such that no one would unilaterally deviate from it (equilibrium protocol).

All previous theoretical results on strategic contention resolution examine only the case of a single channel and show that the equilibrium protocols depend on the feedback that the communication system gives to the players. Here we present multi-channel equilibrium protocols in two main feedback classes, namely *acknowledgement-based* and *ternary*. In particular, we provide equilibrium characterizations for more than one channels, and give specific anonymous, equilibrium protocols with finite and infinite expected latency. In the equilibrium protocols with infinite expected latency, all players transmit successfully in optimal time, i.e. $\Theta(n/k)$, with probability tending to 1 as $n/k \to \infty$.

The results of this chapter have been published in the Proceedings of the 11th International Symposium on Algorithmic Game Theory (SAGT 2018) [46], and in the Proceedings of the 16th Workshop on Approximation and Online Algorithms (WAOA 2018) [47] (co-

authored with Christodoulou and Spirakis).

4.1 Overview

4.1.1 Motivation

In the last sixteen years a great number of works in the Electrical and Electronics Engineering community has been devoted to designing medium access control (MAC) protocols that achieve high throughput. Their main approach is to consider, instead of the initial single-channel scheme, multi-channel schemes (*multi-channel* MAC protocols) which resolve contention caused by packet collisions (e.g. [40,110,130,135,136,145]). Apart from high throughput, an additional benefit of introducing more channels in such a system is robustness, meaning no great dependence on a single channel's functionality. However, to the authors' knowledge, *strategic* behaviour study in multi-channel systems is limited to the Aloha protocol ([98]), contrary to the case of single-channel systems (e.g. [10,43–45,69]). In this work, we examine the problem of *strategic contention resolution* in multi-channel systems, where obedience to a suggested protocol is not required. We seek only *anonymous*, equilibrium protocols, that is, protocols which do not use player IDs. If a player's protocol depends on her ID, then equilibria are simple, but can be unfair as well; scheduling each player's transmission through a priority queue according to her ID is an equilibrium.

We provide two types of equilibrium protocols. The first type, called *FIN-EQ*, describes an anonymous, equilibrium protocol that yields finite expected time of successful transmission (*latency*) to a player. Similarly, the second type, called *IN-EQ*, describes an anonymous, equilibrium protocol which yields infinite expected latency to a player but is also *efficient*, i.e, all players transmit successfully within $\Theta(\frac{\#players}{\#channels})$ time with high probability. In this chapter, we say that the expected latency is "infinite" when it equals the number of time-steps t that the protocol runs for; the term comes from the fact that $t \to \infty$. We study equilibria for two classes of feedback protocols: (a) acknowledgement-based protocols, where the user gets just the information of whether she had a successful transmission or not, only when she tries to transmit her packet, and (b) protocols with ternary feedback, where the user is informed about the number of pending players in each time-step regardless of whether she attempted transmission or not. Previous results on these classes of protocols have been produced only for the case of a single transmission channel ([43,69]). Here we investigate the multiple-channels case.

In the last part of this chapter we seek efficient protocols for both feedback classes. Due to an impossibility result that we show (Theorem 11), the technique used in [69] by Fiat et al. for the single-channel setting in order to provide a FIN-EQ that is also efficient, cannot be applied when there are more than one channel. This fact discourages us from searching for efficient FIN-EQ protocols and, instead, points to the search for efficient IN-EQ protocols, which indeed we find. One could argue that an anonymous protocol with infinite expected time until successful transmission, such as the IN-EQ protocols we provide, does not incentivize a player to participate in such a communication system. To this we reply that finite but exponential waiting-time for a large amount of players (see protocol in Theorem 10, Section 4.3.2) is equally bad for a player, since waiting for e.g. e^{10} msec in Real-Time-Communications is like waiting forever. In other words, if a FIN-EQ protocol with exponential expected latency is acceptable, then so is an IN-EQ protocol.

4.1.2 Contribution and a roadmap for the chapter

The main contributions of this work are the characterizations of FIN-EQ and IN-EQ protocols in the two aforementioned feedback classes. Note that in the current literature regarding the single-channel setting, there are no characterizations of equilibrium in acknowledgement-based protocols. Also, in the single-channel setting the existence of a symmetric equilibrium with finite expected latency in the class of acknowledgement-based protocols remains an open problem, even for three players. However, for the settings with 2 and 3 transmission channels, we present simple anonymous FIN-EQ protocols for up to 4 and 5 players respectively. Furthermore, these protocols are memoryless, while the only known FIN-EQ protocol in the single-channel setting ([43]) is not.

This chapter is organized in three main parts. Section 4.2 deals with FIN-EQ protocols in the acknowledgement-based feedback setting. In that section we give two characterizations of equilibrium and also provide FIN-EQ protocols for specific numbers of players and channels. Section 4.3 deals with FIN-EQ protocols in the ternary feedback setting and extends the corresponding results for the single-channel setting by Fiat et al. [69]. Finally, in Section 4.4, IN-EQ protocols with deadline are provided with the property that the time until all n players transmit successfully is $\Theta(n/k)$ with high probability, when there are k channels. The latter result makes clear the advantage (with respect to time efficiency) that multiple channels bring to a system with strategic users, which is that the time until all players transmit successfully with high probability is inversely proportional to the number

of available channels.

4.1.3 Related work

Contention in telecommunications is a major problem that results to poor throughput due to packet collisions. Motivated mainly by this problem, many works studying conflict-resolution protocols emerged in the late 70's ([36,37,83,121,142]). Their approach is to resolve a collision when it occurs, and only then allow further transmissions on the channel. In those works the user's packets are assumed either to be generated by some stochastic process, or to appear at the same time in a worst-case scenario. Here, we consider the latter setting, i.e. a worst-case model of slotted time, where at any time-step all users have a packet ready to be transmitted (for an example of a similar bursty-input case, see [22]). As stated in [77], even though real implementations of multiple-access channels do not fit precisely within the slotted-time model, it can be shown (e.g. [82,91]) that results obtained in this model do apply to realistic multiple-access channels.

Also, many works have examined multiple-channel communication protocols. In the data link layer, a Medium Access Control (MAC) protocol is responsible for the flow of data through a multiple-access medium. Our multiple-channels model is motivated by theoretical and experimental results which have shown that higher throughput and lower delay is achieved by using "multi-channel" MAC protocols (see [110,114,135,136]). In [136], the *multi-channel hidden terminal problem* is raised which, additionally to increased packet collisions, results to incapability of the users to "sense" more than one channels at a time (possibly none); therefore a user might not know whether another user transmitted successfully or not (see also [141] for the classical "hidden terminal problem"). This motivates us for the consideration of feedback protocols with minimum feedback, i.e. "acknowledgement-based" protocols (see par.2, Section 4.1.1). Also, settings with stronger feedback have been studied (e.g. the Aloha protocol in [98]) in which a user is informed about the number of users that have not transmitted successfully yet. This is why we consider "ternary feedback" protocols (see par.2, Section 4.1.1).

Apart from the latter, all of the aforementioned works assume that the users blindly follow the given protocol, i.e. the users are not strategic. Contention resolution with strategic users has been studied only in single-channel settings or in the special case of the multiple-channel Aloha protocol. Some interesting cooperative and noncooperative models of slotted Aloha have been analysed in [9,97,98]. Aiming to understand the properties of

contention resolution under selfishness, apart from various feedback settings, many cost functions have also been studied. One of the most meaningful cost functions is the one that models non-zero transmission costs as in [45] (and also [10,98]).

The theoretical works that relate the most to the current work are the seminal paper by Fiat, Mansour and Nadav [69] and two by Christodoulou et al. [43,44] which study protocols for strategic contention resolution with zero transmission costs. These works examine the case of a single transmission channel only. In [69] the feedback is ternary. In that work, a characterization of symmetric equilibrium is provided, along with an efficient FIN-EQ protocol that puts an extremely costly equilibrium after a deadline in order to force users to be obedient. The feedback model of [43] and [44] is the acknowledgement-based. Among other results, [43] provides the unique FIN-EQ protocol for the case of two players and a deadline IN-EQ protocol for at least three players.

4.1.4 The model and definitions

Game structure. We define a *contention game* as follows. Let $N = \{1, 2, \ldots, n\}$ be the set of players, also denoted by $[n]$, and $K = \{1, 2, \ldots, k\}$ the set of channels. Each player has a single packet that she wants to send through a channel in K, without caring about the identity of the channel. All players know n and K. We assume synchronous communications with discretized time, i.e. time slots $t = 1, 2, \ldots$. The players that have not yet successfully transmitted their packet are called *pending* and initially all n players are pending. At any given time slot t, a pending player i has a set $A = \{0, 1, 2, \ldots, k\}$ of *pure strategies*: a pure strategy $a \in A$ is the action of choosing channel $a \in K$ to transmit her packet on, or no transmission ($a = 0$). At time t, a *(mixed) strategy* of a player i is a probability distribution over A that potentially depends on information that i has gained from the process based on previous transmission attempts. If exactly one player transmits on a channel in a given slot t, then her transmission is *successful*, the successful player exits the game (i.e. she is no longer pending), and the game continues with the rest of the players. On the other hand, whenever two or more players try to access the same channel (i.e. transmit) at the same time slot, a *collision* occurs and their transmissions fail, in which case the players remain in the game. If at some time slot $k' \leq k$ players are the only ones attempting transmission, and each of them attempts on a distinct channel then all of them are successful. The game continues until all players have successfully transmitted their packets.

Transmission protocols. Let $X_{i,t} \in A$ be the channel-indicator variable that keeps track of the identity of the channel where player i attempted transmission at time t; value 0 indicates no transmission attempt. For any $t \geq 1$, we denote by $\vec{X}_t$ the transmission vector at time t, i.e. $\vec{X}_t = (X_{1,t}, X_{2,t}, \ldots, X_{n,t})$.

An *acknowledgement-based* protocol uses very limited channel feedback. After each time step t, only players that attempted a transmission receive feedback, and the rest get no information. In fact, the information received by a player i who transmitted during t is whether her transmission was successful (in which case she gets an acknowledgement and exits the game) or whether there was a collision.

In a protocol with *ternary feedback* every pending player in every round is informed about the number of remaining players $m \leq n$. This information is given to the players regardless of their transmission history.

Let $\vec{h}_{i,t}$ be the vector of the *personal transmission history* of player i up to time t, i.e. $\vec{h}_{i,t} = (X_{i,1}, X_{i,2}, \ldots, X_{i,t})$. We also denote by $\vec{h}_t$ the transmission history of all players up to time t, i.e. $\vec{h}_t = (\vec{h}_{1,t}, \vec{h}_{2,t}, \ldots \vec{h}_{n,t})$. A *decision rule* $f_{i,t}$ for a pending player i at time t, is a function that maps $\vec{h}_{i,t-1}$ to a strategy $\vec{P}_{i,t}$, with elements $\Pr(X_{i,t} = a|\vec{h}_{i,t-1})$ for all $a \in A$. When the transmission probability on some $a' \in A$ is not stated in a decision rule it is because it can be deduced from the stated ones.

For a player $i \in N$, a *(transmission) protocol* f_i is a sequence of decision rules $f_i = \{f_{i,t}\}_{t\geq 1} = f_{i,1}, f_{i,2}, \ldots$. Given a protocol f_i for player i, when her decision rules depend on the number of pending players and the personal history of i, then we describe them by the player's probability distribution on the action set A. In this case, we denote by $p^{i,a}_{m,t}$ the probability of player i choosing action a at time t given her personal history h_{t-1} when m players are pending right before t. When the context is clear enough we will drop some of the indices accordingly.

When we state that the players use an *anonymous* protocol f, we will mean that they follow a common protocol $f(= f_1 = \cdots = f_n)$ whose decision rules do not depend on any ID of the player (in our setting players do not have IDs), i.e. the decision rule assigns the same strategy to all players with the same personal history. In particular, for any two players $i \neq j$ and any $t \geq 0$, if $\vec{h}_{i,t-1} = \vec{h}_{j,t-1}$, it holds that $f_{i,t}(\vec{h}_{i,t-1}) = f_{j,t}(\vec{h}_{j,t-1})$. In this case, we drop the subscript i in the notation and write f instead of f_i.

A protocol f_i for player i is a *deadline protocol with deadline* t_0 if and only if there exists a finite $t_0 \geq 1$ such that a particular channel $a_i \in K$ is assigned (deterministically or stochastically) to player i at some time $t \leq t_0$ and $\Pr(X_{i,t} = a_i|\vec{h}_{i,t-1}) = 1$ for every time

slot $t \geq t_0$ and any history $\vec{h}_{i,t-1}$.

Efficiency. Assume that all n players follow an anonymous protocol f. We will call f *efficient* if and only if all players will have successfully transmitted by time $\Theta(n/k)$ with high probability (i.e. with probability tending to 1, as $n/k \to \infty$).

Individual utility. By *protocol profile* $\vec{f} = (f_1, f_2, \ldots, f_n)$ we will call the n-tuple of the players' protocols. For a given transmission sequence $\vec{X}_1, \vec{X}_2, \ldots$, which is consistent with $\vec{f}$, define the *latency* of agent i as $T_i \triangleq \inf\{t : X_{i,t} = a, X_{j,t} \neq a, \text{ for some } a \in K, \forall j \neq i\}$. That is, T_i is the time at which i successfully transmits. Also, define the *finishing time* of $\vec{f}$ as $T \triangleq \sup_i\{T_i\}$, i.e., the least time at which all players have successfully transmitted. Given a transmission history $\vec{h}_t$, the n-tuple of protocols $\vec{f}$ induces a probability distribution over sequences of further transmissions. In that case, we write $C_i^{\vec{f}}(\vec{h}_t) \triangleq \mathbb{E}[T_i|\vec{h}_t, \vec{f}] = \mathbb{E}[T_i|\vec{h}_{i,t}, \vec{f}]$ for the expected latency of a pending agent i given that her current history is $\vec{h}_{i,t}$ and from $t+1$ on she follows f_i. For anonymous protocols, i.e. when $f_1 = f_2 = \cdots = f_n = f$, we will simply write $C_i^f(\vec{h}_t)$ instead. Abusing notation slightly, we will also write $C_i^{\vec{f}}(\vec{h}_0)$ for the *unconditional* expected latency of player i induced by $\vec{f}$. We also define the expected future latency $F_i^{\vec{f}}(\vec{h}_t) \triangleq C_i^{\vec{f}}(\vec{h}_t) - t$ and again, whenever clear from the context, we omit redundant indices or vectors from the notation.

Equilibria. The objective of every player is to minimize her expected latency. We call a protocol g_i a *best response* of player i to the *partial protocol profile* $\vec{f}_{-i}$ if for any transmission history $\vec{h}_t$, player i cannot decrease her expected latency by unilaterally deviating from g_i after t. That is, for all time slots t, and for all protocols f_i' for player i, we have

$$C_i^{(\vec{f}_{-i}, g_i)}(\vec{h}_t) \leq C_i^{(\vec{f}_{-i}, f_i')}(\vec{h}_t),$$

where $(\vec{f}_{-i}, g_i)$ (respectively, $(\vec{f}_{-i}, f_i')$) denotes the *protocol profile* where every player $j \neq i$ uses protocol f_j and player i uses protocol g_i (respectively f_i'). For an anonymous protocol f, we denote by (f_{-i}, g_i) the profile where player $j \neq i$ uses protocol f and player i uses protocol g_i.

We say that $\vec{f} = (f_1, f_2, \ldots f_n)$ is an *equilibrium* if for any transmission history $\vec{h}_t$ the players cannot decrease their expected latency by unilaterally deviating after t; that is, for every player i, f_i is a best response to $\vec{f}_{-i}$.

FIN-EQ and IN-EQ protocols. We call an anonymous protocol *FIN-EQ* if it is an equilibrium protocol and yields finite expected latency to a player. Similarly, we call an anonymous protocol *IN-EQ* if it is an equilibrium protocol, yields infinite expected latency to a player, and is also efficient.

4.2 Equilibrium for Acknowledgement-based Protocols

4.2.1 Nash equilibrium characterizations

The following equilibrium characterizations for the class of acknowledgement-based protocols help us check whether the protocols we subsequently guess are equilibrium protocols. The characterizations are for symmetric and asymmetric equilibria, arbitrary number of channels $k \geq 1$ and number of players $n \geq 2$.

In an acknowledgement-based protocol, the actions of player i at time t depend only (a) on her personal history $\vec{h}_{i,t-1}$ and (b) on whether she is pending or not at t. Let $\vec{f} = (f_1, f_2, \ldots, f_n)$ be a tuple of acknowledgement-based protocols (not necessarily anonymous) for the n players. For a (finite) positive integer τ^*, and a given history $h_{i,\tau^*} = (a_{i,1}, a_{i,2}, \ldots, a_{i,\tau^*})$, define for player i the protocol

$$g_i = g_i(h_{i,\tau^*}) \triangleq \begin{cases} (\Pr\{X_{i,t} = a_{i,t}\} = 1,\ \Pr\{X_{i,t} \neq a_{i,t}\} = 0) & \text{, for } 1 \leq t \leq \tau^* \\ f_{i,t}, \quad \text{for } t > \tau^*. \end{cases} \tag{4.1}$$

A personal history $\vec{h}_{i,\tau^*}$ is *consistent with* the protocol profile $\vec{f}$ if and only if there is a non-zero probability that $\vec{h}_{i,\tau^*}$ will occur for player i under $\vec{f}$. Protocol $g_i(h_{i,\tau^*})$ is *consistent* with $\vec{f}$ if and only if h_{i,τ^*} is consistent with $\vec{f}$, and when clear from the context we write g_i instead. We denote the set of all g_i's, that is, all $g_i(h_{i,t})$'s for all $t \geq 1$, which are consistent with $\vec{f}$, by $\mathcal{G}_i^{\vec{f}}$. If $f_i = f\ \forall i$ (i.e. f is anonymous), then instead of g_i and $\mathcal{G}_i^{\vec{f}}$ we write g and $\mathcal{G}^f$ respectively.

Lemma 3 (Equilibrium characterization 1)**.** *Consider a profile $\vec{f} = (f_1, f_2, \ldots f_n)$ of acknowledgement-based protocols and a protocol $g_i = g_i(h_{i,\tau^*})$ for some $\tau^* \geq 1$. The following statements are equivalent:*
(i) $\vec{f}$ is an equilibrium.
(ii) For every player $i \in [n]$, if $g_i \in \mathcal{G}_i^{\vec{f}}$ then $C_i^{(\vec{f}_{-i}, g_i)}(\vec{h}_0) = \min_{f_i'} C_i^{(\vec{f}_{-i}, f_i')}(\vec{h}_0) = C_i^{\vec{f}}(\vec{h}_0)$.

Proof. To show that $\vec{f}$ being an equilibrium is a sufficient condition, we use the same argument as in Lemma 4 of [43]. In particular, for a player i, due to the Tower Property we have,

$$\begin{aligned} C_i^{\vec{f}}(\vec{h}_0) &= \mathbb{E}[T_i|\vec{h}_{i,0}, \vec{f}] \\ &= \sum_{\vec{h}_{i,\tau^*}} \mathbb{E}[T_i|\vec{h}_{i,0}, (\vec{f}_{-i}, g_i(h_{i,\tau^*}))]\Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\}. \end{aligned} \tag{4.2}$$

For short, we will denote $g_i(h_{i,\tau^*})$ by g_i, thus we denote $\mathbb{E}[T_i|\vec{h}_{i,0}, (\vec{f}_{-i}, g_i(h_{i,\tau^*}))]$ by $C_i^{(\vec{f}_{-i},g_i)}(\vec{h}_0)$. Then, suppose that $\vec{f}$ is an equilibrium and assume for the sake of contradiction that there is a transmission history $\vec{h}_{i,\tau*}$ for player i such that $C_i^{(\vec{f}_{-i},g_i)}(\vec{h}_0) \neq C_i^{\vec{f}}(\vec{h}_0)$. Obviously, if $C_i^{(\vec{f}_{-i},g_i)}(\vec{h}_0) < C_i^{\vec{f}}(\vec{h}_0)$ this would mean that protocol $g_i(\tau^*)$ is better than f_i, thus $\vec{f}$ is not an equilibrium. If, on the other hand, $C_i^{(\vec{f}_{-i},g_i)}(\vec{h}_0) > C_i^{\vec{f}}(\vec{h}_0)$, then from (4.2) there must exist another transmission history $\vec{h}'_{i,\tau^*}$ such that $C_i^{(\vec{f}_{-i},g_i(\vec{h}'_{i,\tau^*}))}(\vec{h}_0) < C_i^{\vec{f}}(\vec{h}_0)$. Therefore, we conclude that $C_i^{\vec{f}}(\vec{h}_0) = C_i^{(\vec{f}_{-i},g_i)}(\vec{h}_0)$ which also equals $\min_{f'_i} C_i^{(\vec{f}_{-i},f'_i)}(\vec{h}_0)$ by definition of the equilibrium, for every transmission history $\vec{h}_{i,\tau^*}$ that is consistent with $\vec{f}$.

To show that $\vec{f}$ being an equilibrium is also a necessary condition, assume that $g_i \in \mathcal{G}_i^{\vec{f}}$ implies $C_i^{(\vec{f}_{-i},g_i)}(\vec{h}_0) = \min_{f'_i} C_i^{(\vec{f}_{-i},f'_i)}(\vec{h}_0)$. Then, equality (4.2) becomes

$$\begin{aligned} C_i^{\vec{f}}(\vec{h}_0) &= \sum_{\vec{h}_{i,\tau^*}} C_i^{(\vec{f}_{-i},g_i(h_{i,\tau^*}))}(\vec{h}_0)\Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\} \\ &= \sum_{\vec{h}_{i,\tau^*}} \min_{f'_i} C_i^{(\vec{f}_{-i},f'_i)}(\vec{h}_0)\Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\} \\ &= \min_{f'_i} C_i^{(\vec{f}_{-i},f'_i)}(\vec{h}_0) \end{aligned}$$

and thus $\vec{f}$ is by definition an equilibrium. □

Corollary 5 (Best response)**.** *Consider a profile $\vec{f} = (f_1, f_2, \dots f_n)$ of acknowledgement-based protocols. For a fixed protocol f'_i of player $i \in [n]$ and some $h_{i,\tau^*} = (a_{i,1}, a_{i,2}, \dots, a_{i,\tau^*})$*

consistent with $(\vec{f}_{-i}, f'_i)$*, define the following protocol.*

$$r_i = r_i(h_{i,\tau^*}) \triangleq \begin{cases} (Pr\{X_{i,t} = a_{i,t}\} = 1, \quad Pr\{X_{i,t} \neq a_{i,t}\} = 0) & \text{, for } 1 \leq t \leq \tau^* \\ f'_{i,t}, & \text{, for } t > \tau^*. \end{cases} \tag{4.3}$$

If for player i *there exists a finite* $\tau^* \geq 1$ *such that* $C_i^{(\vec{f}_{-i}, r_i(h_{i,\tau^*}))}(\vec{h}_0) \geq C_i^{(\vec{f}_{-i}, f_i)}(\vec{h}_0)$ *for every* h_{i,τ^*}*, then* $C_i^{(\vec{f}_{-i}, f'_i)}(\vec{h}_0) \geq C_i^{(\vec{f}_{-i}, f_i)}(\vec{h}_0)$.

Proof. By definition of the expected latency (equation (4.2)) for a fixed τ^* we have:

$$\begin{aligned} C_i^{(\vec{f}_{-i}, f'_i)}(\vec{h}_0) &= \sum_{\vec{h}_{i,\tau^*}} C_i^{(\vec{f}_{-i}, r_i(h_{i,\tau^*}))}(\vec{h}_0) \Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\} \\ &\geq \sum_{\vec{h}_{i,\tau^*}} C_i^{(\vec{f}_{-i}, f_i)}(\vec{h}_0) \Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\} \\ &= C_i^{(\vec{f}_{-i}, f_i)}(\vec{h}_0). \end{aligned}$$

□

Lemma 4 (Equilibrium characterization 2)**.** *Consider a profile* $\vec{f} = (f_1, f_2, \ldots f_n)$ *of acknowledgement-based protocols. The following statements are equivalent:*
(i) $\vec{f}$ *is an equilibrium.*
(ii) For every player $i \in [n]$,

$$\begin{cases} \text{(a)} \quad C_i^{(\vec{f}_{-i}, g_i)}(\vec{h}_0) = C_i^{(\vec{f}_{-i}, r_i)}(\vec{h}_0) = C_i^{\vec{f}}(\vec{h}_0), \quad \forall g_i, r_i \in \mathcal{G}_i^{\vec{f}}, \text{ and} \\ \text{(b)} \quad C_i^{(\vec{f}_{-i}, g_i)}(\vec{h}_0) \leq C_i^{(\vec{f}_{-i}, r_i)}(\vec{h}_0), \quad \forall g_i \in \mathcal{G}_i^{\vec{f}}, r_i \notin \mathcal{G}_i^{\vec{f}}. \end{cases}$$

Proof. Sufficiency of $\vec{f}$ being an equilibrium for condition (ii-a) comes directly from Lemma 3; for condition (ii-b), for the sake of contradiction suppose $\vec{f}$ is an equilibrium and that there exist some protocols $g_i \in \mathcal{G}_i^{\vec{f}}$ and $r_i \notin \mathcal{G}_i^{\vec{f}}$ such that $C_i^{(\vec{f}_{-i}, g_i)}(\vec{h}_0) > C_i^{(\vec{f}_{-i}, r_i)}(\vec{h}_0)$. This means that r_i is a better protocol than f_i, thus $(\vec{f}_{-i}, f_i)$ is not an equilibrium, which is a contradiction.

To prove necessity of $\vec{f}$ being an equilibrium under conditions (ii-a) and (ii-b), for the sake of contradiction, suppose (ii-a) and (ii-b) hold and $\vec{f}$ is not an equilibrium. Then there must exist some protocol f'_i such that $C_i^{(\vec{f}_{-i}, f'_i)}(\vec{h}_0) < C_i^{\vec{f}}(\vec{h}_0)$. Using (4.2) the

latter inequality can be written as

$$\sum_{\vec{h}_{i,\tau^*}} C_i^{(\vec{f}_{-i}, r_i(h_{i,\tau^*}))}(\vec{h}_0)\Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\} < \sum_{\vec{h}_{i,\tau^*}} C_i^{(\vec{f}_{-i}, g_i(h_{i,\tau^*}))}(\vec{h}_0)\Pr\{\vec{h}_{i,\tau^*} \text{ happens for } i\},$$

where $g_i(h_{i,\tau^*})$ is consistent with $\vec{f}$ and $r_i(h_{i,\tau^*})$ is consistent with $(\vec{f}_{-i}, f_i')$. Given the conditions (ii-a) and (ii-b) the latter inequality is a contradiction. □

4.2.2 Acknowledgment-based FIN-EQ protocols

Regarding the search for FIN-EQ protocols, there is no straight-forward way for our equilibrium characterizations, i.e. Lemma 3 and Lemma 4, to be used in order to *find* an equilibrium protocol. However, they allow us to *check* whether the protocols discussed in this section are equilibrium protocols. In this section we give FIN-EQ protocols for $k = 2$ and $k = 3$.

We define the following anonymous, memoryless protocol for $k \geq 2$ channels.

Protocol $f^{\mathbf{k}}$:
For player i, every $t \geq 1$ and any history $\vec{h}_{i,t-1}$,

$$f_{i,t}^k = \left(\Pr\{X_{i,t} = 0\} = 0, \quad \Pr\{X_{i,t} = a\} = \frac{1}{k}, \quad \forall a \in K\right) \tag{4.4}$$

4.2.2.1 n players - 2 transmission channels

Here, we first give an example of a method for checking equilibria (Theorem 5). Then, with a better approach, by employing our equilibrium characterizations (Lemma 3 and Lemma 4), we prove that f^2 is an equilibrium protocol for $n \in \{2, 3, 4\}$ players and $k = 2$ channels (Theorem 7).

Lemma 5. *When all $n \geq 2$ players use protocol f^2 the expected latency of any player is $2^n/n$.*

Proof. The process from the perspective of an arbitrary player i can be modelled as the following Markov chain; the states are named after the number of remaining players including i, and state $\langle\times\rangle$ is the state where i finds herself after successful transmission.

We write p_x^y to denote the transition probability to go from state $\langle x \rangle$ to state $\langle y \rangle$. We have

$$\left.\begin{array}{l} p_m^{\times} = \left(\frac{1}{2}\right)^{m-1} \\ p_m^{m-1} = (m-1)\left(\frac{1}{2}\right)^{m-1} \\ p_m^{m} = 1 - m\left(\frac{1}{2}\right)^{m-1} \end{array}\right\} \forall 3 \leq m \leq n \text{ , and} \tag{4.5}$$

$$p_2^{\times} = \frac{1}{2}, \quad p_2^2 = \frac{1}{2}. \tag{4.6}$$

The expected absorption time from state $\langle n \rangle$ to state $\langle \times \rangle$ is found from the following set of equations:

$$\begin{aligned} & h_m^{\times} = 1 + p_m^m h_m^{\times} + p_m^{m-1} h_{m-1}^{\times}, \quad \text{for all } 3 \leq m \leq n, \\ \text{and} \quad & h_2^{\times} = 1 + p_2^2 h_2^{\times}, \end{aligned}$$

where h_x^y denotes the expected hitting time from state $\langle x \rangle$ to state $\langle y \rangle$. By solving this system of linear equations we get

$$h_n^{\times} = \frac{2^n}{n}, \quad \text{for } n \geq 2.$$

□

In the next theorem we will give an example of a method for checking whether a given protocol profile is an equilibrium, which however could be inconclusive in some cases. Suppose we want to check whether an arbitrary protocol profile $\vec{f}$ is an equilibrium. By definition of the equilibrium, we can fix all protocols except player i's, i.e. $\vec{f}_{-i}$ and check if f_i is a best response to them, and repeat this for every player i. By fixing $\vec{f}_{-i}$ we create a stochastic environment for player i who can be considered to be free to take sequential decisions through time. These decisions correspond to decision rules of f_i. Since, due to the feedback limitations, i has no information about the number of pending players, this situation from her point of view is modeled as an infinite state Partially Observable Markov Decision Process (POMDP). f_i is a best response to $\vec{f}_{-i}$ if and only if f_i is an optimal policy of the POMDP, that is, a set of decisions through time that minimize her expected latency.

However for this kind of POMDPs there are no known techniques to find an optimal

policy. In order to circumvent this problem, we can assume that player i is an advantageous player that always knows how many players are pending. This turns the infinite state POMDP into a finite state Markov Decision Process (MDP), whose optimal policy we can find through known techniques (e.g. [116]). One can see that the optimal policy in the MDP of the advantageous player i yields at most the expected latency of the optimal policy in the POMDP of the initial player i. Thus, if the best policy in the MDP yields the same expected latency as what $\vec{f}$ gives to i, then we know that f_i is a best response; however, if the best policy of the MDP yields smaller expected latency, then we get no information about whether f_i is a best response in the POMDP or not. The proof of the next theorem demonstrates the method and shows that protocol f^2 of (4.4) is an equilibrium protocol for 3 players.

Theorem 5. *For 3 players and 2 channels, f^2 is an equilibrium protocol with expected latency* $8/3$.

Proof. Consider the Markov Decision Process (MDP) $(T, S_t, A_{s,t}, p_t(j|s,a), r_t(s,a))$, where S_t is the state space for time t; $A_{s,t}$ is the set of possible actions that can be taken after observing state s at time t; $p_t(j|s,a)$ defines the transition probability to state $j \in S_{t+1}$ at time $t+1$, and only depends on the state s and chosen action a at time t; $r_t(s,a)$ is the cost function that determines the immediate cost for the agent's choice of action a while in state s. When the state s cannot be observed with certainty at time t, the agent only knows a probability distribution, called *belief state*, over S_t. The process then is called Partially Observable Markov Decision Process (POMDP). An *optimal policy* $\pi : S \to A$ is a function that rules, for each state or belief state, which action to perform, with an objective to minimize the expected cost.

For the proof of the above theorem we will use the following property of POMDPs. This property comes directly from the fact that an agent optimizing over all policies that every time consider her exact state gets a better policy than an agent that knows a probability distribution on the state space (belief states).

Proposition 1. *An optimal policy π_1 of an agent in a POMDP yields as expected cost at least the expected cost of the optimal policy π_2 of the corresponding MDP, in which at any time t the agent observes her exact state.*

To prove Theorem 5 we think as follows. Let us fix protocol f^2 as defined in (4.4) for two players, and let the remaining player i have an arbitrary protocol g_i. Then let us find

the optimal policy for i. If and only if the optimal policy yields expected cost strictly lower than what protocol f^2 would yield for player i (due to Lemma 5, that is 8/3), then f^2 is not an equilibrium protocol. The game stated at Theorem 5, from player i's perspective, is modelled by a POMDP where each state is determined by the number of pending players, with an additional absorbing state - where i goes after successfully transmitting - and i's transmission history for every $t \geq 1$. Player i's belief state at any time t is determined by her belief state at time $t-1$, the action she chose at time $t-1$, and her observation (e.g. her transmission history up to $t-1$). This is a POMDP with infinite states, for which, to the best of our knowledge, currently there are no methods in the literature for finding an optimal policy.

However, we will find the best policy and the expected cost of the corresponding MDP, where player i knows in what state she finds herself after an action and observation. This expected cost is a lower bound on the expected cost of the optimal policy of the original POMDP (see Proposition 1). In the MDP we create, player i knows at any time t how many players are pending and her transmission history up to time t.

Let $p \in \{1,2,3\}$ indicate the number of pending players. Observe that the time steps at which the process has a given p are consecutive; without loss of generality assume that for some p, the process is in the discrete time interval $[\tau_p, \tau_{p-1}-1]$, where we set $\tau_3 = 1$. Consider now the set S_p of all states $s_p(\vec{h}_{i,t})$ of the MDP, where the number of pending players $p \in \{1,2,3\}$ is fixed, whereas the transmission history $\vec{h}_{i,t}$ for $\tau_p \leq t < \tau_{p-1}$ can vary. Because of the protocol f being memoryless, the same action (probability distribution over action space A) of i chosen at any state in S_p produces the same transition probabilities. Therefore, choosing the optimal policy makes the set S_p of states collapse to a single state s_p, where $p \in \{1,2,3\}$. The resulting MDP is a finite MDP with states s_1, s_2, s_3 and $s_\times$, where the latter is an absorption state to which player i goes after a successful transmission. Denote the expected cost of the MDP's optimal policy given that the initial state is s_p by $c(s_p)$. In our problem the immediate cost for any combination of state and action is 1, since we count the number of rounds in which i is pending. Using Lemma 5.4.2 and Theorem 5.4.3 of [116] we can find $c(s_3)$ by solving the following system of linear equations

$$c(s_p) = 1 + \sum_{s' \in \{s_1, s_2, s_3\}} \Pr(s_p \text{ to } s' | \text{ policy } \pi) c(s'). \tag{4.7}$$

Then, by minimizing each $c(s_p)$ over policies π we get the optimal expected costs $C(s_p)$,

$p \in \{1,2,3\}$. As a byproduct of the minimization we find the best policy π^*.

In our problem, a policy π is a tuple $(q_1, z_1, q_2, z_2, q_3, z_3)$, where q_p, $p \in \{1,2,3\}$ determines the probability that player i will attempt a transmission, and z_p, $p \in \{1,2,3\}$ determines the probability that she will attempt the transmission on channel $a = 1$. To give a small example, for a given state s_p, $(\Pr(X_t = 0), \Pr(X_t = 1), \Pr(X_t = 2)) = (1 - q_p, q_p z_p, q_p(1 - z_p))$. By solving system (4.7), we get that

$$c(s_1) = \frac{1}{q_1}, \quad c(s_2) = \frac{2 + 2q_1 - 2q_2}{2q_1 - q_1 q_2}, \quad c(s_3) = 2 + \frac{4 - 2q_2 - 2q_2 q_3 + 2q_1 q_2 q_3}{4q_1 - 2q_1 q_2 + 2q_1 q_3 - q_1 q_2 q_3}$$

which implies that a policy does not depend on any of the z_p's. Now, by minimizing the above expected costs we get $C(s_1) = 1, C(s_2) = 2$ and $C(s_3) = 8/3$ for $q_1 = 1$ and $q_3 = 1$. Note that the optimal policy allows z_1, z_2, z_3 and q_2 to be arbitrary probabilities. q_2 being even 0 is not a contradiction since in our MDP the player is always aware of the pending players (state); in the case where $q_2 = 0$, when the player is in state s_2, she waits one round until the other player transmits successfully and then realizes that she is alone pending in s_1; in the next round she transmits with probability 1.

We have shown that a best policy of an advantageous player gives her the same expected latency as protocol f^2 defined in (4.4) (the expected latency of f^2 is given by Lemma 5). This, combined with Proposition 1 completes the proof of Theorem 5. □

We subsequently exploit the lack of memory and the anonymity of our protocol f^2 defined in equation (4.4) and show more general results on equilibria (Theorem 7), using the characterizations of Lemma 3 and Lemma 4.

Theorem 6. *In a contention game with $k = 2$ channels, consider an anonymous, memoryless protocol of player i with the property: $Pr\{X_{i,t} = 0\} = 0$, for every $t \geq 1$. For more than 4 players any such protocol is not an equilibrium protocol.*

Proof. Assume that an anonymous protocol f as stated in the theorem is an equilibrium protocol for $n \geq 5$ players. We will show that condition (ii-b) of Lemma 4 does not hold. That is, if $n \geq 5$ players use a protocol f with the property that in each time its decision rule assigns zero probability to "no transmission", then there exists a best response that yields strictly better expected latency for an arbitrary player.

Suppose f is an equilibrium protocol. f consists of a decision rule for each time slot t, i.e. a probability distribution on the available channels (with probability 0 of

"no transmission" as the theorem's statement requires). Since all players use this protocol, in an arbitrary time t all players have the same distribution on the channels. For the sake of contradiction, suppose there is some t' for which the decision rule is other than $\left(\Pr\{X_{i,t}=1\}=\frac{1}{2}, \quad \Pr\{X_{i,t}=2\}=\frac{1}{2}\right)$. Without loss of generality, we have $\Pr\{X_{i,t}=1\} > \Pr\{X_{i,t}=2\}$. Thus, an arbitrary player i, at time t, can unilaterally change her distribution to $(\Pr\{X_{i,t}=1\}=0, \quad \Pr\{X_{i,t}=2\}=1)$ and increase her probability of transmitting successfully in the specific round. As a consequence her expected latency would strictly decrease, hence a protocol with a decision rule with different probabilities on each channel cannot be in a symmetric equilibrium. Therefore, the anonymous, equilibrium protocol f, with the property $\Pr\{X_{i,t}=0\}=0$ for every $t \geq 1$, prescribes $\left(\Pr\{X_{i,t}=1\}=\frac{1}{2}, \quad \Pr\{X_{i,t}=2\}=\frac{1}{2}\right)$ for every $t \geq 1$. The expected latency of a player using such a protocol, when there are n pending players, is found in Lemma 5 to be $2^n/n$.

We will show that, when the number of pending players at $t=0$ is $n \geq 5$, protocol

$$g_i \triangleq \begin{cases} (\Pr\{X_{i,1}=1\}=0, \quad \Pr\{X_{i,1}=2\}=0) \\ \left(\Pr\{X_{i,t}=1\}=\frac{1}{2}, \quad \Pr\{X_{i,t}=2\}=\frac{1}{2}\right), \quad \text{for } t \geq 2, \end{cases}$$

is a better response for an arbitrary player i, that is, $C_i^{(f_{-i},g_i)}(\vec{h}_{i,0}) < C_i^f(\vec{h}_{i,0}) = 2^n/n$.

Suppose player i uses protocol g_i when there are $n \geq 5$ pending players at $t=0$. At time $t=2$ she is not aware of the number of players that remain pending. However, there are two cases, either n players are pending in case none of the other $n-1$ players in $t=1$ transmitted successfully, or $n-1$ players remain in case only one of the other $n-1$ players transmitted successfully in $t=1$. Note that there is no way that two players cannot simultaneously transmit successfully in round $t=2$ due to the given protocol f and the number of pending players. The probability for each of the two aforementioned events is,

$$P_{n-1}(x) = \sum_{r=x}^{n-1} (-1)^{r-x} \binom{r}{x} \binom{2}{r} \binom{n-1}{r} r! \left(\frac{1}{2}\right)^r \left(1-\frac{r}{2}\right)^{n-1-r}$$

where x is the number of players that transmit successfully, $0^0 \triangleq 1$, and $\binom{a}{b} \triangleq 0$ for $a < b$. To see how this formula is produced, please refer to the proof of Lemma 9 (Section 4.4), up to equation (4.19). Here, equation (4.19) is used for $z=1$ and $k=2$.

In order to capture the dependence of the expected future latency (after history h_{t-1}) on the number of pending players n, when player i uses g_i and the rest of the players

use f, we denote it by $F_{i,n}^{(\vec{f}_{-i},g_i)}(\vec{h}_{i,t-1})$. Similarly, we denote the expected latency by $C_{i,n}^{(\vec{f}_{-i},g_i)}(\vec{h}_{i,t-1})$. We have,

$$\begin{aligned} C_{i,n}^{(f_{-i},g_i)}(\vec{h}_{i,0}) = F_{i,n}^{(f_{-i},g_i)}(\vec{h}_{i,0}) &= 1 + P_{n-1}(0)F_{i,n}^{(f_{-i},g_i)}(\vec{h}_{i,1}) + P_{n-1}(1)F_{i,n-1}^{(f_{-i},g_i)}(\vec{h}_{i,1}) \\ &= 1 + P_{n-1}(0)\frac{2^n}{n} + P_{n-1}(1)\frac{2^{n-1}}{n-1}. \end{aligned} \tag{4.8}$$

For $n \geq 5$, our formula in (4.19) gives $P_{n-1}(0) = 1 - (n-1)\left(\frac{1}{2}\right)^{n-2}$ and $P_{n-1}(1) = (n-1)\left(\frac{1}{2}\right)^{n-2}$. Therefore (4.8) becomes

$$\begin{aligned} C_{i,n}^{(f_{-i},g_i)}(\vec{h}_{i,0}) &= 1 + \left[1 - (n-1)\left(\frac{1}{2}\right)^{n-2}\right]\frac{2^n}{n} + (n-1)\left(\frac{1}{2}\right)^{n-2}\frac{2^{n-1}}{n-1} \\ &= \frac{2^n}{n} + \frac{4}{n} - 1 \\ &< \frac{2^n}{n} \quad , \text{ for } n > 4 \\ &= C_{i,n}^{f}(\vec{h}_{i,0}). \end{aligned}$$

Thus protocol g_i yields strictly smaller expected latency than f_i for player i when $n \geq 5$, and this means that f is not a symmetric equilibrium for $n \geq 5$. □

Since protocol f^2 belongs to the class of protocols defined in the statement of Theorem 6, the following corollary is immediate.

Corollary 6. *For $n \geq 5$ players and $k = 2$ channels, f^2 is not an equilibrium protocol. In fact, a better response for any player is to not transmit in $t = 1$ and then follow f^2.*

Now we prove two lemmata that, combined with our second characterization of equilibria (Lemma 4), result in one of this section's main theorems (Theorem 7) that determines equilibrium protocols for $n \in \{2, 3, 4\}$ players and $k = 2$ channels. In particular, we will show that for number of players $n = 2$, $n = 3$ and $n = 4$, when $n - 1$ players use f, if some deviator unilaterally chooses any possible protocol g_i as defined in (4.1) that *is* consistent with $\vec{f}$, she will suffer the same expected latency, namely $2^n/n$. Then, we will show that if the deviator unilaterally chooses any possible protocol as defined in (4.1) that *is not* consistent with $\vec{f}$, she will suffer expected latency at least $2^n/n$. These two facts, by Lemma 4, show that f is an equilibrium protocol for $n \in \{2, 3, 4\}$.

Lemma 6. *For $n \geq 2$ players and $k = 2$ channels, any player i that follows a protocol $g_i \in \mathcal{G}^{f^2}$ in the profile (f^2_{-i}, g_i), where f^2 is defined in (4.4), has expected latency $2^n/n$.*

Proof. Consider the contention game with fixed number of players $n \geq 2$ and 2 channels. $n-1$ players use protocol f^2 and a player $i \in [n]$ uses some protocol $g_i(h_{i,\tau^*}) \in \mathcal{G}^{f^2}$ as defined in (4.1), for some $\tau^* \geq 1$. To make easier our reference to the expected future latency of a player in the special case where (almost) all players follow protocol f^2 of (4.4), and to capture the number of players in the notation, we will denote by $D(r_i, n) \triangleq \mathbb{E}[T_i|\overrightarrow{h}_{i,0}, (\overrightarrow{f^2}_{-i}, r_i)]$ and $D(f^2_i, n) \triangleq \mathbb{E}[T_i|\overrightarrow{h}_{i,0}, \overrightarrow{f^2}]$ the expected future latency of player i when n players participate.

First we show that condition (ii-a) of Lemma 4 holds for every $n \geq 2$. From Lemma 5 we know that $D(f^2_i, n) = 2^n/n$, for every $i \in [n]$. Now observe that the set of all protocols $g_i(\tau^*)$ as defined in 4.1 that are consistent with f^2_i, consists of the protocols for which $a_t \neq 0$ for every $1 \leq t \leq \tau^*$ for any $\tau^* \geq 1$. That is, for all possible tuples $(a_1, a_2, \ldots, a_{\tau^*})$ of a given τ^*, there is no $t \leq \tau^*$ for which $a_t = 0$, and this is for all $\tau^* \geq 1$, since a history with "no transmission attempt" in it is not consistent with f^2. Given a tuple $h_{i,\tau^*} = (a_1, a_2, \ldots, a_{\tau^*})$, denote by x_t the indicator variable that equals 1 if player i chooses channel 1, and 0 if she chooses channel 2 in round $t \leq \tau^*$. Formally, a protocol as described above is

$$g_i = g_i(h_{i,\tau^*}) \triangleq \begin{cases} (\Pr\{X_{i,t} = 1\} = x_t, \quad \Pr\{X_{i,t} = 2\} = 1 - x_t) & \text{, for } 1 \leq t \leq \tau^* \\ f^2_{i,t} \quad \text{, for } t > \tau^*, \end{cases}$$

This process where a single player i uses some protocol g_i and has a latency according to g_i and the other players' fixed protocols, can be modelled as a Partially Observable Markov Decision Process (POMDP) with infinite states; in this POMDP, each state is determined by the transmission history of player i and the number of pending players including i, with an additional absorbing state where i goes after successfully transmitting; player i's belief state at any time t is determined by her belief state at time $t-1$, the action she chose at time $t-1$, and her observation (e.g. her transmission history up to $t-1$).

The fact that we consider acknowledgement-based protocols together with the fact that the partial protocol profile f^2_{-i} which produces our POMDP consists of memoryless and time-independent protocols, make the states of our POMDP be independent of player i's history. We now remark that, regardless of the action taken in some belief state from player i playing g_i, the transition probabilities between belief states are independent of time. In

particular, denote by $\langle m, t\rangle$ a state with m pending players including player i at time $t \geq 1$, and by $\langle \times \rangle$ the unique absorption state where i finds herself after successful transmission. We write p_x^y to denote the transition probability to go from state $\langle x\rangle$ to state $\langle y\rangle$. It is easy to see that the transition probabilities among belief states with $1 \leq t \leq \tau^*$ are

$$\left.\begin{array}{l} p_{m,t}^{\times} = \left(\frac{1}{2}\right)^{m-1} \\ p_{m,t}^{m-1,t+1} = (m-1)\left(\frac{1}{2}\right)^{m-1} \\ p_{m,t}^{m,t+1} = 1 - m\left(\frac{1}{2}\right)^{m-1} \end{array}\right\} \forall 3 \leq m \leq n, \quad 1 \leq t \leq \tau^*,$$

$$\text{and} \quad p_{2,t}^{\times} = \frac{1}{2}, \quad p_{2,t}^{2,t+1} = \frac{1}{2}.$$

Observe that the above transition probabilities of any state for which $1 \leq t \leq \tau^*$ are identical to those of equations (4.5) and (4.6) in the proof of Lemma 5; obviously for $t > \tau^*$ the same holds because player i has switched back to protocol f^2. Since player i's actions do not affect the transition probabilities of the resulting belief states, the above POMDP reduces to a Markov chain that is in fact identical to the one defined in the proof of Lemma 5, thus $D(g_i, n) = D(f_i^2, n) = 2^n/n$.

The natural explanation for our POMDP resulting to the above Markov chain is that, if for a given round all players have a given probability of transmission (not necessarily 1) uniformly distributed on the channels and a single deviator picks an arbitrary distribution on the channels for the same probability of transmission (in this case 1), then: (a) the probability with which she transmits successfully remains unchanged because each channel is blocked with equal probability $(1 - 1/2^{n-1})$ by the rest of the players, and (b) the probabilities with which a specific number s of players (excluding i) transmit successfully remain unchanged because, the probability of s players successfully transmitting conditional on i choosing any of the channels is the same (due to the uniform distributions on the channels by the rest of the players) regardless of the channel chosen by i.

Remark: The above arguments hold also in the case of any number $k \geq 1$ of channels when an anonymous, memoryless protocol f is used by all players except i, where f is defined by a probability $0 < z \leq 1$ that is split uniformly on the channels in every time-step (in our proof, $k = 2$ and $z = 1$ for all $t > 0$). In such a case the POMDP is reduced to a corresponding Markov chain that is produced when all players follow f. □

Lemma 7. *For $2 \leq n \leq 4$ players and $k = 2$ channels, any player i that follows protocol*

$r_i \notin \mathcal{G}^{f^2}$ in the profile (f^2_{-i}, r_i), where f^2 is defined in (4.4), has expected latency at least $2^n/n$.

Proof. Consider the contention game with fixed number of players $n \in \{2,3,4\}$ and 2 channels. $n-1$ players use protocol f^2 and a player $i \in [n]$ uses some protocol $r_i = r_i(h_{i,\tau^*}) \notin \mathcal{G}^{f^2}$ as defined in (4.1), for some $\tau^* \geq 1$. It is sufficient to show that the lemma holds, when r_i is a best response to f^2_{-i}, where r_i is constrained to be inconsistent with (f^2_{-i}, f^2_i). Therefore, among such best responses r_i there has to be one with a round $t < \infty$ for which $\Pr\{X_{i,t} = 0\} > 0$ by definition of inconsistency. Let us focus on the smallest such t which we will call from now on t_0, i.e. $t_0 \triangleq \inf\{t : \Pr\{X_{i,t} = 0\} > 0\}$. Let us now define the set of protocols $r_i(h_{i,t_0}) \notin \mathcal{G}^{f^2}$ for the aforementioned t_0. There are two categories of such protocols: Category (1) has $a_{t_0} \neq 0$, and Category (2) has $a_{t_0} = 0$. Each of those categories is partitioned in two other categories: Category (I) has $\Pr\{X_{i,t} = 0\} = 0$ for every $t > t_0$, and Category (II) has $\Pr\{X_{i,t} = 0\} > 0$ for some $t > t_0$. The categories are presented in Table 4.1 below.

Category 1	$a_{t_0} \neq 0$
Category 2	$a_{t_0} = 0$

Category I	$\forall t > t_0$: $\Pr\{X_{i,t} = 0\} = 0$
Category II	$\exists t > t_0$: $\Pr\{X_{i,t} = 0\} > 0$

Table 4.1: The categories of protocol $r_i(h_{i,t_0})$.

Right before time t_0 there are n possible cases that could have occurred: m players are pending including player i, for $1 \leq m \leq n$. In each of those cases we want to find the expected future latency of a player i that unilaterally uses protocol $r_i(h_{i,t_0})$, given history $\vec{h}_{i,t_0-1}$, and given that the pending players right before time t_0 are m; we will denote this by $F^{(\vec{f^2}_{-i}, r_i)}_{i,m}(\vec{h}_{i,t_0-1})$. We will prove our claim step by step, starting from protocols of Category (I) which are easier to analyze, and move on to protocols of Category (II); we start the analysis from the case with the least possible players and build up to the required number of players.

Starting with Category (1-I), the analysis of the proof of Lemma 6 implies that these protocols r_i yield the same expected latency as f^2_i in the tuple f^2, since their process' Markov chain is identical to this of the case (f^2_{-i}, f^2_i). For Category (2-I), player i does not transmit at t_0. Given that right before t_0 there are m pending players including i, at t_0 either all m players remain pending, or $m-1$, or $m-2$; the first event occurs when none of the $m-1$ players using protocol f^2 at t_0 transmitted successfully, the second when only

one of them did, and the third when two of them did. The probability for each of those events is $P_{m-1}(x)$, where x is the number of players that transmit successfully, and can be found in (4.19) for $k = 2$ and $z = 1$. Therefore we have,

$$\begin{aligned} F_{i,m}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) &= 1 + P_{m-1}(0)F_{i,m}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0}) + P_{m-1}(1)F_{i,m-1}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0}) \\ &\quad + P_{m-1}(2)F_{i,m-2}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0}) \\ &= 1 + P_{m-1}(0)D(f_i^2,m) + P_{m-1}(1)D(f_i^2,m-1) + P_{m-1}(2)D(f_i^2,m-2), \end{aligned} \tag{4.9}$$

where f_i' is the protocol followed by i for $t > t_0$. For $m = 1$ it is $P_0(0) = 1$, and $P_0(1) = P_0(2) = 0$. For $m = 2$ it is $P_1(0) = P_1(2) = 0$, and $P_1(1) = 1$. For $m = 3$ it is $P_2(0) = P_2(2) = \frac{1}{2}$, and $P_2(1) = 0$. For $m = 4$ it is $P_3(0) = 1 - 3\left(\frac{1}{2}\right)^2$, $P_3(1) = 3\left(\frac{1}{2}\right)^2$, and $P_3(2) = 0$.

Now, using (4.9), we can see that for $1 \leq m \leq 4$ it is

$$F_{i,m}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq F_{i,m}^{(\vec{f^2}_{-i},f_i)}(\vec{h}_{i,t_0-1}) = D(f_i^2,m) = 2^m/m.$$

In particular,

$$\begin{aligned} &\text{for } m = 1: \quad 2 \geq 1, \\ &\text{for } m = 2: \quad 2 \geq 2, \\ &\text{for } m = 3: \quad \frac{17}{6} \geq \frac{8}{3}, \text{ and} \\ &\text{for } m = 4: \quad 4 \geq 4. \end{aligned}$$

Equivalently, $C_{i,m}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq C_{i,m}^{(\vec{f^2}_{-i},f_i)}(\vec{h}_{i,t_0-1})$, and therefore, due to (4.2), it is

$$C_{i,m}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,0}) \geq C_{i,m}^{(\vec{f^2}_{-i},f_i)}(\vec{h}_{i,0}), \quad \text{for } 1 \leq m \leq 4.$$

Thus, for Category (I) and $2 \leq n \leq 4$, condition (ii-b) of Lemma 4 holds.

For Category (1-II), we prove our claim for $1 \leq m \leq 4$ pending players right before t_0. For $m = 1$, obviously $F_{i,1}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = 1$, which is also the minimum possible when

only one player is pending. For $m = 2$, we have

$$F_{i,2}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = 1 + \Pr\{\text{No player transmits successfully}\}F_{i,2}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0})$$

Now, given that the protocol f^2 used by all players apart from i is time-independent, it should be $F_{i,2}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = F_{i,2}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0})$. Because if $F_{i,2}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) < F_{i,2}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0})$ or $F_{i,2}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) > F_{i,2}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0})$, then f_i' is not a best response; in the former situation player i would prefer $r_i(h_{i,t_0})$ over f_i'; in the latter situation she would prefer a modified protocol $r_i(h'_{i,t_0})$ with $\Pr\{X_{i,t_0} \neq 0\} = 0$ over the current $r_i(h_{i,t_0})$, respectively. The probability of no player transmitting successfully in t_0 is $1/2$, thus we get $F_{i,2}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = 2 = F_{i,2}^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,t_0-1})$, which implies $C_{i,2}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = C_{i,2}^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,t_0-1})$.

For $m = 3$, we have

$$\begin{aligned} F_{i,3}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = 1 &+ \Pr\{\text{No player transmits successfully}\}F_{i,3}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0}) \\ &+ \Pr\{\text{Exactly 1 player other than } i \text{ transmits successfully}\}F_{i,2}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0}) \end{aligned} \tag{4.10}$$

From the previous step, we know that a best response to f_{-i} when there are 2 players pending including i yields expected latency to i equal to 2. Also, the probability that exactly one player other than i transmits successfully when there are 3 players pending, is $1/2$. So, (4.10) gives

$$F_{i,3}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq 2 + \Pr\{\text{No player transmits successfully}\}F_{i,3}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0})$$

Again, given that the protocol f used by all players apart from i is time-independent, it should be $F_{i,3}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) = F_{i,3}^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,t_0})$ for the same reasons explained in the case of $m = 2$. The probability of no player transmitting successfully in t_0 is $1/2$, thus we get $F_{i,3}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq 8/3 = F_{i,3}^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,t_0-1})$, which implies $C_{i,3}^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq C_{i,3}^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,t_0-1})$.

Finally, for $m = 4$, we have

$$F_{i,4}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) = 1 + \Pr\{\text{No player transmits successfully}\}F_{i,4}^{(\overrightarrow{f^2}_{-i},f_i')}(\overrightarrow{h}_{i,t_0}) + \Pr\{\text{Exactly 1 player other than } i \text{ transmits successfully}\}F_{i,3}^{(\overrightarrow{f^2}_{-i},f_i')}(\overrightarrow{h}_{i,t_0}) \tag{4.11}$$

From the previous step, we know that a best response to f_{-i}^2 when there are 3 players pending including i yields expected latency to i at least 8/3. Also, the probability that exactly one player other than i transmits successfully when there are 4 players pending, is 3/8. So, (4.11) gives

$$F_{i,4}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) \geq 2 + \Pr\{\text{No player transmits successfully}\}F_{i,4}^{(\overrightarrow{f^2}_{-i},f_i')}(\overrightarrow{h}_{i,t_0})$$

Again, given that the protocol f^2 used by all players apart from i is time-independent, it should be $F_{i,4}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) = F_{i,4}^{(\overrightarrow{f^2}_{-i},f_i')}(\overrightarrow{h}_{i,t_0})$ for the same reasons explained for $m \in \{2,3\}$. The probability of no player transmitting successfully in t_0 is 1/2, thus we get $F_{i,4}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) \geq 4 = F_{i,4}^{(\overrightarrow{f^2}_{-i},f_i^2)}(\overrightarrow{h}_{i,t_0-1})$, which implies

$$C_{i,4}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) \geq C_{i,4}^{(\overrightarrow{f^2}_{-i},f_i^2)}(\overrightarrow{h}_{i,t_0-1}).$$

Thus, for Category (1-II) and $2 \leq n \leq 4$, condition (ii-b) of Lemma 4 holds.

Now we proceed with the proof of the statement for $1 \leq m \leq 4$ for the final category, namely Category (2-II), using the results from Category (1-II). For every $m \geq 1$, equation (4.8) holds. For $m = 1$, we have

$$F_{i,1}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) = 1 + 1 \cdot F_{i,1}^{(\overrightarrow{f^2}_{-i},f_i')}(\overrightarrow{h}_{i,t_0}) \geq 2,$$

where the above inequality comes from the fact that the minimum expected future latency for $m = 1$ is 1 (found in Category (1-II)). By applying the same methodology for $2 \leq m \leq 4$ we have

$$F_{i,m}^{(\overrightarrow{f^2}_{-i},r_i)}(\overrightarrow{h}_{i,t_0-1}) \geq 3 + \left[1 - (m-1)\left(\frac{1}{2}\right)^{m-2}\right]\frac{2^m}{m} \geq \frac{2^m}{m} = F_{i,m}^{(\overrightarrow{f^2}_{-i},f_i^2)}(\overrightarrow{h}_{i,t_0-1}).$$

Then, by taking into account our lower bounds for $F_i^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t})$ when $0 \leq t \leq t_0 - 1$ and for all possible numbers m of remaining players (including i), we get

$$F_i^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq F_i^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,t_0-1}),$$

which implies

$$C_i^{(\vec{f^2}_{-i},r_i)}(\vec{h}_{i,t_0-1}) \geq C_i^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,t_0-1}).$$

Then, from Corollary 5 and equation (4.2) it is $C_i^{(\vec{f^2}_{-i},f_i')}(\vec{h}_{i,0}) \geq C_i^{(\vec{f^2}_{-i},f_i^2)}(\vec{h}_{i,0})$ and this completes the proof. □

Theorem 7. *For $n \in \{2,3,4\}$ players and $k = 2$ channels, f^2 is an equilibrium protocol with expected latencies 2, 8/3 and 4, respectively.*

Proof. By combining Lemma 6, Lemma 7 and the equilibrium characterization of Lemma 4. □

4.2.2.2 n players - 3 transmission channels

Here, by employing our characterizations, namely Lemma 3 and Lemma 4, we give an acknowledgement-based, equilibrium protocol for $n \in \{2,3,4,5\}$ players and $k = 3$ channels.

Theorem 8. *For $n \in \{2,3,4,5\}$ players and $k = 3$ channels, f^3 defined in (4.4) is an equilibrium protocol with expected latencies 3/2, 15/8, 189/80 and 597/200, respectively.*

We omit the proof because the proof idea is the same as that of Theorem 7. However, the analysis here is done for each value of n separately, since we do not have a closed form (similar to that of Lemma 5) for the expected latency of n players using protocol f^3 for 3 channels. This is because, although using standard Markov chain techniques a linear recurrence relation of the expected latency is easily found, this recurrence relation has non-constant coefficients, for which - to our knowledge - there are no techniques in the literature to solve them[1].

[1]We note that reducing the recurrence relation to one with constant coefficients using already existing techniques did not work.

4.3 Equilibria for Ternary Feedback Protocols

In this section we consider anonymous protocols with *ternary feedback*. In this feedback setting, a pending player knows at every time t the number $m \leq n$ of pending players. This knowledge is given to each player regardless of her transmission history. Our analysis is for a special class of protocols, termed *time-dependent* where each player considers (besides the number m from the feedback) only time $t \geq 1$ and not her whole transmission history. Time-dependent protocols constitute a subset of the (most general) class of history-dependent protocols. In this section we deal only with time-dependent protocols and extend the model and results of [69] which studied the single channel setting. We also conjecture that general history-dependent protocols have the same set of equilibria as time-independent protocols; it seems that the ability of a player to remember her history h_{t-1} only contributes in that it helps her estimate the number of pending players m at round t, so that they can adjust their transmission probabilities in order to maximize their probability of success. Therefore in the ternary feedback setting where m is known to the players at every t, we conjecture that the whole history is not needed.

4.3.1 Nash equilibrium characterization

In Theorem 9 we give a characterization of FIN-EQ protocols for $n \geq 1$ players and $k = 2$ channels in the time-dependent case for ternary feedback. The theorem follows from the below analysis.

Suppose $n \geq 2$ players use the same protocol f in a system with 2 available transmission channels. At time t, where $t \geq 1$, the decision rule of player i among m pending players is described by the probabilities with which she will transmit on channel 1 and channel 2, i.e. $p^{i,1}_{m,t}$ and $p^{i,2}_{m,t}$ respectively. Also, since the information about the value of t is common knowledge to the players, $p^{i,1}_{m,t} = p^{j,1}_{m,t}$ and $p^{i,2}_{m,t} = p^{j,2}_{m,t}$ for any two pending players i, j. Therefore we can omit the player indicator superscript from the probabilities and write $p^1_{m,t}$ and $p^2_{m,t}$ respectively.

Now suppose that the anonymous protocol f is an equilibrium and also that $p^1_{m,t} \neq p^2_{m,t}$. Without loss of generality $p^1_{m,t} > p^2_{m,t}$. Then a player could unilaterally deviate at round t to $p^1_{m,t} = 0$, $p^2_{m,t} = 1$, thus maximizing her own probability of success. Therefore, in an anonymous, equilibrium protocol, at any time $t \geq 1$ and every number $m \geq 2$ of pending players, each player assigns equal transmission probabilities to the channels. Hence, we

also drop the channel indicator superscript and write $p_{m,t}$. Note that $p_{m,t} \in [0, \frac{1}{2}]$.

We will slightly abuse the notation here and write $C_{m,t}$ and $F_{m,t}$ instead of $C_m(h_{t-1})$ and $F_m(h_{t-1})$, for the *expected cost* of a player (e.g. Alice) and the *expected future latency* of a player respectively, at time $t \geq 1$, where there are $1 \leq m \leq n$ pending players. Note that, since the protocol is symmetric, we have replaced the subscript that indicates the player's identity with the one that indicates the number of pending players, and we also have omitted the superscript f.

We have[2]

$$C_{m,t} = P_m^{\times} \cdot t + P_m^{m-1} \cdot C_{m-1,t+1} + P_m^{m-2} \cdot C_{m-2,t+1} + P_m^{m} \cdot C_{m,t+1}$$
$$\text{or equivalently,} \quad F_{m,t} = 1 + P_m^{m-1} \cdot F_{m-1,t+1} + P_m^{m-2} \cdot F_{m-2,t+1} + P_m^{m} \cdot F_{m,t+1},$$

where for $m \geq 2$:
$$\begin{aligned}
P_m^{\times} =& \Pr\{\text{Alice transmits successfully}\} \\
=& 2p_{m,t}(1-p_{m,t})^{m-1}, \\
P_m^{m-1} =& \Pr\{\text{Exactly 1 player other than Alice transmits successfully}\} \\
=& 2(m-1)p_{m,t}\left[(1-p_{m,t})^{m-1} - (m-1)p_{m,t}(1-2p_{m,t})^{m-2}\right], \\
P_m^{m-2} =& \Pr\{\text{Exactly 2 players other than Alice transmit successfully}\} \\
=& (m-1)(m-2)p_{m,t}^2(1-2p_{m,t})^{m-2}, \\
P_m^{m} =& \Pr\{\text{No player transmits successfully}\} = 1 - P_m^{\times} - P_m^{m-1} - P_m^{m-2}.
\end{aligned}$$

For $m = 1$ the pending player has probability of no transmission equal to zero, therefore $F_{1,t} = 1$ for every $t \geq 1$.

Now, given that $m \geq 2$ players are pending, the equilibrium protocol cannot assign to them probability $p_{m,t} = 0$ at any time t. That is because a unilateral deviator that surely transmitted to a channel would be successful and therefore she would acquire strictly smaller latency than any other player. Since transmission to both channels is in the support of the decision rule of a player at time t, both sure transmission attempt to some channel and no transmission attempt should yield the same expected latency to a player in a Nash equilibrium. In the sequel we will use the expected future latency at time $t \geq 1$, $F_{m,t}$ for our analysis. The expected future latency of Alice when she surely transmits on an

[2]The probabilities are correct by defining $0^0 = 1$.

arbitrary channel in round t with $m \geq 2$ pending players (including herself) is

$$F_{m,t} = 1 + Q_m^{m-1} \cdot F_{m-1,t+1} + (1 - Q_m^{\times} - Q_m^{m-1}) \cdot F_{m,t+1}, \tag{4.12}$$

$$\begin{aligned}
\text{where for } m \geq 3: \quad Q_m^{\times} =& \Pr\{\text{Alice transmits successfully}\} \\
=& (1-p_{m,t})^{m-1}, \\
Q_m^{m-1} =& \Pr\{\text{Exactly 1 player other than Alice transmits successfully}\} \\
=& (m-1)p_{m,t}\left[(1-p_{m,t})^{m-2} - (1-2p_{m,t})^{m-2}\right], \\
Q_m^{m} =& \Pr\{\text{No player transmits successfully}\} = 1 - Q_m^{\times} - Q_m^{m-1}, \\
\text{for } m = 2: \quad Q_2^{\times} =& 1 - p_{m,t}, \quad Q_2^1 = 0, \quad Q_2^2 = p_{m,t}.
\end{aligned} \tag{4.13}$$

The expected future latency of Alice when she surely does not attempt transmission in round t with $m \geq 2$ pending players (including herself) is

$$F_{m,t} = 1 + S_m^{m-1} \cdot F_{m-1,t+1} + S_m^{m-2} \cdot F_{m-2,t+1} + (1 - S_m^{m-1} - S_m^{m-2}) \cdot F_{m,t+1}, \tag{4.14}$$

$$\begin{aligned}
\text{where for } m \geq 3: \quad S_m^{m-1} =& \Pr\{\text{Exactly 1 player other than Alice transmits successfully}\} \\
=& 2(m-1)p_{m,t}\left[(1-p_{m,t})^{m-2} - (m-2)p_{m,t}(1-2p_{m,t})^{m-3}\right], \\
S_m^{m-2} =& \Pr\{\text{Exactly 2 players other than Alice transmit successfully}\} \\
=& (m-1)(m-2)p_{m,t}^2(1-2p_{m,t})^{m-3}, \\
S_m^{m} =& \Pr\{\text{No player transmits successfully}\} = 1 - S_m^{m-1} - S_m^{m-2}, \\
\text{for } m = 2: \quad S_2^1 =& 2p_{m,t}, \quad S_2^0 = 0, \quad S_2^2 = 1 - 2p_{m,t}.
\end{aligned} \tag{4.15}$$

By equating the right-hand sides of (4.12) and (4.14) we get an equality that includes the required probability $p_{m,t}$ and the expected future latencies $F_{m-1,t+1}, F_{m-2,t+1}$ and $F_{m,t+1}$.

Theorem 9. *An anonymous, time-dependent protocol with ternary feedback for n players and 2 transmission channels, with transmission probability $p_{m,t}$ of any of m pending players at $t \geq 1$, is an equilibrium protocol if and only if the right-hand sides of* (4.12) *and* (4.14) *are equal for some $p_{m,t} \in (0, 1/2]$.*

The equilibrium probability depends on the number of pending players m, time t and the expected future latencies $F_{m,t+1}, F_{m-1,t+1}$ and $F_{m-2,t+1}$ and defines the equilibrium

protocol. However, it is difficult to be expressed in closed form. Contrary to the case of a single channel studied in [69], where $p_{m,t}$ can be nicely expressed as a function of $F_{m-1,t+1}$ and $F_{m,t+1}$ in closed form, this does not seem to be the case in the current setting.

We should mention here that in the single-channel setting studied in [69] the decision rule $p_{m,t} = 1$ for $m \geq 3$ is in equilibrium. However, in the case of two channels, a similar result (e.g. $p_{m,t} = 1/2$) for any number of pending players does not seem to hold. Indeed, in time t with $m \geq 5$ pending players playing $p_{m,t} = 1/2$, the best response with strictly better expected latency is $p_{m,t} = 0$.

4.3.2 History-independent FIN-EQ protocols

Let us now consider anonymous, history-independent protocols, that is, protocols whose decision rules depend only on the number $1 \leq m \leq n$ of pending players. Now, the decision rule p_m of the players does not depend on their transmission history (and therefore on time as well), hence a player's expected future latency F_m does not depend on her transmission history. In this class of protocols the following theorem fully characterizes the equilibria.

Theorem 10. *There exists a unique, anonymous, history-independent, equilibrium protocol with ternary feedback for n players and 2 transmission channels, which is: any player among $2 \leq m \leq n$ remaining players, for every $t \geq 1$ attempts transmission to each channel with equal probability p_m. This probability is $\Theta(\frac{1}{\sqrt{m}})$ and yields expected future latency $e^{\Theta(\sqrt{m})}$ for every player.*

Proof. By manipulating the equilibrium conditions (4.12) and (4.14) we find

$$F_m = \frac{\left[Q_{m-1}^{m-2} S_m^{m-1} + S_m^{m-2}(1 - Q_{m-1}^{m-1})\right] - Q_m^{m-1}(Q_{m-1}^{m-2} - S_m^{m-2})}{(1 - Q_m^m)\left[Q_{m-1}^{m-2} S_m^{m-1} + S_m^{m-2}(1 - Q_{m-1}^{m-1})\right] - Q_{m-1}^{m-2} Q_m^{m-1}(1 - S_m^m)}. \tag{4.16}$$

From this we can also get F_{m-1}, thus, replacing these two in relation (4.12), which, in the history-independent case becomes

$$(1 - Q_m^m) F_m = 1 + Q_m^{m-1} F_{m-1}, \tag{4.17}$$

we get the recurrence relation for the transmission probability p_m to each channel. The resulting recurrence relation of p_m is non-linear with non-constant coefficients and for its

form there is no methodology in the literature that solves it - to the authors' knowledge. However, we can determine the asymptotic behaviour of p_m in the following way.

First, we show by induction that p_m is uniquely determined. The recurrence relation of p_m holds for $m \geq 2$ since our probabilities Q and S are defined for this domain only. That is because probabilities Q and S stem from the requirement that "transmission" and "no transmission" are both in the support of the decision rule for a player, which is not true in the case of $m = 1$. As a base case of our induction we use $m = 2$, for which we find from (4.12) and (4.14) as unique solution the pair ($p_2 = 1/2$, $F_2 = 2$). Now consider some $m \geq 2$ and assume that all $p_{m'}$ are uniquely determined for every m', $2 \leq m' \leq m$, and thus all $F_{m'}$ are uniquely determined by (4.16). Let us replace m with $m+1$ in (4.17), and fix p_m and F_m with the known ones. This gives us a rational univariate function - let us call it h - of p_{m+1}, i.e. $h(p_{m+1}) = (1 - Q_{m+1}^{m+1})F_{m+1} - 1 - Q_{m+1}^{m}F_m$. We would like to find the roots of h in the interval $(0, 1/2]$. By substituting Q_{m+1}^{m+1}, Q_{m+1}^{m} and F_m from (4.13) and (4.16) respectively, and then examining the first and second derivative of h, we can see that $h(0) = 0$, h has its unique minimum for some $p'_{m+1} \in (0, 1/2)$, and it is strictly decreasing in $[0, p'_{m+1}]$. In $[p'_{m+1}, 1/2]$ it is strictly increasing and $h(1/2) \geq 0$. Therefore, in $(0, 1/2]$ there is a unique root p^*_{m+1} of h.

Now we proceed in showing that the asymptotic behaviour in both sides of the recurrence relation (4.17) is the same for $p_m \in \Theta(1/\sqrt{m})$. First, we express the probabilities Q and S (see sets of equations (4.13) and (4.15)) in terms of $Q_m^{\times}$, and then we put $p_{m,t} = p_m \in \Theta(1/\sqrt{m})$. This gives:

$$Q_m^{\times} \in e^{-\Theta(\sqrt{m})}, \quad Q_m^{m-1} = Q_m^{\times} \cdot f_1(m), \quad Q_m^{m-2} = 0, \quad Q_m^{m} = 1 - Q_m^{\times} \cdot f_2(m), \quad \text{and}$$
$$S_m^{\times} = 0, \quad S_m^{m-1} = Q_m^{\times} \cdot g_1(m), \quad S_m^{m-2} = \left(Q_m^{\times}\right)^2 \cdot g_2(m), \quad S_m^{m} = 1 - Q_m^{\times} \cdot g_3(m),$$

where the functions $f_1(m), f_2(m), g_1(m), g_3(m)$ are in $\Theta(\sqrt{m})$ and $g_2(m)$ is in $\Theta(m)$. Now that we have described the asymptotic behaviour of the probabilities Q and S, we can find the asymptotic behaviour of the expected future latency F_m using (4.16). By carefully simplifying the numerator and denominator in the right-hand side of (4.16) we get

$$F_m = \frac{1}{Q_m^{\times} \cdot h_1(m)}, \quad \text{where } h_1(m) \in \Theta(\sqrt{m}).$$

Recall that $Q_m^{\times} \in e^{-\Theta(\sqrt{m})}$, thus $F_m \in e^{\Theta(\sqrt{m})}$. The above formula for F_m also implies that $F_{m-1} = 1/\left(Q_{m-1}^{\times} \cdot h_2(m)\right)$, where $h_2(m) \in \Theta(\sqrt{m})$. By substituting F_m and F_{m-1} in the

recurrence relation (4.17), we show that the asymptotic behaviour in both sides of it are the same, in particular, $\Theta(1)$. This completes the proof. □

The latter result is analogous to the one in [69] that characterizes anonymous, history-independent, equilibrium protocols with ternary feedback for the case of a single channel. However here, the proof methodology is different due to the fact that there is no known technique to express the equilibrium transmission probabilities in closed form, therefore their asymptotic behaviour can only be extracted from a recurrence relation, which, contrary to the one in [69], is quite complex. Using dynamic programming, we can compute the equilibrium probabilities in linear time; for up to $m = 100$ the equilibrium probabilities are presented in Figure 4.1.

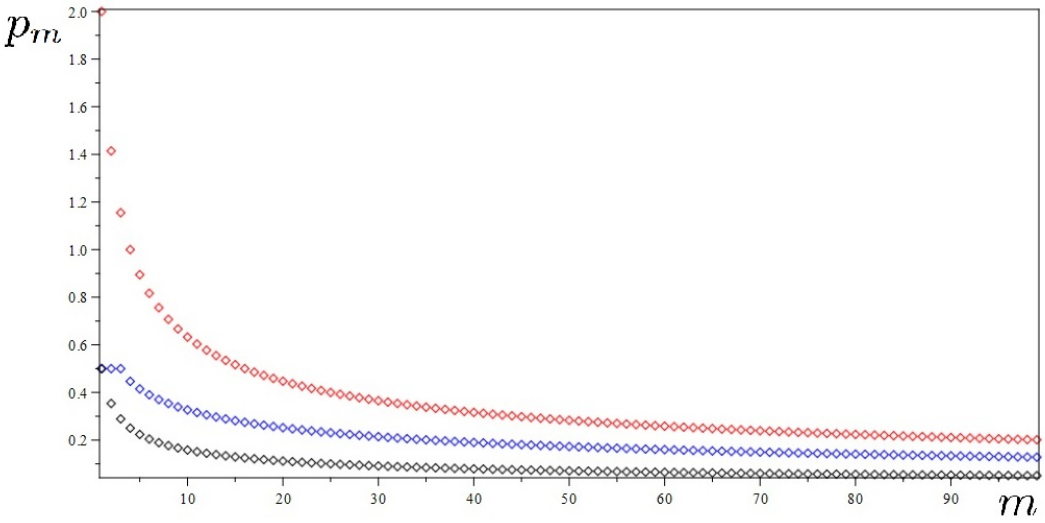


Figure 4.1: Blue: the equilibrium probabilities p_m for $2 \leq m \leq 100$. Red: experimental upper bound, function $\frac{2}{\sqrt{m-1}}$. Black: experimental lower bound, function $\frac{1}{2\sqrt{m-1}}$.

4.4 IN-EQ Protocols for Both Feedback Classes

Ideally, we would like to find an anonymous, equilibrium protocol that is efficient (i.e. the time until all players transmit successfully is $\Theta(n/k)$ with high probability) and also has finite expected latency. For the case of ternary feedback and a single channel such a

protocol was found in [69]. That protocol sets a deadline $t_0 \propto n$ after which it prescribes to the players to transmit with probability 1 on the channel at every time. It is easy to see that transmitting surely at every time is an equilibrium for more than 2 players. The idea of that protocol was to employ that "bad equilibrium" by putting it after the deadline so as to keep the players that were unsuccessful until t_0 for a very long (exponential in n) time. This works as a threat to the players, which they try to avoid by adopting a cooperative behaviour; using a time-dependent, equilibrium protocol they attempt transmission with probability low enough so that all of them are successful before the deadline with high probability. After the long part of the protocol, there is a last part that prescribes to the players to use a history-independent, equilibrium protocol (similar to the one we find for the 2-channel case) which has finite expected future latency. Since all three parts of the protocol are in equilibrium, the whole protocol is in equilibrium as well.

However, for the case of multiple channels in both the ternary feedback and acknowledgement-based feedback classes, a protocol with the above structure cannot be constructed as the following theorem shows. First, let us define the following notion of equilibrium protocol: By *equilibrium with blocking step (EBS)* we call an anonymous, equilibrium protocol with the property that there exists a time-step ($< \infty$) of the protocol in which every pending player has probability of successful transmission equal to 0.

Theorem 11. *In the classes of acknowledgement-based and ternary feedback protocols with $k \geq 2$ channels and $n \geq 2$ players, there exists no equilibrium protocol with blocking step (EBS) and finite expected latency.*

Proof. Assume for the sake of contradiction, that f is an anonymous equilibrium protocol with finite expected latency and it has a blocking step. Suppose all n players follow f, therefore the protocol profile is $\vec{f} = (f_1, f_2, \ldots, f_n) = (f, f, \ldots, f)$. Also, suppose that the blocking step is at $t = t_0$, which means that in any combination of personal histories $\vec{h}_{t_0} = (h_{1,t_0}, h_{2,t_0}, \ldots, h_{m,t_0})$ of the $m \leq n$ pending players which happens with positive probability under $\vec{f}$, no player transmits successfully. Additionally, since f is an equilibrium protocol, at t_0 the probability that some channel is "free" is 0, because if not, a player could deviate unilaterally at t_0 by choosing that channel with some positive probability, and thus improve her expected latency.

Consider the set H that contains all combinations of personal histories $\vec{h}_{t_0-1} = (h_{1,t_0-1}, h_{2,t_0-1}, \ldots, h_{m,t_0-1})$. Consider also the subset $H_>$ of H that contains all combinations that happen with positive probability, and the subset $H_0 = H \setminus H_>$ containing

those that happen with probability 0. For the reasons explained above, any combination $\vec{h}'_{t_0-1} \in H_>$ is characterized by the property that the combination of decision rules of the m pending players at t_0 that it produces necessarily has at least 2 players assigning probability 1 on each channel (so that every channel is surely blocked and no player can deviate unilaterally). Any combination $\vec{h}'_{t_0-1}$ that does not have this property must be in H_0, otherwise a player that under f would assign probability 1 on a surely blocked channel at t_0, she could unilaterally deviate by assigning at t_0 an appropriate positive probability on a channel that is "free" with positive probability, and decreasing her expected latency.

Now pick an arbitrary element $\vec{h}'_{t_0-1}$ of $H_>$, and without loss of generality, suppose that players $1, 2, \ldots, 2k$ block all k channels with probability 1 at t_0. That is

$$\begin{aligned}
f_1(h_{1,t_0-1}) &= (\Pr\{X_{1,t_0} = 1\} = 1,\ \Pr\{X_{1,t_0} \neq 1\} = 0)\,, \\
f_2(h_{2,t_0-1}) &= (\Pr\{X_{2,t_0} = 1\} = 1,\ \Pr\{X_{2,t_0} \neq 1\} = 0)\,, \\
f_3(h_{3,t_0-1}) &= (\Pr\{X_{3,t_0} = 2\} = 1,\ \Pr\{X_{3,t_0} \neq 2\} = 0)\,, \\
f_4(h_{4,t_0-1}) &= (\Pr\{X_{4,t_0} = 2\} = 1,\ \Pr\{X_{4,t_0} \neq 2\} = 0)\,, \\
&\vdots \\
f_{2k-1}(h_{2k-1,t_0-1}) &= (\Pr\{X_{2k-1,t_0} = k\} = 1,\ \Pr\{X_{2k-1,t_0} \neq k\} = 0)\,, \\
f_{2k}(h_{2k,t_0-1}) &= (\Pr\{X_{2k,t_0} = k\} = 1,\ \Pr\{X_{2k,t_0} \neq k\} = 0)\,.
\end{aligned} \tag{4.18}$$

Consider now that the event $\vec{h}^*_{t_0-1}$ where all players have the same history h_{1,t_0-1} right before t_0, and observe that this happens with positive probability, because the game starts with all players having the exact same history, i.e. the empty history, and can continue having the same history by transmitting to the exact same channels for every $t \leq t_0 - 1$ since they will be prescribed identical decision rules by protocol f. Therefore $\vec{h}^*_{t_0-1} \in H_>$. In such an event, all players, just like player 1 in (4.18) above, will transmit with probability 1 on channel 1 at t_0. Therefore the remaining $k-1$ channels will be "free" at t_0, therefore $\vec{h}^*_{t_0-1} \in H_0$ which is a contradiction.

In the above analysis, the arguments about a player being able to unilaterally deviate and decrease her expected latency, need the extra property that the expected latency of the player is finite; because if the expected latency is infinite, unilateral deviation does not make it finite, therefore the player has no incentive to deviate. The proof is complete. □

This impossibility result discourages the search for anonymous, efficient, multiple-

channel, equilibrium protocols with the additional property of finite expected latency, since it seems to the authors that the only candidate that guarantees efficiency is a deadline protocol; but deadline protocols are successful as long as they are EBS. Whether no anonymous, efficient, equilibrium protocol with finite expected latency can be found for multiple channels is one of the most interesting open problems that stem from this work.

Due to the latter impossibility theorem, in the rest of this section we drop the "finite expected latency requirement" and present IN-EQ protocols within the classes of acknowledgement-based and ternary feedback for the general case of $k \geq 1$ channels and any number of $n \geq 2k+1$ players. For this, we employ the deadline idea introduced in [69] and consequently used in [43,44]. Our protocols are efficient, even though the expected latency is infinite.

4.4.1 Acknowledgement-based feedback

We provide an efficient deadline protocol (see Protocol g_1 below) with infinite expected latency for $k \geq 1$ channels and $n \geq 2k+1$ players. This protocol generalizes the efficient protocol of [43] which deals with a single channel and at least 3 players. The general protocol we present uses their idea, that is, estimating the number of pending players (since it is not known in the acknowledgement-based environment) and adjusting the transmission probabilities of the players accordingly, in order to simulate a socially optimal protocol (see protocol SOP below) that allows all transmissions to be successful by time $\Theta(n/k)$ with high probability. Our modification is that, instead of prescribing to the players to always transmit to the single channel once they reach the deadline (so that with some positive probability they get blocked forever), we block all channels with positive probability by prescribing a random assignment of each player to a channel.

In particular, consider $k \geq 1$ transmission channels, $n \geq 2k+1$ players, a fixed constant $\beta \in (0,1)$ and a deadline t_0 to be determined consequently. The window of time steps $\{1, 2, \ldots, t_0 - 1\}$ is partitioned into $r+1$ consecutive intervals (sets of consecutive time steps) $I_1, I_2, \ldots, I_{r+1}$ where r is the unique integer in $\left[-\log_\beta n/2 - 1, -\log_\beta n/2\right]$. For any $j \in \{1, 2, \ldots, r+1\}$ define $n_j = \beta^j n/k$. For $j \in \{1, 2, \ldots, r\}$ the length of interval I_j is $l_j = \lfloor \frac{e}{\beta} n_j \rfloor$. Interval I_{r+1} is special and has length $l_{r+1} = n/k$. We define the following anonymous protocol.

Protocol $\mathbf{g_1}$:
Every player among $1 \leq m \leq n$ pending players for every time step $t \in I_j$ assigns transmission probability $1/\max\{n_j, k\}$ to each channel. Right before the deadline $t_0 = 1 + \sum_{j=1}^{r+1} l_j$ each pending player is assigned to a random channel equiprobably, and for $t \geq t_0$ always attempts transmission to that channel.

Lemma 8. *Protocol g_1 for $n \geq 2k+1$ players and $k \geq 1$ channels, is an equilibrium protocol and it is also efficient.*

Proof. First we prove that g_1 is an equilibrium protocol when $n \geq 2k+1$. Consider an arbitrary player i, and observe that since all players play g_1 the probability that all of them will be still pending by t_0 is $1/n^{t_0} > 0$. Given that, the probability that player i at t_0 will be assigned to the same channel with at least 2 other players is at least the probability that she will be assigned to the same channel with all other players, which is at least $1/k^n > 0$. Hence, the probability that player i can find herself in $t = t_0$ pending together with two other players is positive, and in this case she will remain pending forever. Therefore, i's expected latency is ∞, and since by any unilateral deviation of i she cannot make the aforementioned event empty, her expected latency will always be ∞. Therefore, g_1 is an equilibrium protocol.

Now we proceed by showing that g_1 is also efficient, that is, all players transmit successfully by time $t_0 = 1 + \sum_{j=1}^{r+1} l_j \in \Theta(n/k)$. The proof of efficiency is essentially the same as that of Theorem 11 in [43] and it is omitted. The difference here is that we have tuned n_j and l_{r+1} according to our problem and we have used variable r instead of k. As a consequence, this result is the same as that of the aforementioned theorem in [43], except that ours has n/k instead of n. □

4.4.2 Ternary feedback

In the ternary feedback setting, the use of the unique history-independent equilibrium (see Theorem 10, Section 4.3.2) yields exponential expected latency in the number of players n, and additionally, even one player's latency being any polynomial in n happens with exponentially small probability. This fact points to history-dependent protocols as candidates for efficient equilibria. Here, we look in the more restricted class of time-dependent protocols, where only time t is considered by each player and not her entire transmission

history. We construct a protocol (Theorem 12) which imposes a heavy cost on any player that does not manage to transmit successfully until a certain deadline-round. This forces any potential deviator to play "fairly" until the deadline and follow an anonymous, socially optimal protocol, named SOP (guarantees expected time $\Theta(n/k)$ for all players to pass).

To prove the main theorem of this section we need a series of technical results. As a first step, we give the general Lemma 9 that determines the expected number of successful transmissions in a round where m players have a uniform distribution on the channels, and subsequently in Fact 1 we find the maximum of that expected number. Then, we present another lemma (Lemma 10) that gives an upper bound on the expected finishing time when $m \leq k$. Finally, using all the aforementioned intermediate results, we present a socially optimal protocol in Lemma 11 which is employed in the proof for our main theorem (Theorem 12) that concerns our IN-EQ protocol.

Lemma 9. *Consider a single round with $k \geq 1$ channels and $n \geq 1$ players. Assume that for every player the probability of transmission attempt is $z \in [0,1]$ which she splits equally to all k channels. Then, the expected number[3] of players that transmit successfully is $zn\left(1-\frac{z}{k}\right)^{n-1}$.*

Proof. For a fixed $z \in [0,1]$, denote by X_n the random variable that indicates how many players transmit successfully in a round with n players. Note that when $z = 1$ and $n \geq 2$, the case where $X_n = n-1$ is impossible since in order for some player to have a failed transmission she has to be blocked by someone else.

Our problem reduces to the following balls-and-bins problem: Consider n balls and k bins, where $n \geq 1$. Each ball is thrown with probability z/k to each bin, and not thrown at all with probability $1-z$. Random variable $X_n \in \{0,1,2,\ldots,n\}$ now indicates the number of bins that had a single ball after the experiment.

We want to find $\mathbb{E}[X_n]$. For this, we will employ the probability of the event that x bins contain a single ball given that the round started with n balls. Denote by A_j the event that bin j contains a single ball. Also, we define the probabilities of intersections between such events

$$p_j = \Pr(A_j), \quad p_{jm} = \Pr(A_j \cap A_m), \quad p_{jml} = \Pr(A_j \cap A_m \cap A_l), \quad \ldots$$

[3] We define $0^0 = 1$.

and we write S_r to denote the sums of all distinct p's with r subscripts. That is

$$S_1 = \sum_{j=1}^{k} p_j, \quad S_2 = \sum_{j<m} p_{jm}, \quad S_3 = \sum_{j<m<l} p_{jml}, \quad \dots$$

where the subscripts are in increasing order $j < m < l < \cdots < k$ for uniqueness, so that in the sums each combination appears only once; therefore S_r has $\binom{k}{r}$ terms. In our setting, each term of S_r equals

$$\binom{n}{r} r! \left(\frac{z}{k}\right)^r \left(1 - \frac{rz}{k}\right)^{n-r}$$

since for specific r bins to contain a single ball there are $\binom{n}{r}$ combinations of r balls, which should occupy the r bins with $r!$ orders. Each of those chosen r balls can fall in a bin with probability $\frac{z}{k}$ and each of the rest $n-r$ balls has to fall in some other than those r bins or not be thrown at all, which happens with probability $1 - \frac{rz}{k}$. So,

$$S_r = \binom{k}{r} \binom{n}{r} r! \left(\frac{z}{k}\right)^r \left(1 - \frac{rz}{k}\right)^{n-r}$$

and by the Inclusion-Exclusion Theorem, the probability that exactly x bins contain a single ball is the following[4]

$$\begin{aligned} P_n(x) &= \sum_{r=x}^{n} (-1)^{r-x} \binom{r}{x} S_r \\ &= \sum_{r=x}^{n} (-1)^{r-x} \binom{r}{x} \binom{k}{r} \binom{n}{r} r! \left(\frac{z}{k}\right)^r \left(1 - \frac{rz}{k}\right)^{n-r} \end{aligned} \tag{4.19}$$

[4]For the case where $a < b$ we define $\binom{a}{b} \triangleq 0$ so that the analysis is displayed only once for both cases $n \le k$ and $n > k$.

We want to calculate $\mathbb{E}[X_n]$. We have

$$\begin{aligned}
\mathbb{E}[X_n] &= \sum_{x=0}^{n} xP_n(x) \\
&= \sum_{x=0}^{n}\sum_{r=x}^{n}(-1)^{r-x}x\binom{r}{x}\binom{k}{r}\binom{n}{r}r!\left(\frac{z}{k}\right)^r\left(1-\frac{rz}{k}\right)^{n-r} \\
&= \sum_{r=0}^{n}\sum_{x=0}^{r}(-1)^{r-x}x\binom{r}{x}\binom{k}{r}\binom{n}{r}r!\left(\frac{z}{k}\right)^r\left(1-\frac{rz}{k}\right)^{n-r} \\
&= \sum_{r=0}^{n}\binom{k}{r}\binom{n}{r}r!\left(\frac{z}{k}\right)^r\left(1-\frac{rz}{k}\right)^{n-r}\sum_{x=0}^{r}(-1)^{r-x}x\binom{r}{x} \\
&= \sum_{r=0}^{n}\binom{k}{r}\binom{n}{r}r!\left(\frac{z}{k}\right)^r\left(1-\frac{rz}{k}\right)^{n-r}(-1)^r\sum_{x=0}^{r}(-1)^{x}x\binom{r}{x} \\
&= \binom{k}{1}\binom{n}{1}\frac{z}{k}\left(1-\frac{z}{k}\right)^{n-1}(-1)(-1) \\
&\qquad\qquad \left(\text{since } \sum_{x=0}^{r}(-1)^x x\binom{r}{x} = -1 \text{ for } r=1, \quad 0 \text{ otherwise}\right) \\
&= zn\left(1-\frac{z}{k}\right)^{n-1}.
\end{aligned}$$

□

The following fact shows where the expected number of players of the above theorem is maximized as a function of z (the probability mass devoted to transmission).

Fact 1. *Consider the function $f(z) = zn(1-z/k)^{n-1}$, with domain $[0,1]$ and parameters $k \geq 1$, and $n \geq 1$. The maximum of f is attained for $z = \min\{k/n, 1\}$.*

Proof. The first and second derivatives of f are

$$\begin{aligned}
f'(n) &= n\left(1-\frac{zn}{k}\right)\left(1-\frac{z}{k}\right)^{n-2} \\
f''(n) &= n(n-1)\left(1-\frac{z}{k}\right)^{n-3}\frac{nz-2k}{k^2}
\end{aligned}$$

When $n < k$, then $f'(z) > 0$ and therefore the global maximum of f is attained for $z = 1$, which gives $f(1) = n(1-1/k)^{n-1}$.

When $n \geq k$, the first derivative of f is 0 for (a) $z = k$ when $n \geq 3$, or (b) $z = k/n$

when $n \geq 1$. Case (a) only works if $k = 1$ due to the domain of z and gives $f(1) = 0$. $f'(z)$ is positive in $[0, k/n)$, and negative in $(k/n, 1)$. Therefore, $f(k/n) = k(1 - 1/n)^{n-1}$ is the global maximum. □

Lemma 10. *Suppose there are $k \geq 2$ channels and $2 \leq n \leq k$ players and suppose that all players use the following protocol: A player in every time step $t \geq 1$ has a probability of transmission $1/k$ to every channel. Then, the expected time until everyone transmits successfully is upper bounded by $\frac{1}{1-\ln(e-1)} \ln(\frac{n}{2}) + \left(1 - \frac{1}{k}\right)^{-1}$.*

Proof. Denote by X_m the random variable that indicates how many players transmit successfully in a round t where $m \in \{0, 2, \ldots, n\}$ players are left. Note that the case where $m = 1$ is impossible since in order for some player to have a failed transmission she has to be blocked by someone else. In the next round the expected number of players will be $m - \mathbb{E}[X_m]$. We define the finishing time as the following random variable $T \triangleq inf\{t : m = 0\}$ and we would like to find its expectation.

Our problem reduces to the following balls and bins problem: Consider n balls and k bins, where $2 \leq n \leq k$. At time $t = 1$ all balls are thrown uniformly at random to the k bins. For all the bins that contain a single ball, these balls are removed, and in the next round $m \in \{0, 2, \ldots, n\}$ balls remain. At time $t = 2$ all m balls are thrown uniformly at random to the k bins. The process continues as long as there are remaining balls. Random variable $X_m \in \{0, 1, 2, \ldots, m\}$ now indicates the number of bins that had a single ball when the respective round started with m balls. Note again that $\Pr(X_m = m-1) = 0$ since there is no allocation of balls in the bins such that $m-1$ bins have a single ball. Random variable $T \triangleq inf\{t : m = 0\}$ is the finishing time of this process.

We define the function $f(m)$ to be the expected finishing time $\mathbb{E}[T]$ when m players remain. We assume that this function is non-decreasing and concave. Then we have,

$$\begin{aligned} f(m) &= 1 + \sum_{i=0}^{m} \Pr(X_m = i) f(m - X_m) \\ &= 1 + \mathbb{E}[f(m - X_m)] \\ &\leq 1 + f\left(\mathbb{E}[m - X_m]\right) && \text{(concavity of } f \text{ and Jensen's inequality)} \\ &= 1 + f(m - \mathbb{E}[X_m]) && \text{(linearity of expectation)} \end{aligned} \quad (4.20)$$

Now by exploiting the monotonicity of the function $f(m)$ in equation (4.20), and using Lemma 9 we only need to find a lower bound on $\mathbb{E}[X_m]$. This is easy, since $m\left(1 - \frac{1}{k}\right)^{m-1} \geq$

$m\left(1-\frac{1}{k}\right)^{k-1} \geq m/e$. Then from equation (4.20) we get

$$\begin{aligned} f(m) &\leq 1 + f\left(m\left(1-\frac{1}{e}\right)\right) \\ &\leq r + f\left(m\left(1-\frac{1}{e}\right)^{r}\right). \end{aligned}$$

We use as base case $f(2)$ for which holds that $f(2) = 1 + k\frac{1}{k^2}f(2)$, or equivalently, $f(2) = (1-1/k)^{-1}$. Then the r for which $m\left(1-\frac{1}{e}\right)^{r} = 2$ finally gives us

$$f(m) \leq \frac{1}{1-\ln(e-1)}\ln(\frac{m}{2}) + \left(1-\frac{1}{k}\right)^{-1}$$

□

Let us define the following anonymous, history-independent protocol which we prove to be efficient. However, we remark that it is not in equilibrium, due to Theorem 10 which characterizes the unique, anonymous, equilibrium protocol that is history-independent.

Protocol SOP:
Every player among $1 \leq m \leq n$ pending players, in each round $t \geq 1$ assigns transmission probability $1/\max\{m,k\}$ to each channel.

Lemma 11. *Protocol SOP for $k \geq 1$ channels and $n > k$ players has expected finishing time $O((n-k)/k)$.*

Proof. Suppose protocol SOP as stated in the theorem is used. Then, the transmission probability of each player in each round is uniform on the set of channels K. Using the framework of Lemma 9, according to protocol SOP for variable z we have $z = \min\{k/m, 1\}$, and we know from Fact 1 that this value maximizes the number of successful transmissions in a round with m players. Denote by X_m the random variable that keeps track of the number of successful transmissions in a single round with $m > k$ pending players. Then, according to Lemma 9, in a round with $m > k$ pending players it is $\mathbb{E}[X_m] = k(1-1/m)^{m-1}$.

Define the function $f(m)$ to be the expected finishing time when there are $m > k$ pending players. We assume that this function is non-decreasing and concave. Then we

have

$$\begin{aligned} f(m) &= 1 + \sum_{i=0}^{m} \Pr(X_m = i) f(m - X_m) \\ &= 1 + \mathbb{E}[f(m - X_m)] \\ &\leq 1 + f\left(\mathbb{E}[m - X_m]\right) && \text{(concavity of } f \text{ and Jensen's inequality)} \\ &= 1 + f(m - \mathbb{E}[X_m]) && \text{(linearity of expectation)} \end{aligned} \quad (4.21)$$

Now by exploiting the monotonicity of the function $f(m)$ in equation (4.21), and using Lemma 9 we only need to find a lower bound on $\mathbb{E}[X_m]$. This is easy, since $k\left(1 - \frac{1}{m}\right)^{m-1} \geq k/e$. Then from equation (4.21) we get

$$\begin{aligned} f(m) &\leq 1 + f\left(m - \frac{k}{e}\right) \\ &\leq r + f\left(m - r\frac{k}{e}\right). \end{aligned}$$

We use as base case $f(k)$ for which holds that $f(k) \leq \frac{1}{1-\ln(e-1)} \ln(\frac{k}{2}) + \left(1 - \frac{1}{k}\right)^{-1}$, due to Lemma 10. Then the r for which $m - r\frac{k}{e} = k$ finally gives us

$$f(m) \leq e\frac{m-k}{k} + \frac{1}{1 - \ln(e-1)} \ln(\frac{k}{2}) + \left(1 - \frac{1}{k}\right)^{-1}.$$

□

Using the above lemmata we are able to prove the following.

Lemma 12. *(a) If at $t = 0$ there are n pending players, the probability that more than k players are pending at time $t_1 = 2e(n-k)/k$ is at most $exp\left(-\frac{n-k}{2ek}\right)$.*
(b) If at $t = 0$ there are k pending players, the probability that not all players have transmitted successfully at time $t_2 = 2e(n-k)/k$ is at most $exp\left(-\frac{n-k}{2ek}\right)$.

Proof. Let $\{Y_t\}_{t=1}^{t_1}$ be random variables which indicate the number of successful transmissions that occur in each time-step from $t = 1$ up to $t_1 \triangleq 2e(n-k)/k$, given that there are n pending players at time $t = 0$. For the events for which $Y \triangleq \sum_{t=1}^{t_1} Y_t > n - k$ we have the desired outcome. For the rest, since the pending players in each round $1 \leq t \leq t_1$ are $m > k$, the protocol prescribes to each player probability $1/m$ on each channel. There-

fore, by Lemma 9, we have $\mathbb{E}[Y_t] = k\,(1-1/m)^{m-1}$. In the next claim we show that Y_t stochastically dominates a random variable $Z_t \in \{0,1,\ldots,k\}$ that indicates the number of successful transmissions in round $1 \leq t \leq t_1$ but, in this process, the players that transmit successfully are placed back to the group of pending players.

Claim 4. *Pr{$Y_t \geq x$} $\geq$ Pr{$Z_t \geq x$}, for all $x \in \{0,1,\cdots,k\}$.*

Proof. We will prove the above claim by showing the stronger fact that, for any fixed number $1 \leq m \leq n-1$ of pending players at time t,

$$\Pr\{Y_t \geq x \mid m \text{ pending players}\} \geq \Pr\{Y_t \geq x \mid m+1 \text{ pending players}\},$$

for all $x \in \{0,1,\cdots,k\}$.

Indeed, by substituting the probabilities of the above inequality we get,

$$\binom{m}{x} x! \left(\frac{1}{m}\right)^x \left(1-\frac{x}{m}\right)^{m-x} \geq \binom{m+1}{x} x! \left(\frac{1}{m+1}\right)^x \left(1-\frac{x}{m+1}\right)^{m+1-x},$$

$$\text{or equivalently,} \quad (m+1)^m (m-x)^{m-x} \geq m^m (m-x+1)^{m-x},$$

$$\text{and finally,} \quad \left(1+\frac{1}{m}\right)^m \geq \left(1+\frac{1}{m-x}\right)^{m-x},$$

which is true, since the function $f(w) = (1+1/w)^w$ is strictly increasing. The claim follows from the fact that for any fixed $x \in \{0,1,\cdots,k\}$,

$$\Pr\{Z_t \geq x\} = \Pr\{Y_t \geq x \mid n \text{ pending players}\}. \qquad \square$$

Clearly $\{Z_t\}_{t=1}^{t_1}$ are independent random variables bounded in $[0,k]$. Let $Z \triangleq \sum_{t=1}^{t_1} Z_t$ and $\mu_1 \triangleq \mathbb{E}[Z] = \sum_{t=1}^{t_1} \mathbb{E}[Z_t] = t_1 k\,(1-1/n)^{n-1}$. Then by Hoeffding's inequality [86] and the stochastic domination we have,

$$\Pr(Y \leq n-k) \leq \Pr(Z \leq n-k) = \Pr\left(Z \leq \frac{\mu_1}{2e\,(1-1/n)^{n-1}}\right) \leq \Pr\left(Z \leq \frac{\mu_1}{2}\right)$$

$$\leq exp\left(-\frac{(1-1/2)^2\mu_1^2}{t_1(k-0)^2}\right) \leq exp\left(-\frac{1}{4}\frac{t_1}{e^2}\right) = exp\left(-\frac{n-k}{2ek}\right),$$

where in the last three inequalities we used the fact that $(1-1/n)^{n-1} \geq 1/e$.

For the second part of the proof, suppose the process is at round $t = 0$ with k pending players. Let $\{X_t\}_{t=1}^{t_2}$ be random variables which indicate the number of successful transmissions that occur in each time-step from $t = 1$ up to $t_2 \triangleq 2e(n-k)/k$, given that there are k pending players at time $t = 0$. The pending players in each round $1 \leq t \leq t_2$ are $m \leq k$, hence the protocol prescribes to each player probability $1/k$ on each channel. By Lemma 9, we have $\mathbb{E}[X_t] = m\,(1-1/k)^{m-1}$. Now, observe that X_t stochastically dominates a random variable $W_t \in \{0, 1, \ldots, k\}$ that indicates the number of successful transmissions in round $1 \leq t \leq t_2$ but, in this process, the players that transmit successfully are placed back to the group of pending players. The latter observation is easy to see since an argument similar to the Claim that was stated earlier holds in this case.

Clearly, $\{W_t\}_{t=1}^{t_2}$ are independent random variables bounded in $[0, k]$. Let $W \triangleq \sum_{t=1}^{t_2} W_t$ and $\mu_2 \triangleq \mathbb{E}[W] = \sum_{t=1}^{t_2} \mathbb{E}[W_t] = t_2 k\,(1-1/k)^{k-1}$. Then by Hoeffding's inequality [86] and the stochastic domination we have,

$$\begin{aligned}
\Pr(X \leq k-1) \leq \Pr(W \leq k) &= \Pr\left(W \leq \frac{\mu_2 k}{2e(n-k)\,(1-1/k)^{k-1}}\right) \\
&\leq \Pr\left(W \leq \frac{\mu_2}{2}\right) \leq exp\left(-\frac{(1-1/2)^2\mu_2^2}{t_2(k-0)^2}\right) \leq exp\left(-\frac{1}{4}\frac{t_2}{e^2}\right) \\
&= exp\left(-\frac{n-k}{2ek}\right),
\end{aligned}$$

where in the last three inequalities we used the fact that $(1-1/k)^{k-1} \geq 1/e$, and $n \geq 2k+1$. This completes the proof of the lemma. □

We define the following anonymous protocol. In the next theorem we show that it is an equilibrium protocol and also that it is efficient.

Protocol *r*:

Let the deadline be $t_0 = 4e(n-k)/k$. Every player among $1 \leq m \leq n$ pending players for $1 \leq t \leq t_0 - 1$ assigns transmission probability $1/\max\{m, k\}$ to each channel. Right before t_0 each pending player is assigned to a random channel equiprobably, and for $t \geq t_0$ always attempts transmission to that channel.

Theorem 12. *Protocol r for $n \geq 2k+1$ players and $k \geq 1$ channels is an equilibrium protocol whose finishing time is $\Theta(n/k)$ with probability tending to 1 as $n/k \to \infty$.*

Proof. First, we show that it is an equilibrium protocol when $n \geq 2k+1$. The expected latency of a player using protocol r is ∞. That is because there is an event with positive probability in which some player i finds herself in an equilibrium where at least 2 of the other players have been assigned to each and all of the k channels and transmit there in every time slot. In particular, with probability at least $k(\frac{1}{n})^{t_0-1} > 0$ all players will be pending right after $t_0 - 1$. Given this, with probability $\binom{n-1}{2,2,\ldots,2,n-1-2k}(\frac{1}{k})^{n-1} > 0$ exactly 2 out of $n-1$ players will be assigned to each of the $k-1$ channels and the remaining players (including player i), which are at least 3, are assigned to the remaining channel. Therefore, the aforementioned two events occur with positive probability, and then for player i all channels are blocked for every $t \geq t_0$, resulting to infinite latency. Hence, the expected latency of a player using protocol r is ∞.

Now suppose that player i unilaterally deviates to some protocol r'. The event that all players are pending right before t_0 remains non-empty, since the event that all players transmit on the same channel as i for every $1 \leq t \leq t_0 - 1$ happens with positive probability. Given that, the event that at least 2 of the players other than i will be assigned to each channel happens with positive probability. Therefore, the deviator's expected latency remains ∞ and r is an equilibrium protocol.

Finally, we will show that, when $n \in \omega(k)$, this protocol is also efficient, i.e. the time until all n players transmit successfully is linear in n/k with probability tending to 1 as $\frac{n}{k} \to \infty$. By Lemma 12, the probability that not all players have successfully transmitted by time $t_1 + t_2 = 4e(n-k)/k$ is at most $\exp\left(-\frac{n-k}{2ek}\right) + \exp\left(-\frac{n-k}{2ek}\right) = 2\exp\left(-\frac{n-k}{2ek}\right)$. Therefore, when $n \in \omega(k)$, no player is pending after $4e(n-k)/k$ rounds with high probability. □

Chapter 5

Connected Subgraph Defense Games

This chapter studies a game in a structured population. We consider a security game over a network played between a *defender* and k *attackers*. Every attacker chooses, probabilistically, a node of the network to damage. The defender chooses, probabilistically as well, a connected induced subgraph of the network of λ nodes to scan and clean. Each attacker wishes to maximize the probability of escaping her cleaning by the defender. On the other hand, the goal of the defender is to maximize the expected number of attackers that she catches. This game is a generalization of the model from the seminal paper of Mavronicolas et al. [100]. We are interested in Nash equilibria of this game, as well as in characterizing *defense-optimal* networks which allow for the best *equilibrium defense ratio*; this is the ratio of k over the expected number of attackers that the defender catches in equilibrium.

We provide a characterization of the Nash equilibria of this game and defense-optimal networks. The equilibrium characterizations allow us to show that even if the attackers are centrally controlled the equilibria of the game remain the same. In addition, we give an algorithm for computing Nash equilibria. Our algorithm requires exponential time in the worst case, but it is polynomial-time for λ constantly close to 1 or n. For the special case of tree-networks, we further refine our characterization which allows us to derive a polynomial-time algorithm for deciding whether a tree is defense-optimal and if this is the case it computes a defense-optimal Nash equilibrium. On the other hand, we prove that it is `NP`-hard to find a best-defense strategy if the tree is not defense-optimal. We complement this negative result with a polynomial-time constant-approximation algorithm that computes solutions that are close to optimal ones for general graphs. Finally, we provide asymptotically (almost) tight bounds for the *Price of Defense* for any λ; this is the

worst equilibrium defense ratio over all graphs.

The results of this chapter have been published in the Proceedings of the 12th International Symposium on Algorithmic Game Theory (SAGT 2019) [4] (co-authored with Akrida, Deligkas and Spirakis).

5.1 Overview

With technology becoming a ubiquitous and integral part of our lives, we find ourselves using several different types of computer networks. An important issue when dealing with such networks, which are often prone to security breaches [42], is to prevent and monitor unauthorized access and misuse of the network or its accessible resources. Therefore, the study of network security has attracted a lot of attention over the years [139]. Unfortunately, such breaches are often inevitable, since some parts of a large system are expected to have weaknesses that expose them to security attacks; history has indeed shown several successful and highly-publicized such incidents [137]. Therefore, the challenge for someone trying to keep those systems and networks of computers secure is to counteract these attacks as efficiently as possible, once they occur.

To that end, inventing and studying appropriate theoretical models that capture the essence of the problem is an important line of research, ongoing for a few years now [102,103]. Here, extending some known models for very simple cases of attacks and defenses [100,101], we introduce and analyze a more general model for a scenario of network attacks and defenses modeling it as a *defense game*.

The Network Security Game. We follow the terminology established by the seminal paper of Mavronicolas et al. [100]. We consider a network whose nodes are vulnerable to infection by threats called *attackers*; think of those as viruses, worms, Trojan horses or eavesdroppers [73] infecting the components of a computer network. Available to the network is a security software (or firewall), called the *defender*. The defender is only able to "clean" a limited part of the network from threats that occur; the reason for the limited cleaning capacity of the defender may be, for example, the cost of purchasing a global security software. The defender seeks to protect the network as much as possible, and on the other hand, every attacker seeks to increase the likelihood of not being caught. Both the attackers and the defender make individual decisions for their positioning in the network with the aim to maximize their own objectives.

Every attacker targets (and attacks) a node chosen via her own probability distribution over the nodes of the network. The defender cleans a connected induced subgraph of the network with size λ, chosen via her own probability distribution over all connected induced subgraphs of the graph with λ nodes. The attack of a particular attacker is successful unless the node chosen by the attacker is incident to an edge (link) being cleaned by the defender, i.e. to an edge belonging in the induced subgraph chosen by the defender. One could equivalently think of the defender selecting a set of λ connected nodes to defend, and an attacker is successful if and only if she attacks a node that is not being defended. Since attacks and defenses over a large computer network are self-interested procedures that seek to maximize damage and protection, respectively, it is natural to model this network security scenario as a non-cooperative *strategic game* on graphs with two kinds of players: $k \geq 1$ *attackers*, each playing a *vertex* of the graph, and a single *defender* playing a *connected induced subgraph* of the graph. The *(expected) payoff* of an attacker is the probability that she is not caught by the defender; the *(expected) payoff* of the defender is the (expected) number of attackers she catches. We are interested in the Nash equilibria [112,113] associated with this graph theoretic game, where no player can unilaterally improve her (expected) payoff by switching to another probability distribution. We are also interested in understanding and characterizing the networks that allow for a good *defense ratio*: given a strategy profile, i.e. a combination of strategies for the network entities (attackers and defender), the defense ratio of a network is the ratio of the total number of attackers over the defender's expected payoff in that strategy profile.

5.1.1 Contribution

In this work we depart from and significantly extend the line of work of Mavronicolas et al. in their seminal paper [100] on defense games in graphs; we term the type of games we consider *Connected Subgraph Defense (CSD) games*. In our model the defender is more powerful than in [100–103], since her power is parameterized by the size, λ, of the defended part of the network. We allow λ to take values from 1 to n, while in [100–103] only the case where $\lambda = 2$ was studied. We study many questions related to CSD games. We extend the notions of *defense ratio* and *defense-optimal graphs* for CSD games. In fact, the defense ratio of a given graph G and a given strategy profile S of the attackers and the defender is the ratio of the number of attackers, k, over the defender's expected payoff (the number of attackers she catches on expectation). We thoroughly investigate the notion of the defense

ratio for Nash equilibria strategy profiles.

Firstly, we precisely characterize the Nash equilibria and defense-optimal graphs in CSD games. This allows us to show that, in equilibrium, the game version of k uncoordinated attackers and a single defender is equivalent to the version in which a single leader coordinates the k attackers, meaning that both versions of the game have the exact same equilibria and defense ratio. We present an LP-based algorithm to compute an exact equilibrium of any given CSD game, whose running time is polynomial in $\binom{n}{\lambda}$. Then, we focus on tree-graphs. There, we further refine our equilibrium characterization which allows us to derive a polynomial-time algorithm for deciding whether a tree is defense-optimal and, if this is the case, it computes a defense-optimal Nash equilibrium. A tree is defense-optimal if and only if it can be partitioned into $\frac{n}{\lambda}$ disjoint sub-trees. On the other hand, we prove that it is NP-hard to find a best-defense strategy if the tree is not defense-optimal. We remark that a very crucial parameter for defense-optimality of a graph G is the "best" probability with which any vertex of G is defended in a NE; we call that probability *MaxMin probability* and denote it by $p^*(G)$. Then, for any graph G, the defense ratio in equilibrium is shown to be exactly $\frac{1}{p^*(G)}$. Although it is hard to exactly compute $p^*(G)$ even for trees, we complement this negative result with a polynomial-time constant-approximation algorithm that computes solutions that are close to the optimal ones for any λ, for any given general graph. In particular, we approximate the (best) defense ratio of any graph within a factor of $2 + \frac{\lambda-3}{n}$. Finally, we provide asymptotically tight bounds for the Price of Defense for any $\lambda \in \omega(1) \cap o(n)$, and almost tight bounds for any other value of λ.

5.1.2 Related work

Our graph-theoretic game is a direct generalization of the defense game considered by Mavronicolas et al. [100–103]. In the latter, the authors examined the case where the size of the defended part of the network is $\lambda = 2$, i.e. where the defender "cleans" an edge. This leads to a nice connection between equilibria and (fractional) matchings in the graph [102]. But when λ is greater than 2, one has to investigate (as we shall see here) how to sparsely cover the graph by as small a number as possible of connected induced subgraphs of size λ. This direction can be seen as an extension of fractional matchings to covers of the graph by equisized connected subgraphs. Sparse covering of graphs by connected induced subgraphs (clusters), not necessarily equisized, is a notion known to be useful also for distributed algorithms, since it affects message communication complexity [15].

In another line of work, Kearns and Ortiz [89] study *Interdependent Security games* in which a large number of players must make individual decisions regarding security. Each player's *safety* may depend on the actions of the entire population (in a complex way). The graph-theoretic game that we consider could be seen as a particular instance of such games with some sort of limited interdependence: the actions of the defender and an attacker are interdependent, while the actions of the attackers are not dependent on each other.

Aspnes et al. [14] consider a graph-theoretic game that models containment of the spread of viruses on a network; each node individually must choose to either install anti-virus software at some cost, or risk infection if a virus reaches it without being stopped by some intermediate node with installed anti-virus software. Aspnes et al. [14] prove several algorithmic properties for their graph-theoretic game and establish connections to a certain graph-theoretic problem called *Sum-of-Squares Partition*.

A game on a weighted graph with two players, the *tree player* and the *edge player*, was studied by Alon et al. [5]. At each play, the tree player chooses a spanning tree and the edge player chooses an edge of the graph, and the payoffs of the players depend on whether the chosen edge belongs in the spanning tree. Alon et al. investigate the theoretical aspects of the above game and its connections to the *k-server problem* and *network design*.

Finally, there is a long line of work on security games [12] where many scenarios are modelled using graph theoretic problems [87,93,143,144].

5.1.3 The model and definitions

The game. A *Connected-Subgraph Defense (CSD) game* is defined by a graph $G = (V, E)$, a *defender*, $k \geq 1$ *attackers*, and a positive integer λ. Throughout the current chapter, λ is considered to be a *given* parameter of the game. A pure strategy for the defender is any induced connected subgraph H of G with λ vertices, which we term λ-*subgraph*. For any λ-subgraph H of G we denote $V(H)$ its set of vertices. Since $V(H)$ uniquely defines an induced subgraph of G, we will use the term λ-subgraph to denote either $V(H)$ or H. The *action set* of the defender is $D := \{V(H) | H \text{ is a } \lambda\text{-subgraph of } G\}$ and we will denote its cardinality by θ, i.e. $\theta := |D|$. For ease of presentation, we will also refer to D as $[\theta] := \{1, 2, \ldots, \theta\}$. A pure strategy for each of the attackers is any vertex of G. So, the action set of every attacker is V, the vertex set of G; we denote $n := |V|$ and we similarly refer to V also as $[n]$.

To play the game, the defender chooses a *defense (mixed) strategy*, i.e. a probability

distribution over her action set, and each attacker chooses an *attack (mixed) strategy*, i.e. a probability distribution over the vertices of G. We denote a strategy by $s := (s_1, \ldots, s_d) \in \Delta_d$, i.e. by the probability distribution over d enumerated pure strategies, where $\Delta_d := \{x_1, \ldots, x_d \geq 0 | \sum_{i=1}^{d} x_i = 1\}$ is the $(d-1)$-unit simplex. In a defense strategy $q \in \Delta_\theta$ each pure strategy $j \in [\theta]$ is assigned a probability q_j.

We say that a pure strategy of the defender, i.e. a specific λ-subgraph H of G, *covers* a vertex $v \in V$ if $v \in V(H)$. A defense strategy covers a vertex $v \in V$ if it assigns strictly positive probability to at least one λ-subgraph H of G which contains v.

Definition 11 (Vertex Probability)**.** *The* vertex probability p_i *of vertex* $i \in [n]$*, is the probability that* i *will be covered, formally* $p_i := \sum_{j \in [\theta]:\ i \in j} q_j$.

Payoffs and Strategy profiles. A *strategy profile* is a tuple of strategies $S = (q, t_1, \ldots, t_k)$, where q denotes the defender's strategy and t_j denotes the j-th attacker's strategy, $j \in [k]$. A strategy profile is pure if the support of every strategy has size one. The *payoff* of every attacker is 1 in any pure strategy profile where she does not choose a defended vertex, and 0 in all the rest. The payoff of the defender in a pure strategy profile where she defends $V(H)$, is the number of attackers that choose a vertex in $V(H)$. Under a strategy profile, the *expected payoff* of the defender is the expected number of attackers that she catches, which we call *defense value*, and the expected payoff of the attacker is the probability that she will not get caught. A *best response* strategy for a participant is a strategy that maximizes her expected payoff, given that the strategies of the rest of the participants are fixed. A *Nash equilibrium* is a strategy profile where all the participants are playing a best response strategy. In other words, neither the defender nor any of the attackers can increase their expected payoff by unilaterally changing their strategy.

Definition 12 (Defense Ratio)**.** *For a given graph* G *we define a measure of the quality of a strategy profile* S*, called* defense ratio of G *and denoted* $DR(G, S)$*, as the ratio of the total number of attackers* k *over the defense value.*

In this work we are only interested in the cases where S is an equilibrium. For a given graph, when in equilibrium, the defender's expected payoff is unique (due to Corollary 7 (a)) and achieves the *equilibrium defense ratio* $\mathrm{DR}(G, S^*)$, where S^* is an equilibrium. The defense strategy in S^* which achieves this defense ratio will be termed *best-defense strategy.*

Definition 13 (MaxMin Probability, p^*)**.** *We call* MaxMin Probability *of a graph G the maximum, over all defense strategies, minimum vertex probability in G, that is:*

$$p^*(G) := \max_{q \in \Delta_\theta} \min_{i \in [n]} p_i.$$

As we will show in Lemma 13, the equilibrium defense ratio of a graph G turns out to be $\mathrm{DR}(G, S^*) = 1/p^*(G)$.

Definition 14 (Price of Defense)**.** *The* Price of Defense, PoD*, for a given parameter λ of the game, is the worst defense ratio, over all graphs, achievable in equilibrium, that is:*

$$\mathrm{PoD}(\lambda) = \max_{G} \mathrm{DR}(G, S^*).$$

Definition 15 (Defense-Optimal Graph)**.** *For a given λ, a graph G^* that achieves the minimum equilibrium defense ratio over all graphs, i.e. $G^* \in \arg\min_G \mathrm{DR}(G, S^*)$, is called* defense-optimal graph.

In the following, for ease of presentation, whenever we refer to defense optimality, we implicitly assume that λ has a fixed value.

5.2 Nash Equilibria

In this section, we provide a characterization of Nash equilibria in CSD games, as well as important properties of their structure which prove useful for the development of our algorithms in the remainder of the Chapter.

Theorem 13 (Equilibrium characterization)**.** *For a given graph G, in any equilibrium with support $S \subseteq [\theta]$ of the defender and support $T_j \subseteq [n]$ of each attacker $j \in [k]$, the following conditions are necessary and sufficient:*

1. *$\min_{i \in [n]} p_i$ is maximized over all defense strategies, and*
2. *$\bigcup_{j \in [k]} T_j \subseteq V^*$, where $V^* := \{i \mid \min_{i \in [n]} p_i$ is maximized over all defense strategies$\}$, and*
3. *every $s \in S$ has the maximum expected total number of attackers on its vertices over all pure strategies.*

Proof. First we will prove that the conditions in the statement of the theorem hold in equilibrium, i.e. equilibrium is sufficient for the conditions to hold.

Condition 1. By definition, in an equilibrium the defender and each attacker have chosen a best response. Suppose that the defender has chosen some strategy $q = (q_1, q_2, \ldots, q_\theta)$ over her action set $[\theta]$, and we will consider this strategy to be a vector variable for now. Given q, each vertex $i \in [n]$ has a vertex probability p_i. Now consider the minimum vertex probability $p' := \min_{i\in[n]} p_i$, and the set $V' \subseteq V$ consisting of the vertices with vertex probability p', i.e. $V' := \arg\min_{i\in[n]} p_i$. Since an attacker plays a best response, her support will be a subset of V'; otherwise, if she assigns probability $t_v > 0$ on a vertex $v \notin V'$ (with $p_v > p'$) her expected payoff (see quantity (5.2)) can be strictly increased by choosing to move all of t_v to another vertex $u \in V$, thus increasing her expected payoff by $t_u(p_v - p')$. Therefore, every attacker's support will be a subset of V'.

Now suppose that there are $k \geq 1$ attackers and let us denote the set of attackers by $[k]$. We will denote by t_{ji} the probability that the strategy of attacker $j \in [k]$ has assigned on vertex $i \in [n]$. The expected payoff of the defender is:

$$\sum_{i\in[n]} \left(p_i \sum_{j\in[k]} t_{ji} \right). \tag{5.1}$$

Since as we argued above, in an equilibrium, each attacker's strategy has support that is subset of V', the expected payoff of the defender will be

$$\sum_{i\in V'} \left(p_i \sum_{j\in[k]} t_{ji} \right) + \sum_{i\in V\setminus V'} \left(p_i \sum_{j\in[k]} t_{ji} \right) = p' \cdot \sum_{i\in V'} \left(\sum_{j\in[k]} t_{ji} \right) = p' \cdot \sum_{j\in[k]} \left(\sum_{i\in V'} t_{ji} \right) = p' \cdot k,$$

where the first equality is due to the fact that $p_i = p' \ \forall i \in V'$ and $t_{ji} = 0 \ \forall i \in V \setminus V'$, and the last equality is due to the fact that the support of any strategy $t_j = (t_{j1}, \ldots, t_{ji})$ of an attacker $j \in [k]$ is a subset of V'. In an equilibrium, the defender also plays a best response, i.e. she maximizes her expected utility. Therefore, given the above quantity, the defender in an equilibrium has expected utility $\max_{q\in\Delta_\theta} p' \cdot k$, and Condition 1 of the theorem's statement is satisfied.

Condition 2. The proof is by contradiction. Assume an equilibrium profile where the defender has strategy $q = (q_1, \ldots, q_\theta)$ and there is an attacker, a, with strategy $t = (t_1, \ldots, t_n)$ whose support includes vertex $v \in [n]$ with $p_v > p'$, where $p' := \min_{i\in[n]} p_i$.

Then a's expected payoff is

$$\sum_{\substack{i\in V\\ i\neq v}} t_i(1-p_i) + t_v(1-p_v). \tag{5.2}$$

However, a can increase her expected payoff by moving all her probability t_v to a vertex v' for which $p_{v'} = p'$, which contradicts the equilibrium assumption.

Condition 3. The proof is by contradiction. Suppose that in an equilibrium the defender has strategy $q^* \in \Delta_\theta$, where $\text{supp}(q^*) := S$. According to Condition 1, this strategy achieves $p^*(G)$, and let us define the set

$$V^* := \{i \in [n] \mid \min_{i\in[n]} p_i \text{ is maximized over all defense strategies}\}$$

. We denote by N_i the random variable that indicates the number of attackers on vertex $i \in [n]$, under the strategy profile determined by the strategy of the defender and each attacker. The expected utility of the defender is as in (5.1), or equivalently, $\sum_{i\in[n]} (p_i \cdot \mathbb{E}[N_i])$. Since, as argued above, in an equilibrium each attacker has support in V^*, the defender's expected payoff is in fact $p^* \cdot \sum_{i\in V^*} \mathbb{E}[N_i]$.

For the sake of contradiction, suppose that for the expected total number of attackers on two different pure defense strategies $s_1 \in S$ and $s_2 \in [\theta]$ it holds that $\mathbb{E}\left[\sum_{i\in s_1} N_i\right] < \mathbb{E}\left[\sum_{j\in s_2} N_j\right]$, and equivalently $\mathbb{E}\left[\sum_{i\in s_1\setminus s_2} N_i\right] < \mathbb{E}\left[\sum_{j\in s_2\setminus s_1} N_j\right]$. Then, the expected payoff of the defender can be strictly increased if she chooses a strategy $q' = (q'_1, \ldots, q'_\theta)$ where $q'_{s_1} = 0$ and $q'_{s_2} = q^*_{s_2} + q^*_{s_1}$. In particular, when the defender plays q^* her expected payoff is

$$U^* = p^* \cdot \mathbb{E}\left[\sum_{i\in V\setminus(s_1\cup s_2)} N_i\right] + p^* \cdot \mathbb{E}\left[\sum_{j\in s_1\cap s_2} N_j\right] + p^* \cdot \mathbb{E}\left[\sum_{l\in s_2\setminus s_1} N_l\right] + p^* \cdot \mathbb{E}\left[\sum_{r\in s_1\setminus s_2} N_r\right],$$

whereas when she plays q' it is

$$\begin{aligned} U' &= p^* \cdot \mathbb{E}\left[\sum_{i \in V \setminus (s_1 \cup s_2)} N_i\right] + p^* \cdot \mathbb{E}\left[\sum_{j \in s_1 \cap s_2} N_j\right] + (p^* + q^*_{s_1}) \cdot \mathbb{E}\left[\sum_{l \in s_2 \setminus s_1} N_l\right] \\ &\qquad + (p^* - q^*_{s_1}) \cdot \mathbb{E}\left[\sum_{r \in s_1 \setminus s_2} N_r\right] \\ &= U^* + q^*_{s_1} \cdot \left(\mathbb{E}\left[\sum_{l \in s_2 \setminus s_1} N_l\right] - \mathbb{E}\left[\sum_{r \in s_1 \setminus s_2} N_r\right]\right) \\ &> U^*, \end{aligned}$$

which contradicts the equilibrium assumption. Therefore, for every pure defense strategy $s_1 \in S$ it holds that $\mathbb{E}\left[\sum_{i \in s_1} N_i\right] \geq \mathbb{E}\left[\sum_{j \in s_2} N_j\right]$ for every $s_2 \in [\theta]$.

Now we will prove that equilibrium is necessary for the three conditions of the statement to hold. Suppose that all conditions hold and $p^*(G)$ is achieved for the defense strategy $q = (q_1, \ldots, q_\theta)$. We will show that the defender and each attacker play a best response.

Consider an attacker $j \in [k]$ with strategy $t = (t_1, \ldots, t_n)$ and support $T_j \subseteq V^*$ according to Condition 2. Her expected payoff is

$$\sum_{i \in T_j} t_i(1 - p^*) = 1 - p^*.$$

It suffices to consider unilateral deviations of j to pure strategies. Any pure strategy $i' \in T_j$ gives her expected payoff $1 - p^*$, since $p_{i'} = p^*$ (because $T_j \subseteq V^*$). Any pure strategy $i' \in V^* \setminus T_j$ also gives her expected payoff $1 - p^*$ for the same reason. Finally, any pure strategy $i' \in V \setminus V^*$ gives her expected payoff $1 - p_{i'} < 1 - p^*$ by the definition of V^*. Therefore every attacker plays a best response.

Now consider the defender with strategy $q = (q_1, \ldots, q_\theta)$ and support $S \subseteq [\theta]$. According to Condition 1 of the theorem's statement, q results to vertices of G having vertex probability p^*. By Condition 3, for any pure defense strategy $s_1 \in S$ it holds that $\mathbb{E}\left[\sum_{i \in s_1} N_i\right] \geq \mathbb{E}\left[\sum_{j \in s_2} N_j\right]$ for every $s_2 \in [\theta]$, and let us denote $N_{max} := \mathbb{E}\left[\sum_{i \in s_1} N_i\right]$. Now consider a unilateral deviation $q' = (q'_1, \ldots, q'_\theta)$ of the defender. Her expected payoff

is

$$\begin{aligned} U(q') &= \sum_{j\in[\theta]} \left(q'_j \mathbb{E}\left[\sum_{i\in j} N_i\right]\right) \\ &\leq \sum_{j\in[\theta]} q'_j N_{max} \\ &= N_{max} \\ &= \sum_{j\in S} \left(q_j \mathbb{E}\left[\sum_{i\in j} N_i\right]\right) \\ &= U(q), \end{aligned}$$

where the penultimate equation holds due to the fact that $\sum_{j\in S} q_j = 1$. Therefore, q is a best response for the defender, and the three conditions of the theorem's statement imply a strategy profile that is an equilibrium. □

Lemma 13. *For any given graph G, the equilibrium defense ratio is* $\mathrm{DR}(G, S^*) = \frac{1}{p^*(G)}$, *where* $p^*(G) := \max_{q\in\Delta_\theta} \min_{i\in[n]} p_i$ *and S^* is an equilibrium.*

Proof. By Theorem 13, in an equilibrium, every attacker will have in her support only vertices that are defended with probability exactly $p^*(G)$. Therefore, the expected number of attackers that the defender catches is $p^*(G) \cdot k$. By definition of the defense ratio, $\mathrm{DR}(G, S^*) = \frac{k}{p^*(G)\cdot k} = \frac{1}{p^*(G)}$. □

Corollary 7. *The following hold:*

(a) For a given graph G, in any equilibrium, the expected payoff of the defender and each attacker is unique.

(b) For a given graph G, in any equilibrium with support $S \subseteq [\theta]$ of the defender, for every $s \in S$ there exists a vertex $v \in s$ such that $p_v = p^*(G)$.

(c) In any CSD game on a graph G, the problem of finding the equilibrium defense ratio (or equivalently, $p^*(G)$) for $k \geq 2$ attackers reduces to the same problem in the game with $k = 1$ attacker, which is a two-player constant-sum game.

Proof. *(a)* By Theorem 13, in an equilibrium the defender chooses a strategy that induces probability $p^*(G)$ to some vertex of G (Condition 1). Also, each of the attackers

has in her support T only vertices with vertex probability $p^*(G)$. Therefore, all attackers attack only such vertices and the expected payoff of the defender is $k \cdot p^*(G)$. Consider also an attacker with strategy $t = (t_1, t_2, \ldots, t_n)$. Her expected payoff is $\sum_{i \in [n]} t_i(1 - p_i)$, where p_i is the vertex probability of vertex i. This value is equal to $\sum_{i \in T} t_i(1 - p^*(G)) = 1 - p^*(G)$. Since $p^*(G)$ is unique for a graph G, the expected payoffs of the defender and each attacker is unique.

(b) The proof is by contradiction. Consider an equilibrium where the defender's strategy is $q \in [\theta]$ with support S, and there exists a pure strategy $s \in S$ for which every vertex $v \in s$ has $p_v > p^*(G)$. By Condition 2 of Theorem 13, no attacker has in her support a vertex in s. Therefore, the defender can strictly increase her expected payoff by moving all her probability $q_s > 0$ from s to some other pure strategy s' that contains a vertex which is in the support of some attacker.

(c) Observe that for any given graph G, the quantity $p^*(G)$, by definition, only depends on the graph and not the number of attackers k. That is, $p^*(G)$ is the same for every $k \geq 1$. Lemma 13 states that in any equilibrium S^*, it is $\mathrm{DR}(G, S^*) = \frac{1}{p^*(G)}$, therefore the defense ratio in an equilibrium does not depend on k. This means that when we are given G and we are interested in the equilibrium defense ratio, we might as well consider the game with the single defender and a single attacker. By definition of the game (see Section 5.1.3) the latter is a two-player constant-sum game.

□

The following corollary implies that coordination (resp. individual selfishness) of the attackers cannot increase the attackers' (resp. defender's) expected payoff in equilibrium.

Corollary 8. *Every equilibrium with uncoordinated attackers (i.e. as described in Section 5.1.3) is an equilibrium with coordinated (i.e. centrally controlled) attackers, and vice versa.*

Proof. Let q^* be a best-defense strategy for the defender. Then, in any best response of any attacker, coordinated or not, every attacker plays only pure strategies that yield maximum payoff against q^*; i.e. they play only strategies that are defended with probability $p^*(G)$. If this was not the case, either an uncoordinated attacker could increase her payoff by unilaterally changing her strategy, or the "coordinator" could increase the payoff the attackers collectively get by dictating all the attackers to play vertices that are covered with probability $p^*(G)$.

So, assume that we have an equilibrium in the uncoordinated case. This is an equilibrium for the coordinated case as well: according to Theorem 13, all attackers play vertices that are defended with probability $p^*(G)$ and thus the expected collective payoff of the attackers cannot be increased, and furthermore the expected total number of attackers on the vertices of a pure strategy that is in the support of the defender is maximized over all pure defense strategies, so no unilateral deviation of the defender can increase her expected payoff.

Conversely, in any equilibrium in the coordinated setting the "coordinator" dictates all the attackers to attack vertices that are covered with probability $p^*(G)$, satisfying Conditions 1,2 of Theorem 13. Also in the equilibrium of the coordinated setting, similarly to Condition 3 of Theorem 13, the "coordinator" will have placed the attackers in a way such that the vertices of any pure defense strategy in the support have maximum expected total number of attackers over all pure defense strategies; otherwise the defender can increase her expected payoff by neglecting a pure strategy with smaller than maximum expected total number of attackers, and move the probability assigned on that pure strategy to another one that has maximum expected total number of attackers. By Theorem 13, this is an equilibrium also for the uncoordinated setting. □

The following theorem provides an algorithm for computing an equilibrium for any CSD game, whose running time is polynomial in n when $\lambda = c$ or $\lambda = n - c$, where c is a constant natural number.

Theorem 14. *For any given graph G and parameter λ, there is an algorithm that computes $p^*(G)$ and also finds an equilibrium in time polynomial in $\binom{n}{\lambda}$.*

Proof. Given a graph G, the number of attackers $k \geq 1$, and some $\lambda \in \{1, 2, \ldots, n\}$, the action set D of the defender is constructed by the vertex sets of at most $\binom{n}{\lambda}$ λ-subgraphs, so for D's cardinality θ it holds that $\theta \leq \binom{n}{\lambda}$. Consider now the mixed strategy $q \in \Delta_\theta$ of the defender, where each pure strategy $j \in [\theta]$ is assigned probability q_j. Consider also the vertex probability p_i for each vertex $i \in [n]$. According to Corollary 7 (a) and (c), the unique $p^*(G)$ in the case of a single attacker can be used to derive an equilibrium for the case of $k \geq 2$ attackers. Therefore, we will find $p^*(G)$ for a single attacker, find an equilibrium for that case, and then extend this equilibrium to one in the case of $k \geq 2$ attackers. In more detail, after we find the defense strategy q^* that maximizes $\min_{i \in [n]} p_i$ (Condition 1 of Theorem 13), i.e. yields $p^*(G)$ on the set $V^* := \arg\max_{q \in \Delta_\theta} \min_{i \in [n]} p_i$,

an equilibrium is achieved if the single attacker assigns probability $1/|V^*|$ to each vertex of V^*; that is because all conditions of Theorem 13 are satisfied. Then, an equilibrium for $k \geq 2$ is achieved if every attacker plays the same strategy as the single attacker; that is because again all conditions of Theorem 13 are satisfied.

The crucial observation that allows us to design such an algorithm is that we can compute $p^*(G)$ via a Linear Program which has $O\left(\binom{n}{\lambda}\right)$ many variables and $O(n)$ constraints, and therefore its running time is in the worst case polynomial in $\binom{n}{\lambda}$, for $\lambda \in \{2, 3, \ldots, n-1\}$. For the trivial cases $\lambda = 1$ and $\lambda = n$, $D = \{\{i\} | i \in V\}$ and $D = V$ respectively, therefore $p^*(G) = 1/n$ and $p^*(G) = 1$ respectively. So in the rest of the proof we will imply that $\lambda \in \{2, 3 \ldots, n-1\}$. It remains to show how $p^*(G)$ is computed.

Let us denote $p^* := p^*(G) := \max_{q \in \Delta_\theta} \min_{i \in [n]} p_i$. The computation of p^* can be done as follows: First, consider each of the $\binom{n}{\lambda}$ subsets of V of size λ, and find if it is a proper λ-subgraphs of G (i.e. connected); this can be done by running a Depth (or Breadth) First Search algorithm for each subset of size λ. If it is not, then continue with the next subset. If it is, we consider it in the action set $[\theta]$, and assign to it a variable q_j which stands for its assigned probability in a general defense strategy. Now, by definition, for some vertex $i \in [n]$, $p_i = \sum_{\substack{j \in [\theta] \\ i \in j}} q_j$. Therefore, we will consider only pure strategies j which are λ-subgraphs to create the p_i's. To compute the minimum p_i over all i's we introduce the variable p' and write the following set of n inequalities as a constraint in our Linear Program:

$$\sum_{\substack{j \in [\theta] \\ i \in j}} q_j \geq p' \quad , \text{ for } i \in \{1, 2, \ldots, n\}.$$

The variable constraints are $p', q_1, q_2, \ldots, q_\theta \geq 0$ and also $\sum_{j=1}^{\theta} q_j = 1$, and all of the aforementioned constraints can be written in canonical form by applying standard transformations. Finally, the objective function of the Linear Program is variable p' and we require its maximization, which is the value p^*.

□

5.2.1 Connections to other types of games

Although CSD games are defined as a normal form game with $k+1$ players, we can observe that they are a special case of other well-studied types of games: polymatrix games and

Stackelberg games.

A polymatrix game is defined by a graph where every vertex represents a player and every edge represents a two-player game played by the endpoints of the edge. Every player has the same set of pure strategies in every game he is involved and to play the game he plays the same (mixed) strategy in every game. The payoff of every player is the sum they get from every two-player game they participate in. In a CSD game we observe the following. Firstly, the payoff of every attacker depends only on the strategy the defender plays, thus every attacker is involved only in one two-player game. In addition, all the attackers have the same set of pure strategies and they share the same payoff matrix. Similarly, the payoff the defender gets from catching an attacker depends only on the strategy the defender and this specific attacker chose. Hence, the payoff of the defender can be decomposed into a sum of payoffs from k two-player games. So, a CSD game can be seen as a polymatrix game where the underlying graph is a star with k leaves that correspond to the attackers and the defender is the center of the star. Although many-player polymatrix games have exponentially smaller representation size compared to the equivalent normal-form representation, we should note that this polymatrix game is of exponential size in the worst case since the defender can have exponential in n pure strategies to choose from.

A Stackelberg game is an extensive form two-player game. In the first round, one of the players commits to a (mixed) strategy. In the second round, the other player chooses a best response against the committed strategy of her opponent. In a Stackelberg equilibrium the first player is playing a strategy that maximizes her expected payoff, given that the second player plays a best response (mixed strategy). The MaxMin probability $p^*(G)$ for a CSD game on a graph G corresponds to a Stackelberg equilibrium. By Corollary 7(c), any CSD game with $k \geq 1$ attackers has the same p^* as that of the case with $k = 1$. Furthermore, as in a Stackelberg game, in the CSD game with $k = 1$ the defender chooses a mixed strategy that maximizes her expected payoff, given that the attacker plays a best response (mixed strategy). Therefore, when we are interested in the defense-ratio in equilibrium of a CSD game for some arbitrary $k \geq 1$, finding a Stackelberg equilibrium of the corresponding CSD game with $k = 1$ suffices.

5.3 Defense-Optimal Graphs

We now focus our attention on defense-optimal graphs. We first characterize defense-optimal graphs with respect to the MaxMin probability p^* and then use this characterization

to analyze more specific classes of graphs like Hamiltonian graphs and tree graphs. We begin by an exact computation of the equilibrium defense ratio of any defense-optimal graph.

Theorem 15. *In any defense-optimal graph G, we have that $DR(G, S^*) = \frac{n}{\lambda}$.*

Proof. First we will show that $\frac{n}{\lambda}$ is a lower bound on the equilibrium defense ratio $\mathrm{DR}(G, S^*)$ and then prove that it is tight. According to Lemma 13, a lower bound on $\mathrm{DR}(G, S^*)$ can be found by equivalently founding an upper bound on $p^*(G)$ over all graphs G with n vertices. Let us show that $p^*(G) \leq \frac{\lambda}{n}$ for every G.

Suppose there is a graph G' such that $p^*(G') > \frac{\lambda}{n}$, and let us focus only on G'. Suppose also that the defender has an action set $[\theta]$ on G'. Fix the strategy $q = (q_1, \ldots, q_\theta) \in \Delta_\theta$ that achieves $p^*(G')$. Then, by definition of $p^*(G')$, for the vertex probabilities p_i it holds that $p_i > \frac{\lambda}{n}$ for all $i \in [n]$. Therefore, it is

$$\sum_{i=1}^{n} p_i > \lambda. \tag{5.3}$$

Also, by definition of a defense strategy, if X denotes the random variable corresponding to the number of vertices that the defender covers, then:

$$\mathbb{E}[X] = \sum_{j=1}^{\theta} q_j \cdot |L_j| = \lambda \qquad (\text{where } L_j \text{ is a } \lambda\text{-subgraph of } G, \text{ hence } |L_j| = \lambda \quad \forall j \in [\theta]). \tag{5.4}$$

Let us introduce the indicator variables X_{ij}, $i \in [n]$, $j \in [\theta]$ with value 1 if vertex $i \in L_j$, and 0 otherwise. Then,

$$\begin{aligned} \mathbb{E}[X] &= \sum_{j=1}^{\theta} q_j \sum_{i=1}^{n} X_{ij} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{\theta} q_j X_{ij} \\ &= \sum_{i=1}^{n} p_i \\ &> \lambda \qquad (\text{by inequality (5.3)}), \end{aligned} \tag{5.5}$$

which contradicts (5.4).

It remains to show that the lower bound $\frac{n}{\lambda}$ on $\mathrm{DR}(G, S^*)$ is tight. This is easy to do by showing that $\frac{\lambda}{n}$ is a tight upper bound on $p^*(G)$: any Hamiltonian graph has $p^*(G) = \frac{\lambda}{n}$ as we show in Observation 1. □

As an intermediate corollary of Theorem 15 we get the following characterization of defense-optimal graphs.

Corollary 9. *A graph G is defense-optimal if and only if all of its vertices are defended with probability $\frac{\lambda}{n}$.*

Proof. Necessity of defense-optimality is trivial: every vertex has vertex probability $\frac{\lambda}{n}$, therefore $p^*(G) = \frac{\lambda}{n}$, so by Theorem 15 the graph is defense-optimal.

Sufficiency of defense-optimality is also easy to see using the equations (5.4), (5.5) of the proof of Theorem 15. Suppose that the graph is defense-optimal and consider an equilibrium where the defense strategy is $q = (q_1, \ldots, q_\theta)$. Then the sum of vertex probabilities is $\sum_{i=1}^{n} p_i = \lambda$ according to the aforementioned equations. Therefore, if there exists a vertex v with vertex probability $p_v > \frac{\lambda}{n}$ then there is another vertex u with probability $p_u < \frac{\lambda}{n}$. This means that $p^*(G) < \frac{\lambda}{n}$, and as a result the graph is not defense-optimal which contradicts our assumption. □

Someone may wonder whether Corollary 9 can be further exploited to prove that, in general, there are best-defense strategies in defense-optimal graphs are uniform, i.e. every pure strategy s in the support S of the defender is assigned probability $1/|S|$. However, as we demonstrate in Figure 5.1 this is not the case. On the other hand, this claim is true for Hamiltonian graphs and tree graphs.

Observation 1. *All Hamiltonian graphs are defense-optimal.*

Proof. Consider an arbitrary Hamiltonian graph G with n vertices. We will show that the graph can achieve vertex probability $p_i = \frac{\lambda}{n}$ for every $i \in [n]$, thus by Corollary 9 it is defense-optimal. Consider a Hamiltonian cycle of G and let us denote it by H. In the rest of the proof H will be the graph under study. Now consider the whole action set D of the defender, i.e. every path on H starting from a vertex i going clockwise and ending at vertex $i + \lambda - 1$. Observe that there are only n such paths, therefore $\theta := |D| = n$. By assigning probability $\frac{1}{n}$ to each pure strategy $j \in [\theta]$, since each vertex is in exactly λ pure strategies, each vertex $i \in [n]$ has vertex probability $p_i = \lambda \cdot \frac{1}{\theta} = \frac{\lambda}{n}$. □

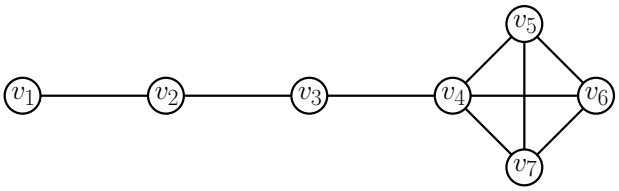


Figure 5.1: An example of a defense-optimal graph G with no uniform best-defense strategy. Here $n = 7$, $\lambda = 3$ and $p^*(G) = 3/7$ is achievable by assigning probability $3/7$ to pure strategy $\{v_1, v_2, v_3\}$ and probability $1/7$ to each of the pure strategies $\{v_4, v_5, v_6\}$, $\{v_4, v_5, v_7\}$, $\{v_4, v_6, v_7\}$, $\{v_5, v_6, v_7\}$, so the graph is defense optimal. Suppose that there is a uniform best-defense strategy for the graph, with support of size r. Observe that v_1 cannot participate in more than one pure defense strategies, so in the uniform defense strategy the vertex probability p_{v_1} has to be $1/r$ (by definition of uniformity), but it also has to be $3/7$ due to Corollary 9. Since $r \in \mathbb{N}$, this is a contradiction, and there is no uniform best-defense strategy for G.

5.3.1 Tree graphs

In this section we focus on the case where the graph is a tree. We first further refine the characterization of defense-optimal graphs for trees. Then, we utilise this characterization to derive a polynomial-time algorithm that decides in polynomial time whether a given tree is defense-optimal, and if that is the case, it constructs in polynomial time a defense-optimal strategy for it. On the other hand, in the case where the tree is not defense-optimal, we show that it is NP-hard to compute a best-defense strategy for it, namely it is NP-hard to compute $p^*(G)$. We first provide Lemma 14 which will be used in our polynomial-time algorithm for checking defense-optimality on trees. Henceforth, we write that a graph is *covered* by a defense strategy if every vertex of the graph is covered by a λ-subgraph that is in the support of the defense strategy.

Lemma 14. *A tree T is defense-optimal if and only if T can be decomposed into $\frac{n}{\lambda}$ disjoint λ-subgraphs.*

Proof. **(⇒)** Let T be defense-optimal. We will show that the support of any best-defense strategy on T must comprise of pure strategies that are disjoint λ-subgraphs which altogether cover every $v \in V$. Since those are disjoint and cover T, it follows that their number is $\frac{n}{\lambda}$ in total.

If $\lambda = 1$ then the above trivially holds. Assume that $\lambda \geq 2$ and consider a best-defense

strategy on T whose support comprises of a collection $\mathcal{L}$ of λ-subgraphs.

Let $u \in V$ be a leaf of T and let $v \in V$ be its parent. Any λ-subgraph in $\mathcal{L}$ covering u must also cover v, since $\lambda \geq 2$. Also, any λ-subgraph in $\mathcal{L}$ covering v must also cover u, otherwise p_v would be greater than p_u. Now, consider the neighbors of v. For those of them that are leaves, the same must hold as holds for u, namely v and its leaf-children must all be covered by the same exact λ-subgraph(s).

Consider the case where there is a leaf $u \in V$, such that a *single* λ-subgraph contains u, its parent v, and all the other leaf-children of v (and, possibly, other vertices connected to v). Then we can remove this λ-subgraph from $\mathcal{L}$ and the corresponding tree from T. This leaves the remainder of T being a forest comprising of trees $T_1, \ldots, T_x$, each of which has a (best-) defense strategy comprising of the corresponding subset of (the remainder of) $\mathcal{L}$ on T_i. Notice that it must be the case that every tree T_i, $i = 1, 2, \ldots, x$, has size at least λ (otherwise the initial collection $\mathcal{L}$ would not have covered T). So, if there is always a leaf u in some tree of the forest, such that a *single* λ-subgraph contains u, its parent v, and all the other leaf-children of v (and, possibly, other vertices connected to v), we can proceed in the same fashion for each of the T_i's, always removing a λ-subgraph from $\mathcal{L}$, and the corresponding vertices from T, until we end up with an empty tree. This means that $\mathcal{L}$ was indeed a collection of disjoint λ-subgraphs covering T.

However, assume for the sake of contradiction that at some "iteration" the assumption does not hold, namely assume that there is a tree in the forest with no leaf u, such that a single λ-subgraph contains u, its parent v, and all the other leaf-children of v (and, possibly, other vertices connected to v). This means that there are (at least) two λ-subgraphs in $\mathcal{L}$, namely L_1, L_2, that cover u. Due to our initial observations, u, together with its parent v and all of v's leaf-children are contained in both L_1 and L_2. Since those are different λ-subgraphs, there is a vertex z in the tree which belongs to L_2 but does not belong to L_1. Since $p_z = p_v$ (due to the fact that $\mathcal{L}$ is the support of the defense-optimal strategy and Corollary 9), it must hold that there is a different λ-subgraph, L_3, which covers z but does not cover v or any of its leaf-children. If L_3 also covers a vertex in $L_1 \setminus L_2$[1], then there is a cycle in the tree which is a contradiction. So L_3 must not cover vertices in $L_1 \setminus L_2$. Since $_3$ is different to L_2, there must be a vertex z' in the tree which belongs in L_3 but not in L_2 (also not in L_1). Since $p_{z'} = p_z$ (due to the fact that $\mathcal{L}$ is the support of the defense-optimal strategy and Corollary 9), it must hold that there is a different λ-subgraph,

[1]We use $L_i \setminus L_j$ for some λ-subgraphs L_i, L_j to denote the set of vertices which are contained in L_i but not in L_j.

L_4, which covers z' but does not cover z or any of the vertices in L_2. Similarly to before, if L_4 covers a vertex in $L_1 \setminus L_2$, then there is a cycle in the tree which is a contradiction. So L_4 must not cover vertices in L_1 or in L_2.

Proceeding in the same way, we result in contradiction since the tree has finite number of vertices and there will need to be an overlap in coverage of some L_j with some L_i, $j > i + 1$, which would mean that there is a cycle in the tree.

Therefore, there cannot be any overlaps between the λ-subgraphs of $\mathcal{L}$, meaning that $\mathcal{L}$ comprises of $\frac{n}{\lambda}$ disjoint λ-subgraphs which altogether cover T.

($\Leftarrow$) Let $\mathcal{L} = \{L_1, \ldots, L_{\frac{n}{\lambda}}\}$ be a collection of $\frac{n}{\lambda}$ disjoint λ-subgraphs that altogether cover T. Let the defender play each L_i, $i \in \{1, \ldots, \frac{n}{\lambda}\}$, equiprobably, that is, with probability $1/\binom{n}{\lambda} = \frac{\lambda}{n}$. Then every vertex $v \in V$ is covered with probability $p_v = \frac{\lambda}{n} = p^*(G)$, meaning that T is defense-optimal. □

With Lemma 14 in hand we can derive a polynomial-time algorithm that decides if a tree is defense-optimal, and if it is, to produce a best-defense strategy.

Theorem 16. *There exists a polynomial-time algorithm that decides whether a tree is defense-optimal, and if it is, it outputs a best-defense strategy.*

Proof. The algorithm works as follows. Initially, there is a pointer associated with a counter in every leaf of the tree T that moves "upwards" towards an arbitrary root of the tree. For every move of the pointer the corresponding counter increases by one. The pointer moves until one of the following happens: either the counter is equal to λ, or it reaches a vertex with degree greater of equal to 3 where it "stalls". In the case where the counter is equal to λ, we create a λ-subgraph of T, we delete this λ-subgraph from the tree, we move the pointer one position upwards, and we reset the counter back to zero. If a pointer stalls at a vertex of degree $d \geq 3$, it waits until all $d - 1$ pointers reach this vertex. Then, all these pointers are merged to a single one and a new counter is created whose value is equal to the sum of the counters of all d pointers. If this sum is more than λ, then the algorithm returns that the graph is not defense-optimal. If this sum is less than or equal to λ, then we proceed as if there was initially only this pointer with its counter; if the new counter is equal to λ, then we create a λ-subgraph of T and reset the counter to 0; else the pointer moves upwards and the counter increases by one. To see why the algorithm requires polynomial time, observe that we need at most n pointers and n counters and in addition every pointer moves at most n times.

We now argue about the correctness of the algorithm described above. Clearly, if the algorithm does not output that the tree is not defense-optimal, it means that it partitioned T into λ-subgraphs. So, from Lemma 14 we get that T is defense-optimal and the uniform probability distribution over the produced partition covers every vertex with probability $\frac{\lambda}{n}$. It remains to argue that when the algorithm outputs that the graph is not defense-optimal, this is indeed the case. Consider the case where we delete a λ-subgraph of the (remaining) tree. Observe that the λ-subgraph our algorithm deleted should be uniquely covered by this λ-subgraph in any best-defense strategy; any other λ-subgraph would overlap with some other λ-subgraph. Hence, the deletion of such a λ-subgraph was not a "wrong" move of our algorithm and the remaining tree is defense-optimal if and only if the tree before the deletion was defense-optimal. This means that any deletion that occurred by our algorithm did not make the remaining graph non defense-optimal. So, consider the case where after a merge that occurred at vertex v we get that the new counter is $c > \lambda$. Then, we can deduce that all the subtrees rooted at v associated with the counters have strictly less than λ vertices. Hence, in order to cover all the $c > \lambda$ vertices using λ-subgraphs, at least two of these λ-subgraphs cover vertex v. Hence, the condition of Lemma 14 is violated. But since every step of our algorithm so far was correct, it means that v cannot be covered only by one λ-subgraph. Hence, our algorithm correctly outputs that the tree is not defense-optimal. □

In Theorem 16 we showed that it is easy to decide whether a tree is defense-optimal and if this is the case, it is easy to find a best-defense strategy for it. Now we prove that if a tree is not defense-optimal, then it is NP-hard to compute $p^*(G)$. Note that the problem of computing $p^*(G)$ reduces to the problem of finding a best-defense strategy for graph G. Therefore finding a best-defense strategy is also NP-hard.

Theorem 17. *Computing $p^*(G)$ in* CSD *games is* NP*-hard, even if the graph G is a tree. Consequently, finding a best-defense strategy is* NP*-hard.*

Proof. We will prove the theorem by reducing from 3-Partition. In an instance of 3-Partition we are given a multiset with n positive integers $a_1, a_2, \ldots, a_n$ where $n = 3m$ for some $m \in \mathbb{N}_{>0}$ and we ask whether it can be partitioned into m triplets $S_1, S_2, \ldots, S_m$ such that the sum of the numbers in each subset is equal. Let $s = \sum_{i=1}^{n} a_i$. Observe then that the problem is equivalent to asking whether there is a partition of the integers to m triplets such that the numbers in every triplet sum up to $\frac{s}{m}$. Without loss of generality

we can assume that $a_i < \frac{s}{m}$ for every $i \in [n]$; if this was not the case, the problem could be trivially answered. So, given an instance of 3-PARTITION, we create a tree $G = (V, E)$ with $s + 1$ vertices and $\lambda = \frac{s}{m} + 1$. The tree is created as follows. For every integer a_i, we create a path with a_i vertices. In addition, we create the vertex v_0 and connect it to one of the two ends of each path. We will ask whether $p^*(G) \geq \frac{1}{m}$.

Firstly, assume that the given instance of 3-PARTITION is satisfiable. Then, given S_j we create a $(\frac{s}{m} + 1)$-subgraph of G as follows. If $a_i \in S_j$, then we add the corresponding path of G to the subgraph. Finally, we add vertex v_0 in our $(\frac{s}{m} + 1)$-subgraph and the resulting subgraph is connected (by the construction of G). Since the sum of a_i's equals $\frac{s}{m}$, the constructed subgraph has $\frac{s}{m} + 1$ vertices. If we assign probability $\frac{1}{m}$ to every $(\frac{s}{m} + 1)$-subgraph we get that $p_v \geq \frac{1}{m}$ for every $v \in V$.

To prove the other direction, assume that $p^*(G) \geq \frac{1}{m}$ and observe the following. Firstly, since as we argued it is $a_i < \frac{s}{m}$ for every $i \in [n]$, it holds that every $(\frac{s}{m} + 1)$-subgraph of G contains vertex v_0. Thus, $p_{v_0} = 1$ and $\sum_{v \neq v_0} p_v \geq \frac{s}{m}$, since there are s vertices other than v_0 and for each one of them holds that $p_v \geq \frac{1}{m}$. In addition, observe that $\sum_{v \in V} p_v = \lambda = \frac{s}{m} + 1$. Hence, we get that $p_v = p^*(G) = \frac{1}{m}$ for every vertex $v \neq v_0$. In addition, observe that every pure defense strategy that covers a leaf of this tree, covers all the vertices of the branch. Hence, for every branch of the tree, all its vertices are covered by the same set of pure strategies; if a vertex u that is closer to v_0 is covered by one strategy that does not cover the whole branch, then the leaf u' of the branch is covered with probability less than u. So, in order for $p_v = p^*(G) = \frac{1}{m}$ for every $v \neq v_0$, it means that there exist a $(\frac{s}{m} + 1)$-subgraph that *exactly* covers a subset of the paths; this means that if a $(\frac{s}{m} + 1)$-subgraph covers a vertex in a path, then it covers every vertex of the path. Hence, by the construction of the graph, we get that this $(\frac{s}{m} + 1)$-subgraph of G corresponds to a subset of integers in the 3-PARTITION instance that sum up to $\frac{s}{m}$. Since, 3-PARTITION is NP-hard, we get that computing $p^*(G)$ is NP-hard. Also, since finding a best-defense strategy is at least as hard, we conclude it is NP-hard. □

5.3.2 General graphs

We conjecture that contrary to checking defense-optimality of tree graphs and constructing a corresponding defense-optimal strategy in polynomial time, it is NP-hard to even decide whether a given (general) graph is defense-optimal.

Conjecture 1. *It is* NP*-hard to decide whether a graph is defense-optimal.*

5.4 Approximation Algorithm for $p^*(G)$

We showed in the previous section that, given a graph G, it is NP-hard to compute $p^*(G)$, and consequently, NP-hard to find a best-defense strategy. We also presented in Theorem 14 an algorithm for computing the exact value $p^*(G)$ of a given graph G (and therefore its best defense ratio), but this algorithm has running time polynomial in the size of the input only in the cases $\lambda = c$ or $\lambda = n - c$, where c is a constant natural. On the positive side, we present now a polynomial-time algorithm which, given a graph G of n vertices, returns a defense strategy with defense ratio which is within factor $2 + \frac{\lambda-3}{n}$ of the best defense ratio for G. In particular, it achieves defense ratio $1/p' \leq \left(2 + \frac{\lambda-3}{n}\right)/p^*(G)$, where $p' = \min_{i\in[n]} p_i$ and every p_i, $i \in [n]$ is the vertex probability determined by the constructed defense strategy. We henceforth write that a collection $\mathcal{L}$ of λ-subgraphs covers a graph $G = (V, E)$, if every vertex of V is covered by some λ-subgraph in $\mathcal{L}$. The algorithm presented in this section returns a collection $\mathcal{L}$ of at most $\frac{2n-3}{\lambda} + 1$ λ-subgraphs that covers G. Therefore, the uniform defense strategy over $\mathcal{L}$ assigns probability at least $1/\left(\frac{2n-3}{\lambda} + 1\right)$ to each λ-subgraph.

For any collection $\mathcal{L}$ of λ-subgraphs and for any $v \in V$, let us denote by $\text{coverage}_{\mathcal{L}}(v)$ the number of λ-subgraphs in $\mathcal{L}$ which v belongs in. Observe that:

$$\sum_{v\in V} \text{coverage}_{\mathcal{L}}(v) = |\mathcal{L}| \cdot \lambda, \tag{5.6}$$

where $|\mathcal{L}|$ denotes the cardinality of $\mathcal{L}$.

We first prove Lemma 15, to be used in the proof of the main theorem of this Section. We henceforth denote by $V(G)$ and $E(G)$ the vertex set and edge set, respectively, of some graph G.

Lemma 15. *For any tree T of n vertices, and for any $\lambda \leq n$, we can find a collection $\mathcal{L}$ of distinct λ-subgraphs such that for every $v \in V$, it holds that $1 \leq$ coverage$_{\mathcal{L}}(v) \leq$ degree(v), except maybe for (at most) $\lambda - 1$ vertices, where for each of them it holds that coverage$_{\mathcal{L}}(v) =$ degree$(v) + 1$.*

Proof. We will prove the statement of the lemma by providing Algorithm 1 that takes as input T and λ and outputs the requested collection $\mathcal{L}$ of λ-subgraphs.

The algorithm starts by picking an arbitrary vertex v to serve as the root of the tree. Then it performs a Depth-First-Search (DFS) starting from v. We will distinguish between

Algorithm 1 MAIN ALGORITHM

Require: A tree graph $T = (V, E)$ of n vertices, and a natural $\lambda \leq n$.
Ensure: A collection $\mathcal{L}$ of distinct λ-subgraphs that satisfies the statement of Lemma 15.

1: i, global variable. *% The index of the λ-subgraph L_i.*
2: *count*, global variable. *% Is 0 until the whole tree is covered, then it becomes 1 to allow for the last λ-subgraph to be completed, if it is not already.*
3: S, global variable. *% The set of vertices already covered by the algorithm.*
4: *vertex*, global variable. *% The vertex considered to be inserted in a λ-subgraph.*

5: $S \leftarrow \emptyset$
6: $i \leftarrow 1$
7: $L_i \leftarrow \emptyset$
8: Pick an arbitrary vertex v of T and consider it the root.
9: $vertex \leftarrow v$
10: $count \leftarrow 0$
11: **while** $count < 2$ **do**
12: **while** $S \neq V$ **do**
13: **while** $vertex \in S$ **do** *% The while-loop to ensure that the first element of L_i is uncovered.*
14: **if** *vertex* has a child $u \notin S$ **then**
15: $vertex \leftarrow u$
16: **else**
17: $vertex \leftarrow$ parent of *vertex*
18: **while** $|L_i| < \lambda$ **do** *% The while-loop that fills in the current λ-subgraph L_i.*
19: $L_i \leftarrow L_i \cup \{vertex\}$
20: $S \leftarrow S \cup \{vertex\}$
21: **if** *vertex* has a child $u \notin S$ **then**
22: $vertex \leftarrow u$
23: **else**
24: $vertex \leftarrow$ parent of *vertex*
25: **if** $count < 1$ **then**
26: $i \leftarrow i + 1$
27: $L_i \leftarrow \emptyset$
28: **else**
29: **break**
30: $S \leftarrow \emptyset$
31: $i \leftarrow i - 1$
32: Pick an arbitrary vertex $v \in L_i$ and consider it the root.
33: $vertex \leftarrow v$
34: $count \leftarrow count + 1$

visiting a vertex and *covering* a vertex in the following way. We say that DFS visited a vertex if it considered that vertex as a candidate to be inserted to some λ-subgraph, and we say that DFS covered a vertex if it visited *and* inserted the vertex at some λ-subgraph. By definition, DFS visits in a greedy manner first an uncovered child, and only if there is no such child, it visits its parent (lines 14-17, 21-24). The set-variable that keeps track of the covered vertices is S.

Starting with the root of T, the algorithm simply visits the whole vertex set according to DFS, putting each visited vertex in the same λ-subgraph L_i (starting with $i = 1$) (lines 18-24), and when $|L_i| = \lambda$, a new empty λ-subgraph L_{i+1} is picked to get filled in with λ vertices (lines 26-27) taking care of one extra thing: The first vertex that the algorithm puts in an empty λ-subgraph L_i, $i \in \{1, 2, \dots\}$ is guaranteed to be one that has not been covered by any other λ-subgraph so far (lines 13-17). This ensures that no two λ-subgraphs will eventually be identical.

The algorithm will not only visit all vertices in T, but also cover them. That is because there is no point where the algorithm checks whether the currently visited vertex is uncovered and then does not cover it. On the contrary, it covers every vertex that it visits, except for some already covered one in case the current λ-subgraph is empty (lines 13-24). And since DFS by construction visits every vertex, we know that at some point the whole vertex set will be covered, or equivalently, $\text{coverage}_{\mathcal{L}}(v) \geq 1, \forall v \in V$. Therefore, the algorithm will eventually exit the while-loop in lines 12-29.

Now we prove that, after the algorithm terminates, every vertex $v \in V$ is covered at most $degree(v)$ times, except for at most $\lambda - 1$ vertices that are covered $degree(v) + 1$ times. Observe that DFS visits every vertex v at most $degree(v)$ times: (a) v will be visited after its parent u only if v is uncovered (lines 14-15, 21-22), v will get covered (lines 19-20), and will not get visited ever again by its parent since it will be covered (lines 16-17, 23-24). (b) v will be visited at most once by each of its children, say w, only if w does not have an uncovered child (lines 16-17, 23-24), and v will not get ever visited by its parent since v will be covered, and also v cannot be visited a second time by any of its children, since they can never be visited again (they can only be visited through v since T is a tree). Therefore, any vertex v will be visited exactly once after its parent is visited, and at most once by each of its children, having a total of at most $degree(v)$ visits. And since, as argued above, the total number of times a vertex will be covered is at most the number of times it will get visited, when DFS terminates (i.e $S = V$), it will be $\text{coverage}_{\mathcal{L}}(v) \leq degree(v)$, for every $v \in V$.

However, note that the last nonempty λ-subgraph L_i might not consist of λ vertices since the entire V was covered and DFS could not proceed further. In this case, the algorithm empties the set S that keeps track of the covered nodes, takes the current L_i and fills it in with exactly another $\lambda - |L_i|$ vertices. This is done by picking an arbitrary vertex from L_i and setting it as the root of T, and performing one last DFS starting from it until L_i has λ vertices in total (lines 30-33). To ensure that the DFS will continue only until it fills in this current L_i, the algorithm counts the number of times that it runs the while-loop of DFS, namely lines 12-29, via the variable *count* (line 34), which escapes the while-loop of DFS in case DFS has filled in L_i (lines 28-29) and terminates. Observe that in the last λ-subgraph L_i, a vertex v inserted in the last iteration of DFS ($count = 1$) and was not inserted in L_i by the first run ($count = 0$) might have been covered by the first run of DFS exactly $degree(v)$ times, therefore when the algorithm terminates it has been covered $degree(v)+1$ times. Since by the end of the first DFS run L_i had at least one vertex, the cardinality of such vertices that are covered more times than their degree are at most $\lambda - 1$. □

We can now prove the following.

Lemma 16. *For any graph G of n vertices, and for any $\lambda \leq n$, there exist (at most) $\frac{2n-3}{\lambda} + 1$ λ-subgaphs of G that cover G.*

Proof. Consider a spanning tree T of G. Then Lemma 15 applies to T. Observe that a collection $\mathcal{L}$ as described in the statement of the aforementioned lemma has the same qualities for G since $V(T) = V(G)$ and $E(T) \subseteq E(G)$. That is, $\mathcal{L}$ is a collection of distinct λ-subgraphs of G, such that for every $v \in V$, it holds that $1 \leq \text{coverage}_{\mathcal{L}}(v) \leq \text{degree}(v)$, except maybe for (at most) $\lambda-1$ vertices, for each v of which it is $\text{coverage}_{\mathcal{L}}(v) = \text{degree}(v)+1$, where by $\text{degree}(v)$ we denote the degree of vertex v in T.

Fix a particular value for λ and consider a collection $\mathcal{L}$ of λ-subgraphs as constructed in the proof of Lemma 15. Then, by equation (5.6),

$$|\mathcal{L}| = \frac{\sum_{v \in V} \text{coverage}_{\mathcal{L}}(v)}{\lambda} \leq \frac{\sum_{v \in V} \text{degree}(v) + (\lambda - 1)}{\lambda} = \frac{2(n-1)}{\lambda} + \frac{\lambda - 1}{\lambda} = \frac{2n-3}{\lambda} + 1.$$

□

We conclude with the simple algorithm that achieves a defense strategy with defense ratio which is within factor $2 + \frac{\lambda-3}{n}$ of the best defense ratio for G.

Algorithm 2 Approximating the best defense ratio

Require: Graph $G = (V, E)$ of n vertices, a natural $\lambda \leq n$.
Ensure: A defense strategy that satisfies the statement of Theorem 18.

1: Find a spanning tree T of G.
2: Construct a collection $\mathcal{L}$ of λ-subgraphs of T as described in the proof of Lemma 15.
3: Assign probability $q_i = \frac{1}{|\mathcal{L}|}$ to every λ-subgraph in $\mathcal{L}$, $i = 1, 2, \ldots, |\mathcal{L}|$.
4: **return** The above uniform defense strategy over the collection $\mathcal{L}$.

Theorem 18. *Given any graph $G = (V, E)$, Algorithm 2 computes in time $O(|E|)$ a defense strategy such that, for any combination of attack strategies, the resulting strategy profile S yields defense ratio* $\mathrm{DR}(G, S) \leq \left(2 + \frac{\lambda-3}{n}\right) \cdot \mathrm{DR}(G, S^*)$.

Proof. As argued in Lemma 16, there is a collection $\mathcal{L}$ of λ-subgraphs with $|\mathcal{L}| \leq \frac{2n}{\lambda} + 1 - \frac{3}{\lambda}$ which covers G. Therefore, the uniform defense strategy returned by Algorithm 2 (which determines the vertex probability p_i for each vertex i) achieves a minimum vertex probability $p' := \min_{i \in [n]} p_i$ for which it holds that:

$$p' = \frac{1}{|\mathcal{L}|} \geq \frac{1}{\frac{2n}{\lambda} + 1 - \frac{3}{\lambda}} = \frac{\frac{\lambda}{n}}{2 + \frac{\lambda-3}{n}} \geq \frac{1}{2 + \frac{\lambda-3}{n}} \cdot p^*(G),$$

where the first equality is due to the fact that any leaf $v \in V$ of the spanning tree T of G through which $\mathcal{L}$ was created has $\mathrm{coverage}_{\mathcal{L}}(v) = 1$, and therefore there is such a vertex v in G that is covered by exactly one λ-subgraph; and the last inequality is due to the fact that $p^*(G) \leq \lambda/n$ for any graph G (due to Corollary 9), where $p^*(G)$ is the MaxMin probability of G.

The above inequality implies that if the defender chooses the prescribed strategy the minimum defense ratio cannot be too bad. That is because in the worst case for the defender, each and every attacker will choose a vertex v' on which the aforementioned strategy of the defender results to vertex probability p' (so that the attacker is caught with minimum probability). As a result, the defender will have the minimum possible expected payoff which is $p' \cdot k$. Thus, for the constructed defend strategy and any combination of attack strategies, the resulting strategy profile S yields defense ratio:

$$\mathrm{DR}(G, S) \leq \frac{k}{p' \cdot k} \leq \left(2 + \frac{\lambda - 3}{n}\right) \cdot \frac{1}{p^*(G)} = \left(2 + \frac{\lambda - 3}{n}\right) \cdot \mathrm{DR}(G, S^*),$$

where the last equality is due to Lemma 13.

With respect to the running time, notice that Step 1 of Algorithm 2 can be executed in time $O(|V| + |E(G)|) = O(|E(G)|)$. Step 2 can be executed in time $O(|V| + |E(T)|) = O(|V|)$. Finally, Step 3 can be executed in constant time. Therefore, the total running time of Algorithm 2 is $O(|E(G)|)$. □

Corollary 10. *For any graph G there is a polynomial (in both n and λ) time approximation algorithm (Algorithm 2) with approximation factor $1/\left(2 + \frac{\lambda-3}{n}\right)$ for the computation of $p^*(G)$.*

The merit of finding a probability p' that approximates (from below) $p^*(G)$ for a given graph G through an algorithm such as Algorithm 2 is in guaranteeing to the defender that, no matter what the attackers play, she always "catches" at least a portion p' of them in expectation, where the best portion is $p^*(G)$ in an equilibrium. Algorithm 2 guarantees that the defender catches at least $1/\left(2 + \frac{\lambda-3}{n}\right)$ of the attackers in expectation.

5.5 Bounds on the Price of Defense

In the following theorem we give a lower bound on the PoD for any given n and $2 \leq \lambda \leq n-1$ by constructing a graph G with particular (very small) $p^*(G)$ (which, by Lemma 13 implies great best defense ratio).

Theorem 19. *The* PoD*(λ) is lower bounded by $\lfloor \frac{2(n-1)}{\lambda} \rfloor$ and $\lfloor \frac{2(n-1)}{\lambda+1} \rfloor$ for λ even and odd respectively, when $\lambda \in \{2, 3, \ldots, n-1\}$.*

Proof. We will prove the statement by showing that for any given n and $\lambda \in \{2, 3, \ldots, n-1\}$, there exists a graph $G = (V, E)$ on n vertices that requires (at least) some number roughly $b = \lfloor \frac{2(n-1)}{\lambda+1} \rfloor$ of λ-subgraphs to be covered and additionally this graph's structure achieves $p^*(G)$ for the uniform defense strategy, i.e. each λ-subgraph is assigned equal probability $1/b$.

The graph we construct is the following. First, consider a line graph with σ vertices, where $\sigma = \lceil \frac{\lambda}{2} \rceil$. Keep a *central vertex* to use later, and using only $n-1$ vertices, create as many *complete lines* with σ vertices as possible, i.e. $b = \lfloor \frac{n-1}{\sigma} \rfloor$. Create another *incomplete line* (if needed) with strictly less than σ vertices using the remaining ones $n - 1 - b \cdot \sigma$. Now draw an edge from the central vertex to a single leaf of each of the constructed lines.

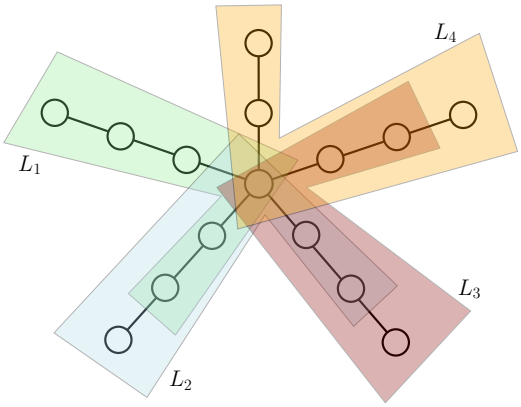


Figure 5.2: An example of **Case 1** of Theorem 19, where $n = 15$ and $\lambda = 6$. Here, graph G has $\sigma = 3$ and $b = 4$. The λ-subgraphs L_1, L_2, L_3, L_4 that constitute the support of a best-defense strategy are shown with various colors.

For examples of the construction of G in each of the below three cases, see Figures 5.2, 5.3, and 5.4.

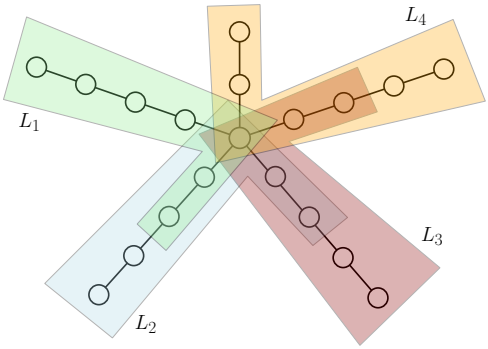


Figure 5.3: An example of **Case 2(a)** of Theorem 19, where $n = 19$ and $\lambda = 7$. Here, graph G has $\sigma = 4$ and $b = 4$. The λ-subgraphs L_1, L_2, L_3, L_4 that constitute the support of a best-defense strategy are shown with various colors.

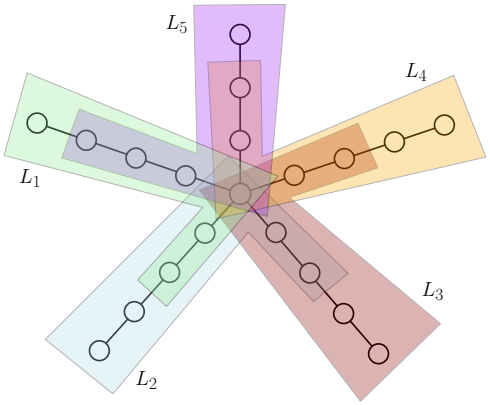


Figure 5.4: An example of **Case 2(b)** of Theorem 19, where $n = 20$ and $\lambda = 7$. Here, graph G has $\sigma = 4$ and $b = 4$. The λ-subgraphs L_1, L_2, L_3, L_4, L_5 that constitute the support of a best-defense strategy are shown with various colors.

Consider now a defense strategy $q := (q_1, q_2, \ldots, q_\theta) \in \Delta_\theta$ and the vertex probabilities

$p_1, p_2, \ldots, p_n$ it induces on the vertices of G.

Case 1: λ is even. In this case $\sigma = \lambda/2$ and observe that the diameter of this graph G is equal to λ, therefore no λ-subgraph that covers a leaf of a complete line can cover a leaf of another complete line. Also, any λ-subgraph that covers a leaf of a complete line can cover the whole incomplete line. Therefore, this graph can be covered by b λ-subgraphs but no less. Assume that q covers G, i.e. $p_i > 0, \forall i \in [n]$, and let us focus on the set V_{com} of leaves of the complete lines of G, where $|V_{com}| = b$ as argued earlier, and denote V_{com} by $[b]$. Consider the vertex probabilities p_i, $i \in [b]$, and note that $\sum_{i\in[b]} p_i \leq 1$ where strict inequality holds for the case where there exists some pure strategy $L_j \in supp(q)$ such that $L_j \cap V_{com} = \emptyset$. Then for $p' := \min_{i\in[b]} p_i$ it holds that $p' \leq 1/b$, otherwise $p_i > 1/b$, $\forall i \in [b]$ and therefore $\sum_{i\in[b]} p_i > 1$ which is a contradiction. Also, for $p_i = 1/b$, $\forall i \in [b]$, it is $p' = 1/b$, which yields $p^*(G) := \max_{q\in\Delta_\theta} p' = 1/b$.

Case 2: λ is odd. In this case $\sigma = (\lambda + 1)/2$ and the diameter of G equals $\lambda + 1$, therefore no λ-subgraph that covers a leaf of a complete line can cover a leaf of another complete line.

- **Subcase (a): $\sigma - (n - 1 - b \cdot \sigma) \neq 1$.** Any λ-subgraph that covers a leaf of a complete line can cover the whole incomplete line. Therefore, this graph can be covered with b λ-subgraphs but no less. Following the analysis of Case 1, it is $p^*(G) := \max_{q\in\Delta_\theta} p' = 1/b$.

- **Subcase (b): $\sigma - (n - 1 - b \cdot \sigma) = 1$.** No λ-subgraph that covers a leaf of a complete line can cover the leaf of the incomplete line. Therefore, this graph can be covered by $b + 1$ λ-subgraphs but no less. Following similar analysis as that of Case 1, where instead of V_{com} we have $V_{com} \cup \{v_{inc}\}$ where v_{inc} is the leaf of the incomplete line, and instead of b we have $b + 1$, we conclude that $p^*(G) := \max_{q\in\Delta_\theta} p' = 1/(b + 1)$.

For Case 1, and Case 2(a), since each of the leaves of the b complete lines have vertex probability $1/b$, the defense strategy q^* with probability $q^*_i = 1/b$ assigned to the respective pure defense strategy $L_i, i \in [b]$ that contains vertex $i \in [b]$, yields $p^*(G)$. For Case 2(b), since each of the leaves of the b complete lines and the leaf v_{inc} of the incomplete line have vertex probability $1/(b+1)$, the defense strategy q^* with probability $q^*_i = 1/(b+1)$ assigned to the respective pure strategy $L_i, i \in [b] \cup \{v_{inc}\}$ that contains vertex $i \in [b] \cup \{v_{inc}\}$, yields $p^*(G)$.

By the above values of $p^*(G)$ and Lemma 13 the proof of the theorem is complete. □

Corollary 11. *For any given n and $2 \leq \lambda \leq n-1$, it holds that $\lfloor \frac{2(n-1)}{\lambda+1} \rfloor \leq \mathrm{PoD}(\lambda) \leq \frac{2(n-1)+\lambda-1}{\lambda}$. Furthermore, for the trivial cases $\lambda \in \{1, n\}$ it is $\mathrm{PoD}(1) = n$ and $\mathrm{PoD}(n) = 1$.*

Proof. The lower bound is established by Theorem 19. The upper bound is due to Theorem 18. For the cases $\lambda = 1$ and $\lambda = n$, observe that the defender's action set is $D = \{\{i\} | i \in V\}$ and $D = \{V\}$ respectively, therefore $p^*(G) = 1/n$ and $p^*(G) = 1$ respectively, and again from Lemma 13 we get the values in the statement of the corollary. □

Part III

Fair Division

Chapter 6

The Consensus Halving Problem and the Borsuk-Ulam Theorem

In this chapter we study the problem of finding an exact solution to the Consensus Halving problem. While recent work has shown that the approximate version of this problem is PPA-complete [71,72], we show that the exact version is much harder. Specifically, finding a solution with n agents and n cuts is FIXP-hard, and deciding whether there exists a solution with fewer than n cuts is ETR-complete.

Along the way, we define a new complexity class BU, which captures all problems that can be reduced to solving an instance of the Borsuk-Ulam problem exactly. We show that $\texttt{FIXP} \subseteq \texttt{BU} \subseteq \texttt{TFETR}$ and that $\texttt{LinearBU} = \texttt{PPA}$, where LinearBU is the subclass of BU in which the Borsuk-Ulam instance is specified by a linear arithmetic circuit.

The results of this chapter have been published in the Proceedings of the 46th International Colloquium on Automata, Languages and Programming (ICALP 2019) [58] (co-authored with Deligkas, Fearnley and Spirakis).

6.1 Overview

Dividing resources among agents in a fair manner is among the most fundamental problems in multi-agent systems [32]. Cake cutting [11,17,18,31], and rent division [30,65,79] are prominent examples of problems that lie in this category. At their core, each of these problems has a desired solution whose existence is usually proved via a theorem from algebraic topology such as Brouwer's fixed point theorem, Sperner's lemma, or Kakutani's

fixed point theorem.

In this work we focus on a fair-division problem called *Consensus Halving*: an object A represented by $[0, 1]$ is to be divided into two halves A_+ and A_-, so that n agents agree that A_+ and A_- have the same value. Provided the agents have bounded and continuous valuations over A, this can always be achieved using at most n cuts, and this fact can be proved via the Borsuk-Ulam theorem from algebraic topology [133]. The necklace splitting and ham-sandwich problems are two other examples of fair-division problems for which the existence of a solution can be proved via the Borsuk-Ulam theorem [6,7,120].

Recent work has further refined the complexity status of *approximate Consensus Halving*, in which we seek a division of the object so that every agent agrees that the values of A_+ and A_- differ by at most ϵ. Since the problem always has a solution, it lies in TFNP, which is the class of function problems in NP that always have a solution. More recent work has shown that the problem is PPA-complete [71], even for ϵ that is inverse-polynomial in n [72]. The problem of deciding whether there exists an approximate solution with k cuts when $k < n$ is NP-complete [70]. These results are particularly notable, because they identify consensus halving as one of the first natural PPA-complete problems.

While previous work has focused on approximate solutions to the problem, in this work we study the complexity of solving the problem *exactly*. For problems in the complexity class PPAD, which is a subclass of both TFNP and PPA, prior work has found that there is a sharp contrast between exact and approximate solutions. For example, the Brouwer fixed point theorem is the theorem from algebraic topology that underpins PPAD. Finding an approximate Brouwer fixed point is PPAD-complete [120], but finding an exact Brouwer fixed point is complete for (and the defining problem of) a complexity class called FIXP [66].

It is believed that FIXP is significantly harder than PPAD. While PPAD $\subseteq$ TFNP $\subseteq$ FNP, there is significant doubt about whether FIXP $\subseteq$ FNP. The reason for this is that there are Brouwer instances for which all solutions are irrational. This is not particularly relevant when we seek an approximate solution, but is a major difficulty when we seek an exact solution. For example, the square-root-sum problem asks us to decide for integers a_1, a_2, $\ldots$, a_n, t, whether $\sum_{i=1}^{n} \sqrt{a_i} \leq t$. This deceptively simple problem is not known to lie in NP, and can be reduced to the problem of finding an exact Brouwer fixed point [66], which provides evidence that FIXP may be significantly harder than FNP.

6.1.1 Contribution

In this work we study the complexity of solving the consensus halving problem exactly. In our formulation of the problem, the valuation function of the agents is presented as an arbitrary arithmetic circuit, and the task is to cut A such that all agents agree that A_+ and A_- have exactly the same valuation. We study two problems. The (n, n)-Consensus Halving problem asks us to find an exact solution for n agents using at most n cuts, while the (n, k)-Consensus Halving problem asks us to decide whether there exists an exact solution for n agents using at most k cuts, where $k < n$.

Our results for (n, n)-Consensus Halving are intertwined with a new complexity class that we call BU. This class consists of all problems that can be reduced in polynomial time to the problem of finding a solution of the Borsuk-Ulam problem. We show that (n, n)-Consensus Halving lies in BU, and is FIXP hard. The hardness for FIXP implies that the exact variant of consensus halving is significantly harder than the approximate variant: while the approximate problem is PPA-complete, the exact variant is unlikely to be in FNP.

We show that (n, k)-Consensus Halving is ETR-complete. The complexity class ETR consists of all decision problems that can be formulated in the *existential theory of the reals*. It is known that NP $\subseteq$ ETR $\subseteq$ PSPACE [35], and it is generally believed that ETR is distinct from the other two classes. So our result again shows that the exact version of the problem seems to be much harder than the approximate version, which is NP-complete [70].

Just as FIXP can be thought of as the exact analogue of PPAD, we believe that BU is the exact analogue of PPA, and we provide some evidence to justify this. It has been shown that LinearFIXP = PPAD [66], which is the version of the class in which arithmetic circuits are restricted to produce piecewise *linear* functions (FIXP allows circuits to compute piecewise polynomials). We likewise define LinearBU, which consists of all problems that can be reduced to a solution of a Borsuk-Ulam problem using a piecewise linear function, and we show that LinearBU = PPA.

The containment LinearBU $\subseteq$ PPA can be proved using similar techniques to the proof that LinearFIXP $\subseteq$ PPAD. However, the proof that PPA $\subseteq$ LinearBU utilises our BU containment result for consensus halving. In particular, when the input to the consensus halving problem is a piecewise linear function, our containment result shows that the problem actually lies in LinearBU. The PPA-hardness results for consensus halving show that piecewise-linear-consensus halving is PPA-hard, which completes the containment [71,72].

6.1.2 Related work

Although for a long period there were a few results about PPA, recently there has been a flourish of PPA-completeness results. The first PPA-completeness result was given by [78] who showed PPA-completeness of the Sperner problem for a non-orientable 3-dimensional space. In [74] this result was strengthened for a non-orientable and locally 2-dimensional space. In [3], 2-dimensional Tucker was shown to be PPA-complete; this result was used in [71,72] to prove PPA-completeness for approximate consensus halving. In [63] PPA-completeness was proven for a special version of Tucker and for problems of the form "given a discrete fixed point in a non-orientable space, find another one". Finally, in [64] it was shown that octahedral Tucker is PPA-complete. In [106], a subclass of $\texttt{2DLinearFIXP} \subseteq \texttt{FIXP}$ that consists of 2-dimensional fixed-point problems was studied, and it was proven that $\texttt{2DLinearFIXP} = \texttt{PPAD}$.

A large number of problems are now known to be ETR-complete: geometric intersection problems [99,123], graph-drawing problems [1,23,39,124], matrix factorization problems [131,132], the Art Gallery problem [2], and deciding the existence of constrained (symmetric) Nash equilibria in (symmetric) normal form games with at least three players [24–27,75].

6.2 Preliminaries

6.2.1 Arithmetic circuits

An arithmetic circuit is a representation of a continuous function $f : \mathbb{R}^n \to \mathbb{R}^m$. The circuit is defined by a pair $(V, \mathcal{T})$, where V is a set of nodes and $\mathcal{T}$ is a set of gates. There are n nodes in V that are designated to be *input nodes*, and m nodes in V that are designated to be *output nodes*. When a value $x \in \mathbb{R}^n$ is presented at the input nodes, the circuit computes values for all other nodes $v \in V$, which we will denote as $x[v]$. The values of $x[v]$ for the m output nodes determine the value of $f(x) \in \mathbb{R}^m$.

Every node in V, other than the input nodes, is required to be the output of exactly one gate in $\mathcal{T}$. Each gate $g \in \mathcal{T}$ enforces an arithmetic constraint on its output node, based on the values of some other node in the circuit. Cycles are not allowed in these constraints. We allow the operations $\{\zeta, +, -, *\zeta, *, \max, \min\}$, which correspond to the gates shown in Table 6.1. Note that every gate computes a continuous function over its inputs, and thus any function f that is represented by an arithmetic circuit of this form is also continuous.

Gate	Constraint
$G_\zeta(\zeta, v_{out})$	$x[v_{out}] = \zeta$, where $\zeta \in \mathbb{Q}$
$G_+(v_{in1}, v_{in2}, v_{out})$	$x[v_{out}] = x[v_{in1}] + x[v_{in2}]$
$G_-(v_{in1}, v_{in2}, v_{out})$	$x[v_{out}] = x[v_{in1}] - x[v_{in2}]$
$G_{*\zeta}(\zeta, v_{in}, v_{out})$	$x[v_{out}] = x[v_{in1}] \cdot \zeta$, where $\zeta \in \mathbb{Q}$
$G_*(v_{in1}, v_{in2}, v_{out})$	$x[v_{out}] = x[v_{in1}] \cdot x[v_{in2}]$
$G_{\max}(v_{in1}, v_{in2}, v_{out})$	$x[v_{out}] = \max\{x[v_{in1}], x[v_{in2}]\}$
$G_{\min}(v_{in1}, v_{in2}, v_{out})$	$x[v_{out}] = \min\{x[v_{in1}], x[v_{in2}]\}$

Table 6.1: The types of gates and their constraints.

We study two types of circuits in this work. *General* arithmetic circuits are allowed to use any of the gates that we have defined above. *Linear* arithmetic circuits allow only the operations $\{\zeta, +, -, *\zeta, \max, \min\}$, and the $*$ operation (multiplication of two variables) is disallowed. Observe that a linear arithmetic circuit computes a continuous, piecewise linear function.

6.2.2 The Consensus Halving problem

In the consensus halving problem there is an object A that is represented by the $[0, 1]$ line segment, and there are n agents. We wish to divide A into two (not necessarily contiguous) pieces such that every agent agrees that the two pieces have equal value. Simmons and Su [133] have shown that, provided the agents have bounded and continuous valuations over A, then we can find a solution to this problem using at most n cuts.

In this work we consider instances of the consensus halving problem where the valuations of the agents are presented as arithmetic circuits. Each agent has a valuation function $f_i : [0, 1] \to \mathbb{R}$, but it is technically more convenient if they give us a representation of the *integral* of this function. So for each agent i, we are given an arithmetic circuit computing $F_i : [0, 1] \to \mathbb{R}$ where for all $x \in [0, 1]$ we have $F_i(x) = \int_0^x f(y)\, dy$. Then, the value of any particular segment of $[a, b]$ to agent i can be computed as $F_i(b) - F_i(a)$.

A solution to the consensus halving problem is given by a *k-cut* of the object A, which is defined by a vector of *cut-points* $(t_1, t_2, \ldots, t_k) \in [0, 1]^k$, and a vector of *signs* $(s_1, s_2, \ldots, s_{k+1}) \in \{-1, +1\}^{k+1}$. The cut-points t_i split A into up to $k + 1$ pieces. Note that they may in fact split A into fewer than $k + 1$ pieces in the case where two cut-points $t_i = t_j$ overlap. We define X_i to be the ith piece of A, meaning that $X_0 = [0, t_1]$,

$X_i = [t_i, t_{i+1}]$ for all i in the range $1 \leq i < k$, and $X_k = [t_k, 1]$.

The sign vector determines which half of A the piece belongs to. We define $A_+ := \{X_i : s_i = +1\}$ and $A_- := \{X_i : s_i = -1\}$ to be the two halves. For each agent i, we denote the value A_+ to agent i as $F_i(A_+) := \sum_{[a,b] \in A_+} (F_i(b) - F_i(a))$, and we define $F_i(A_-)$ analogously. The k-cut is a solution to the consensus halving problem if $F_i(A_+) = F_i(A_-)$ for all agents i.

We define two computational problems. Simmons and Su [133] have proved that there always exists a solution using at most n cuts, and our first problem is to find that solution.

(n, n)-Consensus Halving

Input: For every agent $i \in [n]$, an arithmetic circuit F_i computing the integral of agent i's valuation function.
Task: Find an n-cut for A such that $F_i(A_+) = F_i(A_-)$, for every agent $i \in [n]$.

For $k < n$ a solution to the problem may or may not exist. So we define the following decision variant of the problem.

(n, k)-Consensus Halving

Input: For every agent $i \in [n]$, an arithmetic circuit F_i computing the integral of agent i's valuation function.
Task: Decide whether there exists a k-cut for A such that $F_i(A_+) = F_i(A_-)$, for every agent $i \in [n]$.

For either of these two problems, if all of the inputs are represented by linear arithmetic circuits, then we refer to the problem as Linear Consensus Halving. We note that the known hardness results [70,71] for consensus halving fall into this class. Specifically, those results produce valuations that are piecewise constant, and so the integral of these functions is piecewise linear, and these functions can be written down as linear arithmetic circuits [117].

6.3 The Class BU

The Borsuk-Ulam theorem states that every continuous function from the surface of a $(d+1)$-dimensional sphere to the d-dimensional Euclidean space maps at least one pair of antipodal points to the same point.

Theorem 20 (Borsuk-Ulam)**.** *Let $f : S^d \to \mathbb{R}^d$ be a continuous function, where S^d is a $(d+1)$-dimensional sphere. Then, there exists an $x \in S^d$ such that $f(x) = f(-x)$.*

This theorem actually works for any domain D that is antipode-preserving homeomorphism of S^d, where by "antipode-preserving" we mean that for every $x \in D$ we have that $-x \in D$. In this work, we choose S^d to be the sphere in $d+1$ dimensions with respect to L_1 norm: $S^d := \left\{ x \mid x = (x_1, x_2, \ldots, x_{d+1}),\ \sum_{i=1}^{d+1} |x_i| = 1 \right\}$.

We define the *Borsuk-Ulam* problem as follows.

BORSUK-ULAM

Input: A continuous function $f : \mathbb{R}^{d+1} \to \mathbb{R}^d$ presented as an arithmetic circuit.
Task: Find an $x \in S^d$ such that $f(x) = f(-x)$.

Note that we cannot constrain an arithmetic circuit to only take inputs from the domain S^d, so we instead put the constraint that $x \in S^d$ onto the solution.

The complexity class BU is defined as follows.

Definition 16 (BU)**.** *The complexity class* BU *consists of all search problems that can be reduced to* BORSUK-ULAM *in polynomial time.*

6.3.1 LinearBU

When the input to a BORSUK-ULAM instance is a linear arithmetic circuit, then we call the problem LINEAR BORSUK-ULAM, and we define the class LinearBU as follows.

Definition 17 (LinearBU)**.** *The complexity class* LinearBU *consists of all search problems that can be reduced to* LINEAR BORSUK-ULAM *in polynomial time.*

We will show that LinearBU = PPA. The proof that LinearBU $\subseteq$ PPA is similar to the proof that Etessami and Yannakakis used to show that LinearFIXP $\subseteq$ PPAD [66], while the fact that PPA $\subseteq$ LinearBU will follow from our results on consensus halving in Section 6.4.

To prove LinearBU $\subseteq$ PPA we will reduce to the *approximate* Borsuk-Ulam problem. It is well known that the Borsuk-Ulam theorem can be proved via Tucker's lemma, and Papadimitriou noted that this implies that finding an approximate solution to a Borsuk-Ulam problem lies in PPA [120]. This is indeed correct, but the proof provided in [120] is for

a slightly different problem[1]. Since our results will depend on this fact, we provide our own definition and self-contained proof here. We define the approximate Borsuk-Ulam problem as follows.

ϵ-Borsuk-Ulam

Input: A continuous function $f : \mathbb{R}^{d+1} \to \mathbb{R}^d$ presented as an arithmetic circuit, along with two constants $\epsilon, \lambda \in \mathbb{R}$.
Task: Find one of the following.

1. A point $x \in S^d$ such that $\|f(x) - f(-x)\|_\infty \leq \epsilon$.
2. Two points $x, y \in S^d$ such that $\|f(x) - f(y)\|_\infty > \lambda \cdot \|x - y\|_\infty$.

The first type of solution is an approximate solution to the Borsuk-Ulam problem, while the second type of solution consists of any two points that witness that the function is not λ-Lipschitz continuous in the L_∞-norm. The second type of solution is necessary, because an arithmetic circuit is capable, through repeated squaring, of computing doubly-exponentially large numbers, and the reduction to Tucker may not be able to find an approximate solution for such circuits. We now re-prove the result of Papadimitriou in the following lemma.

Lemma 17 ([120])**.** ϵ-Borsuk-Ulam *is in* PPA*.*

Proof. This proof is essentially identical to the one given by Papadimitriou, but various minor changes must be made due to the fact that our input is an arithmetic circuit, and our domain is the L_1-sphere. His proof works by reducing to the Tucker problem. In this problem we have an antipodally symmetric triangulation of S^d with set of vertices V, and a labelling function $L : V \to \{-1, 1, -2, 2, \ldots, -d, d\}$ that satisfies $L(v) = -L(-v)$ for all $v \in V$. The task is to find two adjacent vertices v and u such that $L(v) = -L(u)$, whose existence is guaranteed via Tucker's lemma. Papadimitriou's containment proof goes via the hypercube, but in [70] it is pointed out that this problem also lies in PPA when the domain is the L_1-sphere S^d.

To reduce the ϵ-Borsuk-Ulam problem for (f, ϵ, λ) to Tucker, we choose an arbitrary triangulation of S^d such that the distance between any two adjacent vertices is at most ϵ/λ. Let $g(x) = f(x) - f(-x)$. To determine the label of a vertex $v \in V$, first find the coordinate

[1]The problem used in [120] presents the function as a polynomial-time Turing machine rather than an arithmetic circuit, and the Lipschitzness of the function is guaranteed by constraining the values that it can take.

i that maximises $|g(v)_i|$ breaking ties arbitrarily, and then set $L(v) = i$ if $g(v)_i > 0$ and $L(v) = -i$ otherwise.

Tucker's lemma will give us two adjacent vertices v and u satisfying $L(v) = -L(u)$, and we must translate this to a solution to ϵ-Borsuk-Ulam. If $\|g(u) - g(v)\|_\infty > \lambda \cdot \|u - v\|_\infty$, then we have a violation of Lipschitz continuity. Otherwise, we have

$$\begin{aligned} \|g(u) - g(v)\|_\infty &\leq \lambda \cdot \|u - v\|_\infty \\ &\leq \lambda \cdot \frac{\epsilon}{\lambda} \\ &\leq \epsilon \end{aligned}$$

Let $i = L(v)$. Note that by definition we have that $|g(v)_j| \leq |g(v)_i|$ for all j, that $|g(u)_j| \leq |g(u)_i|$ for all j, and that that $g(u)_i$ and $g(v)_i$ have opposite signs. These three facts, along with the fact that $\|g(u) - g(v)\|_\infty \leq \epsilon$ imply that $|g(v)_j| \leq \epsilon$ for all j. Hence we can conclude that $\|f(v) - f(-v)\|_\infty \leq \epsilon$ meaning that v is a solution to ϵ-Borsuk-Ulam. □

To show that `LinearBU` $\subseteq$ `PPA` we will provide a polynomial time reduction from Linear Borsuk-Ulam to ϵ-Borsuk-Ulam. To do this, we follow closely the technique used by Etessami and Yannakakis to show that `LinearFIXP` $\subseteq$ `PPAD` [66]. The idea is to make a single call to ϵ-Borsuk-Ulam to find an approximate solution to the problem for a suitably small ϵ, and to then round to an exact solution by solving a linear program. To build the LP, we depend on the fact that we have access to the linear arithmetic circuit that represents f.

Lemma 18. Linear Borsuk-Ulam *is in* `PPA`.

Proof. Suppose that we have a function f that is represented as a linear arithmetic circuit. We will provide a polynomial time reduction to ϵ-Borsuk-Ulam.

The first step is to argue that, for all $\epsilon > 0$, we can make a single call to ϵ-Borsuk-Ulam in order to find an ϵ-approximate solution to the problem. The only technicality here is that we must choose λ so as to ensure that no violations of λ-Lipschitzness in the L_∞-norm can be produced as a solution.

Fortunately, every linear arithmetic circuit computes a λ-Lipschitz function where the bit-length of λ is polynomial in the size of the circuit. Moreover, an upper bound on λ can easily be computed by inspecting the circuit.

- An input to the circuit has a Lipschitz constant of 1.

- A $+$ gate operating on two gates with Lipschitz constants x and y has a Lipschitz constant of at most $x + y$.
- A $*\zeta$ gate operating on a gate with Lipschitz constant x has a Lipschitz constant of at most $|\zeta| \cdot x$.
- A max or min gate operating on two gates with Lipschitz constants x and y has a Lipschitz constant of at most $\max(x, y)$.

The Lipschitz constant for the circuit in the L_∞-norm is then the maximum of the Lipschitz constants of the output nodes of the circuit. So, for any given $\epsilon > 0$ that can be represented in polynomially many bits, we can make a single call to ϵ-Borsuk-Ulam, in order to find an ϵ-approximate solution to the Borsuk-Ulam problem.

The second step is to choose an appropriate value for ϵ so that the approximate solution can be rounded to an exact solution using an LP. Let $g(x) = f(x) - f(-x)$. Note that $g(x)$ can also be computed by a linear arithmetic circuit, and that $g(x) = 0$ if and only if $f(x) = f(-x)$.

We closely follow the approach of Etessami and Yannakakis [66]. They use the fact that the function computed by a linear arithmetic circuit is piecewise-linear, and defined by (potentially exponentially many) hyperplanes. They give an algorithm that, given a point p in the domain of the circuit, computes in polynomial time the hyperplane that defines the output of the circuit for p. Furthermore, they show that the following can be produced in polynomial time from the representation of the circuit and from p.

- A system of linear constraints $Ax \leq b$ such that a point x satisfies the constraints if and only if the hyperplane that defines the output of the circuit for p also defines the output of the circuit for x.
- A linear formula $Cx + C'$ that determines the output of the circuit for all points that satisfy $Ax \leq b$.

To choose ϵ, the following procedure is used. Let n be the number of inputs to g, and let m be an upper bound on the bit-size of the solution of any linear system with $n + 1$ equations where the coefficients are drawn from the hyperplanes that define the function computed by g. This can be computed in polynomial time from the description of the circuit, and m will have polynomial size in relation to the description of the circuit. We choose $\epsilon < 1/2^m$.

We make one call to ϵ-Borsuk-Ulam to find a point $p \in S^n$ such that $\|f(p) - f(-p)\|_\infty \leq \epsilon$, meaning that $\|g(p)\|_\infty \leq \epsilon$. The final step is to round this to an exact solution of Borsuk-Ulam. To do this, we can modify the linear program used by Etessami and Yannakakis [66]. We apply the operations given above to the circuit g and the point p to obtain the system of constraints $Ax \leq b$ and the formula $Cx + C'$ for the hyperplane defining the output of g for p. We then solve the following linear program. The variables of the LP are a vector x of length n, and a scalar z. The goal is to minimize z subject to:

$$\begin{aligned}
Ax &\leq b && \\
(Cx)_i + C'_i &\leq z && \text{for } i = 1, \ldots, n \\
-((Cx)_i + C'_i) &\leq z && \text{for } i = 1, \ldots, n \\
x_i &\geq 0 && \text{for each } i \text{ with } p_i \geq 0 \\
x_i &< 0 && \text{for each } i \text{ with } p_i < 0 \\
\sum_{i=1}^{n} |x_i| &= 1 && \text{(see below regarding } |x_i|)
\end{aligned}$$

The first constraint ensures that we remain on the same hyperplane as the one defining the output of g for p. The second and third constraints ensure that $\|g(x)\|_\infty \leq z$. The fourth and fifth constraints ensure that x_i has the same sign as p_i, while the sixth constraint ensures that x lies on the surface S^n. Note that the $|x_i|$ operation in the sixth constraint is not a problem, since the fourth and fifth constraints mean that we know the sign of x_i up front, and so we just need to add either x_i or $-x_i$ to the sum. All of the above implies that that x is a z-approximate solution of Borsuk-Ulam for f.

We must now argue that the solution sets $z = 0$. First we note that the LP has a solution, because the point (p, ϵ) is feasible, and the LP is not unbounded since z cannot be less than zero due to the second and third constraints. So let (x^*, z^*) be an optimal solution. This solution lies at the intersection of $n+1$ linear constraints defined by rationals drawn from the circuit representing g, and so it follows that z^* is a rational of bit length at most m. Since $0 \leq z^* \leq \epsilon < 1/2^m$, it follows that $z^* = 0$, and thus x^* is an exact solution to Borsuk-Ulam for f. □

6.4 Containment Results for Consensus Halving

6.4.1 (n, n)-Consensus Halving is in BU and LinearBU = PPA

We show that (n, n)-Consensus Halving is contained in BU. Simmons and Su [133] show the existence of a n-cut solution to the consensus halving problem by applying the Borsuk-Ulam theorem, and we follow their approach in this reduction. However, we must show that the approach can be implemented using arithmetic circuits. We take care in the reduction to avoid G_* gates, and so if the inputs to the problem are all linear arithmetic circuits, then our reduction will produce a Linear Borsuk-Ulam instance. Hence, we also show that (n, n)-Linear Consensus Halving is in LinearBU.

Theorem 21. *The following two containments hold.*

- (n, n)-Consensus Halving *is in* BU.
- (n, n)-Linear Consensus Halving *is in* LinearBU.

Proof. Let us first summarise the approach used by Simmons and Su [133]. Given valuation functions F_i for the n agents, they construct a Borsuk-Ulam instance given by a function $b : S^n \to \mathbb{R}^n$. Each point $(x_1, x_2, \ldots, x_{n+1}) \in S^n$ can be interpreted as a n-cut of $[0, 1]$, where $|x_i|$ gives the *width* of the ith piece, and the sign of x_i indicates whether the ith piece should belong in A_+ or A_-. They then define $b(x)_i = F_i(A_+)$ for each agent i. The fact that $-x$ flips the sign of each piece, but not the width, implies that $b(-x)_i = F_i(A_-)$. Hence, any point that satisfies $b(x) = b(-x)$ has the property that $F_i(A_+) = F_i(A_-)$ for all agents i, and so is a solution to the consensus halving problem.

Our task is to implement this reduction using arithmetic circuits. Suppose that we are given arithmetic circuits F_i implementing the integral of each agent's valuation function. Given a point $(x_1, x_2, \ldots, x_{n+1}) \in S^n$, we show that $b(x)_i = F_i(A_+)$ can be computed via a linear arithmetic circuit. The tricky part of this, is that we must only include the ith piece in the sum if x_i is positive.

We begin by observing that the operation of $|x|$ can be implemented via a linear arithmetic circuit. Specifically, via the following construction:

$$|x| := \max(x, 0) + \max(-x, 0).$$

Hence, we can implement $|x|$ using only max, plus, and constant gates. Then, we define $t_0 := 0$ and $x_0 := 0$, and for each j in the range $1 \leq j \leq n+1$, define:

$$t_j := t_{j-1} + |x_{j-1}|.$$

The value of t_j gives the start of the jth piece. Next, for each j in the range $1 \leq j \leq n+1$ we define:

$$p_j := \max(x_j, 0).$$

Note that p_j is x_j whenever x_j is positive, and zero otherwise. Finally, for $1 \leq j \leq n+1$ define:

$$q_j := F_i(t_j + p_j) - F_i(t_j).$$

Using the reasoning above, we can see that q_j is agent i's valuation for piece j whenever x_j is positive, and zero otherwise. So we can define

$$b(x)_i = \sum_{j=1}^{n+1} q_j,$$

implying that $b(x)_i = F_i(A_+)$, as required.

To complete the proof, it suffices to note that none of the operations specified above use the gate G_*, and so if each F_i is specified by a linear arithmetic circuit, then b will also be a linear arithmetic circuit. □

We note that this also implies that PPA $\subseteq$ LinearBU, thereby completing the proof that PPA = LinearBU. Specifically, Filos-Ratsikas and Goldberg have shown that *approximate*-(n, n)-Consensus Halving is PPA-complete, and their valuation functions are piecewise constant. Therefore, the integrals of these functions are piecewise linear, and so their approximate-(n, n)-Consensus Halving instances can be reduced to (n, n)-Linear Consensus Halving. Hence (n, n)-Linear Consensus Halving is PPA-hard, which along with Lemma 18 implies the following corollary.

Corollary 12. PPA = LinearBU.

6.4.2 (n, k)-Consensus Halving is in ETR

The existential theory of the reals consists of all true existentially quantified formulae using the connectives $\{\wedge, \vee, \neg\}$ over polynomials compared with the operators $\{<, \leq, =, \geq, >\}$. The complexity class ETR captures all problems that can be reduced in polynomial time to the existential theory of the reals.

We prove that (n, k)-Consensus Halving is in ETR. The reduction simply encodes the arithmetic circuits using ETR formulas, and then constrains $F_i(A_+) = F_i(A_-)$ for every agent i.

Theorem 22. (n, k)-Consensus Halving *is in* ETR.

Proof. The first step is to argue that an arithmetic circuit can be implemented as an ETR formula. Let $(V, \mathcal{T})$ be the arithmetic circuit. For every vertex $v \in V$ we introduce a new variable x_v. For every gate $g \in \mathcal{T}$ we introduce a constraint. For the gates in the set $\{G_\zeta, G_+, G_-, G_{*\zeta}, G_*\}$ the constraints simply implement the gate directly, eg., for a gate $G_+(v_{\text{in}1}, v_{\text{in}2}, v_{\text{out}})$ we use the constraint $x[v_{\text{out}}] = x[v_{\text{in}1}] + x[v_{\text{in}2}]$. For a gate $G_{\max}(v_{\text{in}1}, v_{\text{in}2}, v_{\text{out}})$ we use the formula

$$\big((x[v_{\text{out}}] = x[v_{\text{in}1}]) \wedge (x[v_{\text{in}1}] \geq x[v_{\text{in}2}])\big) \vee \big((x[v_{\text{out}}] = x[v_{\text{in}2}]) \wedge (x[v_{\text{in}2}] \geq x[v_{\text{in}1}])\big),$$

and likewise for a gate $G_{\min}(v_{in1}, v_{in2}, v_{out})$ we use the formula

$$\big((x[v_{\text{out}}] = x[v_{\text{in}1}]) \wedge (x[v_{\text{in}1}] \leq x[v_{\text{in}2}])\big) \vee \big((x[v_{\text{out}}] = x[v_{\text{in}2}]) \wedge (x[v_{\text{in}2}] \leq x[v_{\text{in}1}])\big).$$

Taking the conjunction of the constraints for each of the gates yields an ETR formula that implements the circuit.

Now we perform the reduction from consensus halving to the existential theory of the reals. Suppose that we have been given, for each agent i, an arithmetic circuit F_i implementing the integral of agent i's valuation function. We have already shown in the proof of Theorem 21 that, given a description of a k-cut given as a point in S^k, we can create a circuit implementing $F_i(A_+)$ and a circuit implementing $F_i(A_-)$ for each agent i. We also argued in that proof that $\sum_{j=1}^{k+1} |x_j|$ can be implemented as an arithmetic circuit. Our ETR formula is as follows.

$$\exists x \cdot \left(\bigwedge_{i=1}^{n} F_i(A_+) = F_i(A_-) \right) \wedge \sum_{j=1}^{k+1} |x_j| = 1$$

The first set of constraints ensure that x is a solution to the consensus halving problem, and the final constraint ensures that $x \in S^k$. □

Using the same technique, we can also reduce Borsuk-Ulam to an ETR formula. In this case, we get an ETR formula that always has a solution, and so this result places the problem in TFETR, which is the subclass of ETR in which the formula is guaranteed to be true.

Theorem 23. BU $\subseteq$ TFETR.

Proof. The proof is essentially identical to the proof of Theorem 22, and the only difference is that instead of starting with a consensus halving instance, we start with an arbitrary arithmetic circuit representing the function $f : S^d \to \mathbb{R}^d$, for which we wish to find a point x satisfying $f(x) = f(-x)$. We implement the arithmetic circuit in the same way as in Theorem 22, and our ETR formula is:

$$\exists x \cdot \left(\bigwedge_{i=1}^{d} f_i(x) = f_i(-x) \right) \wedge \sum_{j=1}^{d+1} |x_j| = 1.$$

□

6.5 Hardness Results for Consensus Halving

In this section we prove that (n, n)-Consensus Halving is FIXP-hard and that $(n, n-1)$-Consensus Halving is ETR-hard. These two reductions share a common step of embedding an arithmetic circuit into a consensus halving instance. So we first describe this step, and then move on to proving the two individual hardness results.

6.5.1 Embedding a circuit in a Consensus Halving instance: an outline

Our approach is inspired by [70], who provided a reduction from ϵ-GCircuit [41,122] to approximate consensus halving. However, our construction deviates significantly from theirs due to several reasons.

Firstly, the reduction in [70] works *only* for approximate consensus halving. Specifically, some valuations used in that construction have the form of $1/\epsilon$, where ϵ is the approximation

Special Gate	Constraint	Ranges
$G_{()^2}(v_{in}, v_{out})$	$x[v_{out}] = (x[v_{in}])^2$	$x[v_{in}] \in [0,1]$
$G_{*2}^{[0,1]}(v_{in}, v_{out})$	$x[v_{out}] = x[v_{in}] \cdot 2$	$x[v_{in}] \in [0,1/2]$
$G_{-}^{[0,1]}(v_{in1}, v_{in2}, v_{out})$	$x[v_{out}] = \max\{x[v_{in1}] - x[v_{in2}], 0\}$	$x[v_{in1}], x[v_{in2}] \in [0,1]$

Table 6.2: The special types of gates, their constraints and ranges of input.

guarantee, so the construction is not well-defined when $\epsilon = 0$ as it is in our case. Many of the gate gadgets used in [70] cannot be used due to this issue, including the max gate, which is crucially used in that construction to ensure that intermediate values do not get too large. We provide our own implementations of the broken gates. Our gate gadgets only work when the inputs and outputs lie in the range $[0, 1]$, and so we must carefully construct circuits for which this is always the case. The second major difference is that the reduction in [70] does not provide any method of multiplying two variables, which is needed in our case. We construct a gadget to do this, based on a more primitive gadget for squaring a single variable.

Special circuit. Our reduction from an arithmetic circuit to consensus halving will use a very particular subset of gates. Specifically, we will not use $G_{\min}$, $G_{\max}$, or G_*, and we will restrict $G_{*\zeta}$ so that ζ must lie in $(0, 1]$. We do however introduce three new gates, shown in Table 6.2. The gate $G_{()^2}$ squares its input, the gate $G_{*2}^{[0,1]}$ multiplies its input by two, but requires that the input be in $[0, 1/2]$, and the gate $G_{-}^{[0,1]}$ is a special minus gate that takes as inputs $a, b \in [0, 1]$ and outputs $\max\{a - b, 0\}$.

We note that $G_{\min}$, $G_{\max}$, and G_* can be implemented in terms of our new gates according to the following identities.

$$\max\{a,b\} = \frac{a+b}{2} + \frac{|a-b|}{2} = \frac{a}{2} + \frac{b}{2} + \frac{1}{2}\max\{a-b,0\} + \frac{1}{2}\max\{b-a,0\},$$
$$\min\{a,b\} = \frac{a+b}{2} - \frac{|a-b|}{2} = \frac{a}{2} + \frac{b}{2} - \frac{1}{2}\max\{a-b,0\} - \frac{1}{2}\max\{b-a,0\},$$
$$a \cdot b = 2\left[\left(\frac{a}{2} + \frac{b}{2}\right)^2 - \left(\left(\frac{a}{2}\right)^2 + \left(\frac{b}{2}\right)^2\right)\right].$$

Also, a very important requirement of the special circuit is that both inputs of any G_+ gate are in $[0, 1/2]$. To make sure of that, we downscale the inputs before reaching the gate,

and upscale the output, using the fact that $a + b = (a/2 + b/2) \cdot 2$.

The reduction to consensus halving. The reduction follows the general outline of the reduction given in [70]. The construction is quite involved, and so we focus on the high-level picture here.

Each gate is implemented by 4 agents, namely ad, mid, cen, ex in the consensus halving instance. The values computed by the gates are encoded by the positions of the cuts that are required in order to satisfy these agents. Agent ad performs the exact mathematical operation of the gate, and feeds the outcome in mid, who "trims" it in accordance with the gate's actual operation. Then mid feeds her outcome to cen and ex, who make a copy of mid's correct value of the gate, with "negative" and "positive" labels respectively. This value with the appropriate label will be input to other gates.

The most important agents are the ones that perform the mathematical operation of each gate, i.e. agents ad. Figure 6.1 shows the part of the valuation functions of these agents that perform the operation. Each figure shows a valuation function for one of the agents, meaning that the blue regions represent portions of the object that the agent desires. The agent's valuation for any particular interval is the integral of this function over that interval.

To understand the high-level picture of the construction, let us look at the construction for $G_{*\zeta}$. The precise valuation functions of the agents in the construction (see (6.1)) ensure that there is exactly one *input* cut in the region v^+_{in}. The leftmost piece due to that cut in that region will belong to A_+, while the rightmost will belong to A_-. It is also ensured that there is exactly one *output* cut in the region v^a_{out}, and that the first piece in that region will belong to A_- and the second will belong to A_+.

Suppose that gate g_i in the circuit is of type $G_{*\zeta}$ and we want to implement it through a Consensus Halving instance. If we treat v^+_{in} and v^a_{out} in Figure 6.1 as representing $[0, 1]$, then agent ad_i will take as input a cut at point $x \in v^+_{in}$. In order to be satisfied, ad_i will impose a cut at point $y \in v^a_{out}$, such that $F_i(A_+) = F_i(A_-)$, where: $F_i(A_+) = x + (\zeta - y)/\zeta$ and $F_i(A_-) = (1 - x) + y/\zeta$. Simple algebraic manipulation can be used to show that ad_i is satisfied only when $y = \zeta \cdot x$, as required.

We show that the same property holds for each of the gates in Figure 6.1. Two notable constructions are for the gates $G_{()^2}$ and $G^{[0,1]}_{-}$. For the gate $G_{()^2}$ the valuation function of agent ad is non-constant, which is needed to implement the non-linear squaring function. For the gate $G^{[0,1]}_{-}$, note that the output region v^a_{out} only covers half of the possible output

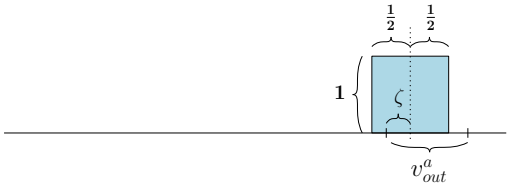
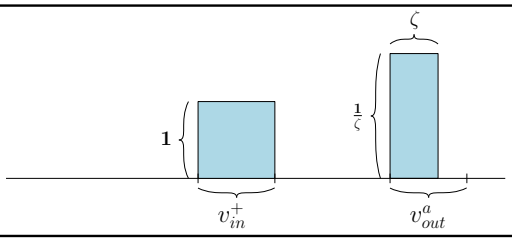
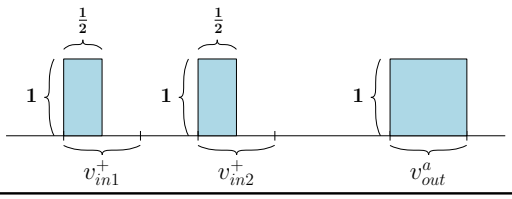
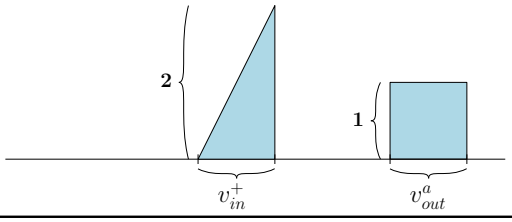
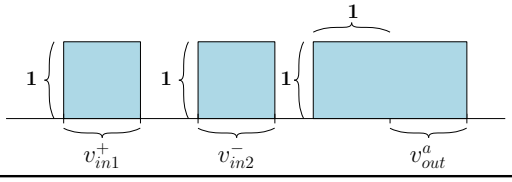
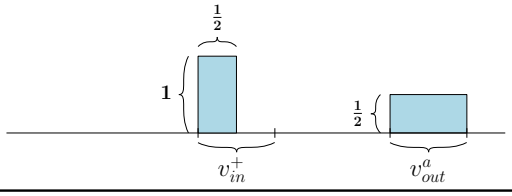
Gate	$\mathbf{G_\pi(t)}$	Valuation function
$\mathbf{G_\zeta}$	$\begin{cases} 1 & \textbf{if } t \in [v^a_{out,l} + \zeta - \frac{1}{2}, v^a_{out,l} + \zeta + \frac{1}{2}] \\ 0 & \textbf{otherwise} \end{cases}$	
$\mathbf{G_{*\zeta}}$	$\begin{cases} 1 & \textbf{if } t \in v^+_{in} \\ 1/\zeta & \textbf{if } t \in [v^a_{out,l}, v^a_{out,l} + \zeta] \\ 0 & \textbf{otherwise} \end{cases}$	
$\mathbf{G_+}$	$\begin{cases} 1 & \textbf{if } t \in [v^+_{in1,l}, v^+_{in1,l} + \frac{1}{2}] \\ 1 & \textbf{if } t \in [v^+_{in2,l}, v^+_{in2,l} + \frac{1}{2}] \\ 1 & \textbf{if } t \in v^a_{out} \\ 0 & \textbf{otherwise} \end{cases}$	
$\mathbf{G_{()^2}}$	$\begin{cases} 2(t - v^+_{in,l}) & \textbf{if } t \in v^+_{in} \\ 1 & \textbf{if } t \in v^a_{out} \\ 0 & \textbf{otherwise} \end{cases}$	
$\mathbf{G^{[0,1]}_-}$	$\begin{cases} 1 & \textbf{if } t \in v^+_{in1} \\ 1 & \textbf{if } t \in v^-_{in2} \\ 1 & \textbf{if } t \in [v^a_{out,l} - 1, v^a_{out,r}] \\ 0 & \textbf{otherwise} \end{cases}$	
$\mathbf{G^{[0,1]}_{*2}}$	$\begin{cases} 1 & \textbf{if } t \in [v^+_{in,l}, v^+_{in,l} + \frac{1}{2}] \\ 1/2 & \textbf{if } t \in v^a_{out} \\ 0 & \textbf{otherwise} \end{cases}$	

Figure 6.1: Gates and their corresponding functions $G_\pi(t)$.

space. The idea is that if the result of $x[v_{in1}] - x[v_{in2}]$ is negative, then the output cut will lie before the output region, which will be interpreted as a zero output by agents mid, cen, ex in the construction. On the other hand, if the result is positive, it will lie in the usual output range, and will be interpreted as a positive number. An example where $x[v_{in1}] = 1/4$ and $x[v_{in2}] = 3/4$ is shown in Figure 6.2.

Ultimately, this allows us to construct a Consensus Halving instance that implements this circuit. This means that for any $x \in [0,1]^n$, we can encode x as a set of cuts, which then force cuts to be made at each gate gadget that encode the correct output for that gate.

Lemma 19. *Suppose that we are given an arithmetic circuit with the following properties.*

- *The circuit uses the gates* $G_\zeta, G_+, G_{*\zeta}, G_{()^2}, G_-^{[0,1]}, G_{*2}^{[0,1]}$.
- *Every* G_ζ *and* $G_{*\zeta}$ *has* $\zeta \in \mathbb{Q} \cap (0,1]$.
- *For every input* $x \in [0,1]^n$*, all intermediate values computed by the circuit lie in* $[0,1]$.

We can construct a Consensus Halving *instance that implements this circuit.*

6.5.2 Proof of Lemma 19

6.5.2.1 Special circuit to Consensus Halving instance

Consider a circuit $H = (V, \mathcal{T})$ that uses gates in $\{G_\zeta, G_+, G_{*\zeta}, G_{()^2}, G_-^{[0,1]}, G_{*2}^{[0,1]}\}$, with $\zeta \in \mathbb{Q} \cap (0,1]$, each gate's inputs/output are in $[0,1]$, and both inputs of G_+ are in $[0,1/2]$. The constraints of the special gates $G_{()^2}, G_-^{[0,1]}, G_{*2}^{[0,1]}$ are shown in Table 6.2.

In general, the input of H is a N-dimensional vector $x \in [0,1]^N$ is given by N nodes with in-degree 0 and out-degree 1, called *input-nodes*. Also, in general, the output of H is a M-dimensional vector $x' \in [0,1]^M$ (the dimension of the circuit's output is of no importance here). Moreover, it could be the case that H is *cyclic*, meaning that it has no input and no output, but here we will consider the general case. Without loss of generality, let the rest of the nodes be of in-degree 1 and out-degree 1, located right after each gate's output. By "right after" we mean that if a gate's output has a branching, the node is placed before the branching. Suppose that the total number of nodes in H is $r := N + |\mathcal{T}| = poly(N)$, since by definition H has polynomial size.

If the node $v_i \in V$ for $i \in [r]$ is at the output of gate g_i we will call it the *output-node of* g_i (otherwise it will be an input-node). For an example see Figure 6.3.

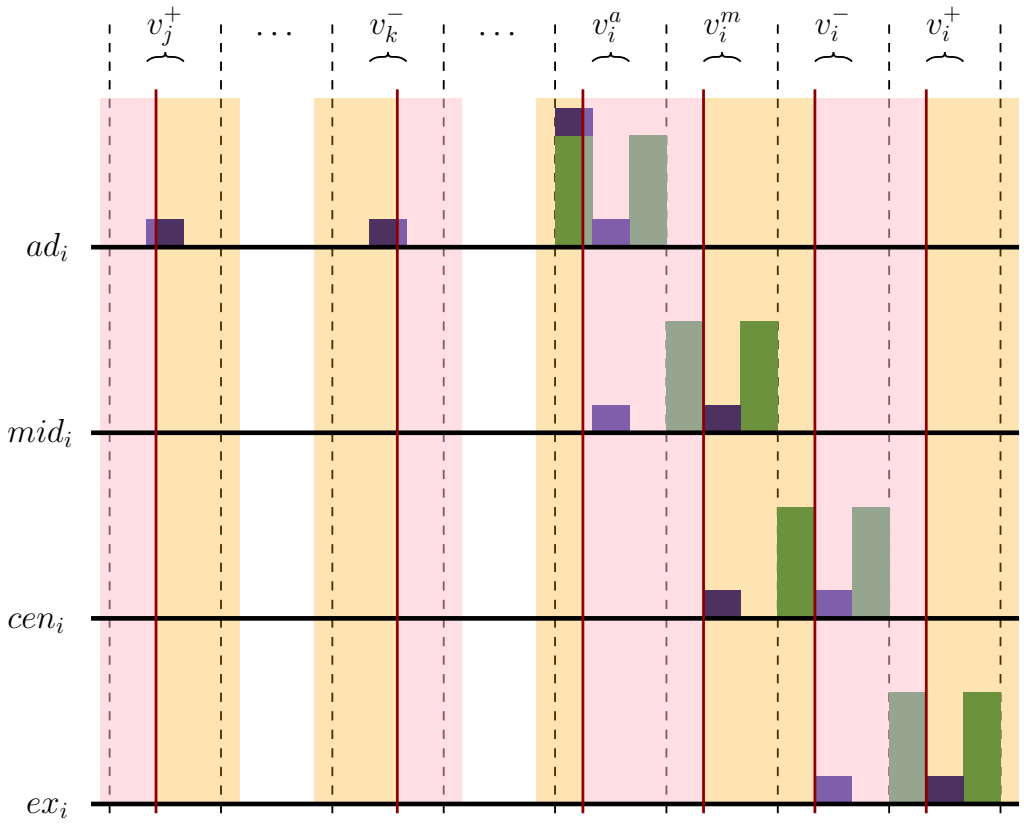


Figure 6.2: An example where the computation at the output $v_{out} := v_i$ of a $G_{-}^{[0,1]}$ gate with inputs $v_{in1} := v_j$ and $v_{in2} := v_k$ is simulated by the Consensus Halving instance. Here $x[v_j] = 1/4$ and $x[v_k] = 3/4$, hence $x[v_i] = 0$. The information about the values of the inputs is encoded by the cuts (red lines) in intervals v_j^+, and v_k^- imposed by agents ex_j and cen_k respectively. The blue and green shapes depict the area below the valuation function of each of the 4 agents. The pink regions have label "+" while the yellow have label "−". Agent ad_i performs the subtraction, by demanding that she is satisfied, and places a cut 1/10 to the left of the left endpoint of interval v_i^a. Then agent mid_i gets satisfied by placing a cut at exactly the left endpoint of interval v_i^m, thus encoding the value 0 which is the correct output value of the gate. Finally, agents cen_i, ex_i copy this value by enforcing similar cuts at the left endpoints of intervals v_i^- and v_i^+ respectively. The encoded values in the latter two intervals are the "negative" and "positive" version of $x[v_i]$.

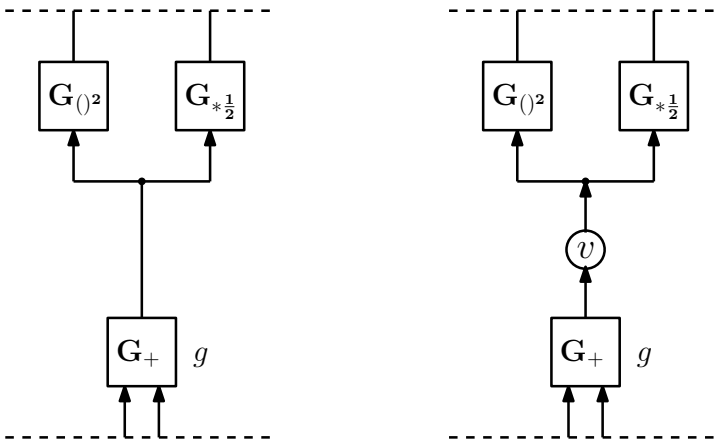


Figure 6.3: Before (leftmost figure) and after (rightmost figure) the creation of a node in series with the output of an addition gate. v is the *output-node* of g.

Consider the node v_i, the output-node of gate g_i. v_i corresponds to 4 consensus halving agents, named ad_i, mid_i, cen_i and ex_i. Player ad_i (Latin for "to") represents the incoming edge *to* node v_i and agent ex_i (Latin for "from") the outgoing edge *from* v_i, while both mid_i and cen_i represent an edge at the *middle* (*center*) of node v_i that connects its input and output. The number of agents created in H is $n := 4r$. The domain of the valuation functions of the agents is $[0, 12r]$. Furthermore, this interval is split to r blocks, with the i-th block being $[b_i, b_{i+1}]$, where $b_i := 12(i-1)$, $i \in [r]$.

According to the definition of the Consensus Halving problem, the domain of the valuation functions of the agents is $[0, 1]$. Although the domain of the valuation functions of the Consensus Halving instance that we reduce to is $[0, 12r]$, this is just for convenience of presentation. In fact, by scaling down each block to length $1/(12r)$ (divide by $12r$), the domain becomes $[0, 1]$ and the correctness of the reduction is preserved.

Let us define the function $border_i(t)$, $t \in [0, 12r]$ for each node v_i, $i \in [r]$. The idea for this function is from [70]. If v_i is the output-node of gate type $G_{*\zeta}$, then

$$border_i(t) = \begin{cases} 4, & t \in [b_i, b_i + 1] \cup [b_i + 1 + \zeta, b_i + 2 + \zeta] \\ 0, & \text{otherwise} \end{cases}$$

If v_i is the output-node of any gate type other than $G_{*\zeta}$, then

$$border_i(t) = \begin{cases} 4, & t \in [b_i, b_i+1] \cup [b_i+2, b_i+3] \\ 0, & \text{otherwise} \end{cases}$$

and also:

- $v_i^a := [b_i+1, b_i+2] := [v_{i,l}^a, v_{i,r}^a]$
- $v_i^m := [b_i+4, b_i+5] := [v_{i,l}^m, v_{i,r}^m]$
- $v_i^- := [b_i+7, b_i+8] := [v_{i,l}^-, v_{i,r}^-]$
- $v_i^+ := [b_i+10, b_i+11] := [v_{i,l}^+, v_{i,r}^+]$
- $G_\pi(t)$ is the function corresponding to gate of type $G_\pi \in \{G_\zeta, G_{*\zeta}, G_+, G_{()^2}, G_-^{[0,1]}, G_{*2}^{[0,1]}\}$ (see Figure 6.1).

The valuation functions of the agents ad_i, mid_i, cen_i and ex_i corresponding to node v_i are,

$$ad_i(t) = \begin{cases} border_i(t) + G_\pi(t), & \text{if } v_i \text{ is the output-node of gate type } G_\pi \\ border_i(t), & \text{if } v_i \text{ is input-node (input of } H). \end{cases} \tag{6.1}$$

$$mid_i(t) = \begin{cases} 4, & t \in [b_i+3, b_i+4] \cup [b_i+5, b_i+6] \\ 1, & t \in v_i^a \cup v_i^m \\ 0, & \text{otherwise} \end{cases}$$

$$cen_i(t) = \begin{cases} 4, & t \in [b_i+6, b_i+7] \cup [b_i+8, b_i+9] \\ 1, & t \in v_i^m \cup v_i^- \\ 0, & \text{otherwise} \end{cases}$$

$$ex_i(t) = \begin{cases} 4, & t \in [b_i+9, b_i+10] \cup [b_i+11, b_i+12] \\ 1, & t \in v_i^- \cup v_i^+ \\ 0, & \text{otherwise} \end{cases}$$

The intuition for the synergy of the 4 agents is the following: Take as a given that in a solution of the created Consensus Halving instance with at most n cuts, a cut is placed

only (almost always[2]) in the intervals $v_i^a, v_i^m, v_i^-, v_i^+$ for every $i \in [r]$. Since the length of each of those intervals is 1, each such cut encodes a number in $[0,1]$. Consider v_i, the output-node of gate g_i with inputs v_j, v_k. Think of the agents ad_i, mid_i, cen_i, ex_i as being sequential, meaning that each of them "computes" a value through a cut in v_i^a, v_i^m, v_i^- or v_i^+ respectively, and feeds it in the next agent. In particular, agent ad_i takes as input the values (in the form of cuts) that nodes v_j, v_k give her, and computes the exact operation that g_i prescribes (e.g. if g_i is type $G_-^{[0,1]}$, ad_i performs subtraction of the input values without capping at 0). Then ad_i feeds this value in mid_i via creating a cut in v_i^a, and mid_i computes the actual value in $[0,1]$ that g_i should output (e.g. if g_i is type $G_-^{[0,1]}$, in this step mid_i caps the value at 0), and feeds it in cen_i via creating a cut in v_i^m. This correct value should be exported for further use from other gates to which v_i is input, but depending on these gates, the positive or negative of that value might be needed (by "positive" and "negative" we mean the label, not the actual sign of the value). That is why a negative version of this value is produced by cen_i and a positive by ex_i, via a cut in v_i^- and v_i^+ respectively. A negative(resp. positive) value is one encoded by a cut that defines an interval at its left which is *negative*(resp. *positive*). Moreover, for every input-node v_j we arbitrarily consider ad_j to encode a negative value, and since (by the structure of the Consensus Halving instance) the labels of the values induced by the 4 agents are alternating, always cen_i(resp. ex_i) encodes a negative(resp. positive) value.

6.5.2.2 1-1 correspondence of circuit values to Consensus Halving cuts

Let us define the functions $z_i(x)$, $i \in [r]$ that depend on the input vector $x \in [0,1]^N$, and compute the value of each node v_i of the arithmetic circuit H. Let us also arbitrarily set $(z_1, \ldots, z_N) := (x_1, \ldots, x_N)$. First, we will show that for every tuple $(z_1(x), \ldots, z_r(x))$ of values that satisfy H, a solution in the constructed Consensus Halving instance with n agents and n cuts ($n := 4r$) encodes the same values via its cuts. We will then show that for every solution of the Consensus Halving instance with n agents and n cuts, the cuts correspond to a unique tuple $(z_1, \ldots, z_r)$ that satisfies H. In the sequel, we call a cut t *negative*(resp. *positive*) if the interval that it defines at its left has negative(resp. positive) label. Without loss of generality, let the interval at the left of the first cut to be a negative interval.

[2]With the only exception being a cut before v_i^a when gate g_i is $G_-^{[0,1]}$ and its result is negative. See Figure 6.2 for an example

Circuit values to cuts. Suppose the tuple $(z_1^*, \ldots, z_r^*)$ satisfies H. We will show that from this solution we can create a Consensus Halving solution with $n := 4r$ cuts, i.e. all of the agents are satisfied. Consider node v_i of H. Let us translate the values z_i^*, $i \in [r]$ into cuts as follows:

- If g_i's type is one of $G_\zeta, G_{*\zeta}, G_+, G_{()^2}, G_{*2}^{[0,1]}$ or v_i is an input-node.
  - Place a cut at $t = v_{i,l}^a + z_i^*$,
  - Place a cut at $t = v_{i,l}^m + z_i^*$,
  - Place a cut at $t = v_{i,l}^- + z_i^*$,
  - Place a cut at $t = v_{i,l}^+ + z_i^*$.
- If g_i's type is $G_-^{[0,1]}$, i.e. $g_i = \max\{g_j - g_k, 0\}$, and $z_j^* \geq z_k^*$.
  - Place a cut at $t = v_{i,l}^a + z_i^*$,
  - Place a cut at $t = v_{i,l}^m + z_i^*$,
  - Place a cut at $t = v_{i,l}^- + z_i^*$,
  - Place a cut at $t = v_{i,l}^+ + z_i^*$.
- If g_i's type is $G_-^{[0,1]}$, i.e. $g_i = \max\{g_j - g_k, 0\}$, and $z_j^* < z_k^*$
  - Place a cut at $t = v_{i,l}^a - (z_k^* - z_j^*)/5$,
  - Place a cut at $t = v_{i,l}^m + z_i^*$,
  - Place a cut at $t = v_{i,l}^- + z_i^*$,
  - Place a cut at $t = v_{i,l}^+ + z_i^*$.

By construction of the valuation functions of the agents, these cuts are placed one after the other, and therefore they alternate between "negative" and "positive", starting with negative. Let us now prove that for every $i \in [r]$, the ad_i agent is satisfied.

$\mathbf{G}_\zeta$: This gate has no input. Consider its output $z_i^* = \zeta$ and its output-node v_i. By our constructed n-cut, a cut is placed at $t = v_{i,l}^a + \zeta$, which cuts exactly in half the total valuation of ad_i in v_i^a (see Figure 6.1). Since the valuation function is symmetric around v_i^a (see (6.1)), agent ad_i is satisfied.

$\mathbf{G}_{*\zeta}$: Consider its input z_j^*, output $z_i^* = \zeta \cdot z_j^*$ and its output-node v_i. By our constructed n-cut, a positive cut is placed at $t = v_{j,l}^+ + z_j^*$ and a negative cut is placed at $t = v_{i,l}^a + z_i^*$. The valuation function is symmetric around v_i^a (see (6.1)), therefore, in order for ad_i to be satisfied, it suffices that $z_j^* \cdot 1 + (\zeta - z_i^*) \cdot \frac{1}{\zeta} = (1 - z_j^*) \cdot 1 + z_i^* \cdot \frac{1}{\zeta}$, which is true.

$\mathbf{G}_{+}$: Consider its inputs z_j^*, z_k^*, its output $z_i^* = z_j^* + z_k^*$ and its output-node v_i. By our constructed n-cut, a positive cut is placed at $t = v_{j,l}^+ + z_j^*$, another positive cut is placed at $t = v_{k,l}^+ + z_k^*$ and a negative cut is placed at $t = v_{i,l}^a + z_i^*$. The valuation function is symmetric around v_i^a (see (6.1)), therefore, in order for ad_i to be satisfied, it suffices that $z_j^* \cdot 1 + z_k^* \cdot 1 + (1 - z_i^*) \cdot 1 = (1/2 - z_j^*) \cdot 1 + (1/2 - z_k^*) \cdot 1 + z_i^* \cdot 1$, which is true.

$\mathbf{G}_{()^{\mathbf{2}}}$: Consider its input z_j^*, output $z_i^* = (z_j^*)^2$ and its output-node v_i. By our constructed n-cut, a positive cut is placed at $t = v_{j,l}^+ + z_j^*$ and a negative cut is placed at $t = v_{i,l}^a + z_i^*$. The valuation function is symmetric around v_i^a (see (6.1)), therefore, in order for ad_i to be satisfied, it suffices that $(z_j^*)^2 + (1 - z_i^*) \cdot 1 = (1 - (z_j^*)^2) + z_i^* \cdot 1$, which is true.

$\mathbf{G}_{*\mathbf{2}}^{[\mathbf{0},\mathbf{1}]}$: Consider its input z_j^*, output $z_i^* = 2 \cdot z_j^*$ and its output-node v_i. By our constructed n-cut, a positive cut is placed at $t = v_{j,l}^+ + z_j^*$ and a negative cut is placed at $t = v_{i,l}^a + z_i^*$. The valuation function is symmetric around v_i^a (see (6.1)), therefore, in order for ad_i to be satisfied, it suffices that $z_j^* \cdot 1 + (1 - z_i^*) \cdot \frac{1}{2} = (1/2 - z_j^*) \cdot 1 + z_i^* \cdot \frac{1}{2}$, which is true.

$\mathbf{G}_{-}^{[\mathbf{0},\mathbf{1}]}$: Consider its inputs z_j^*, z_k^*, its output $z_i^* = \max\{z_j^* - z_k^*, 0\}$ and its output-node v_i. By our constructed n-cut,

- if $z_j^* \geq z_k^*$, then $z_i^* = z_j^* - z_k^*$. By our constructed n-cut, a positive cut is placed at $t = v_{j,l}^+ + z_j^*$, a negative cut is placed at $t = v_{k,l}^- + z_k^*$ and another negative cut is placed at $t = v_{i,l}^a + z_i^*$. In order for ad_i to be satisfied, it suffices that $z_j^* \cdot 1 + (1 - z_k^*) \cdot 1 + (1 - z_i^*) \cdot 1 + 4 = (1 - z_j^*) \cdot 1 + z_k^* \cdot 1 + (4 + 1) + z_i^* \cdot 1$, which is true.

- if $z_j^* < z_k^*$, then $z_i^* = 0$. By our constructed n-cut, a positive cut is placed at $t = v_{j,l}^+ + z_j^*$, a negative cut is placed at $t = v_{k,l}^- + z_k^*$ and another negative cut is placed at $t = v_{i,l}^a - (z_k^* - z_j^*)/5$. In order for ad_i to be satisfied, it suffices that $z_j^* \cdot 1 + (1 - z_k^*) \cdot 1 + \frac{z_k^* - z_j^*}{5} \cdot (4 + 1) + 1 + 4 = (1 - z_j^*) \cdot 1 + z_k^* \cdot 1 + (1 - \frac{z_k^* - z_j^*}{5}) \cdot (4 + 1)$, which is true.

We will now prove that in our constructed n-cut, the agents mid_i, cen_i, ex_i are also satisfied. If g_i is not a $G_-^{[0,1]}$ gate, let us prove that mid_i is satisfied. In our n-cut there is a negative cut at $t = v_{i,l}^a + z_i^*$ and a positive one at $t = v_{i,l}^m + z_i^*$. In order for mid_i to be satisfied, it suffices that $z_i^* \cdot 1 + (1 - z_i^*) \cdot 1 = (1 - z_i^*) \cdot 1 + z_i^* \cdot 1$, which is true. The proof of satisfaction of agents mid_i, cen_i, ex_i is similar, since the succeeding agent's valuation function is the same as the preceding agent's one, shifted 3 units.

If g_i is a $G_-^{[0,1]}$ gate, let us prove that mid_i is satisfied.

- if $z_j^* \geq z_k^*$, then a negative cut is placed at $t = v_{i,l}^a + z_i^*$, and a positive cut is placed at $t = v_{i,l}^m + z_i^*$. In order for mid_i to be satisfied, it suffices that $z_i^* \cdot 1 + (1 - z_i^*) \cdot 1 = (1 - z_i^*) \cdot 1 + z_i^* \cdot 1$, which is true.

- if $z_j^* < z_k^*$, then a negative cut is placed at $t = v_{i,l}^a - (z_k^* - z_j^*)/5$ and a positive cut is placed at $t = v_{i,l}^m$. In order for mid_i to be satisfied, it suffices that $\frac{z_k^* - z_j^*}{5} \cdot 0 + 1 \cdot 1 = 1 \cdot 1$, which is true.

For the agents cen_i and ex_i, it is easy to see that due to their valuation functions, the n-cut we provide forces them to have positive total valuation equal to the negative one.

Cuts to circuit values. Now suppose that the tuple $(t_1^*, \ldots, t_n^*)$ with $0 \leq t_1^* \leq \cdots \leq t_n^* \leq 12r$, represents a n-cut ($n := 4r$) that is a solution of the constructed Consensus Halving instance with n agents, where w.l.o.g. the first $4N$ cuts correspond to the N input-nodes. We will show that from this solution we can construct a tuple $(z_1, \ldots, z_r)$ that satisfies circuit H.

Consider node v_i which is the output-node of gate g_i or it is an input-node. Observe that the valuation function of each of ad_i, mid_i, cen_i and ex_i has more than half of her total valuation inside the interval $[b_i, b_i + 3], [b_i + 3, b_i + 6], [b_i + 6, b_i + 9]$ and $[b_i + 9, b_i + 12]$ respectively. This means that in a solution, each of them has to have at least one cut in her corresponding aforementioned interval. But since these intervals are not overlapping for all n agents, and we need to have at most n cuts, exactly one cut has to be placed by each agent in her corresponding interval.

Consider now the first $4N$ cuts that correspond to the input-nodes. As it is apparent from the definition of these nodes' valuation functions, each agent of ad_i, mid_i, cen_i, ex_i for

$i \in [N]$ has to place her single cut in the interval $v_i^a, v_i^m, v_i^-, v_i^+$ respectively. Given the latter fact, the definition of valuation functions for non input-node agents dictates that there will always be a cut in v_i^+ for every $i \in [r]$. Since $0 \leq t_1^* \leq \cdots \leq t_n^* \leq 12r$, the sequential nature of our agents indicates that the cut t_{4i}^*, i.e. with index $4 \cdot i$, is found in interval v_i^+. Now, let us translate the position of the cut $t_{4i}^*, i \in [r]$ into the value $z_i = t_{4i}^* - v_{i,l}^+$. By a similar argument as that of the previous paragraph showing that the ad_i agents are satisfied, it is easy to see that, by the aforementioned translation, the created tuple $(z_1, \ldots, z_r)$ satisfies circuit H.

Valuation functions to circuits. In the Consensus Halving instances we construct, we have described the valuation functions of the agents mathematically. However, in a Consensus Halving instance the input is an arithmetic circuit, therefore we have to turn each valuation function of each agent $j \in [n]$ into its integral, and subsequently into an arithmetic circuit. Here we describe a method to do that.

The valuation functions we construct in our reduction (see Section 6.5.2.1) are piecewise polynomial functions of a single variable and their degree is at most 1, with k pieces where k is constant. Therefore, their integrals, which are the input of the Consensus Halving problem (captured by arithmetic circuits), are piecewise polynomial functions (with the same pieces) with degree at most 2. Consider the valuation function f of an arbitrary player. Let the pieces of f be $[p_0, p_1), [p_1, p_2), \ldots, [p_{k-1}, p_k]$ where $p_0 = 0$ and $p_k = 1$ and denote $P_1, P_2, \ldots, P_k$ the above pieces respectively. Let us also denote by f^{P_s} the polynomial in interval P_s, $s \in \{1, 2, \ldots, k\}$. In particular, f can be defined as

$$f(t) = \begin{cases} f^{P_1}(t) & , t \in [p_0, p_1) \\ f^{P_2}(t) & , t \in [p_1, p_2) \\ \vdots & \\ f^{P_k}(t) & , t \in [p_{k-1}, p_k], \end{cases} \tag{6.2}$$

and according to the valuation functions used in the reduction (see Section 6.5.2.1), for any given piece P_s there are two kinds of possible functions

(a) $f^{P_s}(t) = c_s$, where $c_s \geq 0$ is a constant, or

(b) $f^{P_s}(t) = 2 \cdot (t - p_{s-1})$.

(The latter comes from the valuation function of an *ad* agent that corresponds to an output node of a $G_{()^2}$ gate.)

We would like to find a formula for the integral of $f(t)$, denoted $F(t)$, and we also require that $F(t)$ is computable by an arithmetic circuit, so that it is a proper input (together with the other agents' integrals of valuation functions) to the Consensus Halving instance. For each piece P_s we will construct an integral, denoted by $F^{P_s}(t)$, such that each such integral will be computable by an arithmetic circuit, and so that it will be $F(t) = \sum_{s\in\{1,2,\ldots,k\}} F^{P_s}(t)$. First, let us construct the function $D_s(t)$ using the domain P_s of $f^{P_s}(t)$:

$$D_s(t) := \min\left\{\max\left\{t, p_{s-1}\right\}, p_s\right\},$$

which takes values

$$D_s(t) = \begin{cases} p_{s-1}, & t < p_{s-1} \\ t, & t \in [p_{s-1}, p_s] \\ p_s, & t > p_s. \end{cases}$$

Now, for function $f^{P_s}(t)$ of case (a), we construct its integral:

$$F^{P_s}(t) := c_s \cdot \left(D_s(t) - p_{s-1}\right),$$

which takes values

$$F^{P_s}(t) = \begin{cases} 0, & t < p_{s-1} \\ c_s \cdot \left(t - p_{s-1}\right), & t \in [p_{s-1}, p_s] \\ c_s \cdot \left(p_s - p_{s-1}\right), & t > p_s. \end{cases}$$

Similarly, for function $f^{P_s}(t)$ of case (b), we also construct its integral:

$$F^{P_s}(t) := \left(D_s(t) - p_{s-1}\right)^2,$$

which takes values

$$F^{P_s}(t) = \begin{cases} 0, & t < p_{s-1} \\ (t - p_{s-1})^2, & t \in [p_{s-1}, p_s] \\ (p_s - p_{s-1})^2, & t > p_s. \end{cases}$$

Finally, for the agent with valuation function $f(t)$, the corresponding function computable by the arithmetic circuit that is input to the Consensus Halving problem is:

$$F(t) := \sum_{s \in \{1,2,\ldots,k\}} F^{P_s}(t).$$

For the integral function $F(t)$ indeed it holds that $F(t) = \int_0^t f(x)\,dx$ as required. That is because, by the way we defined each $F^{P_s}(t)$, for any $t \in P_{s^*}$ it is

$$\begin{aligned} F(t) = \sum_{s \in \{1,2,\ldots,k\}} F^{P_s}(t) &= \sum_{s \in \{1,2,\ldots,s^*-1\}} F^{P_s}(t) + F^{P_s^*}(t) + \sum_{s \in \{s^*+1,\ldots,k\}} 0 \\ &= \sum_{s \in \{1,2,\ldots,s^*-1\}} \int_{P_s} f^{P_s}(x)\,dx + \int_{p_{s^*-1}}^{t} f^{P_s^*}(x)\,dx \\ &= \sum_{s \in \{1,2,\ldots,s^*-1\}} \int_{p_{s-1}}^{p_s} f(x)\,dx + \int_{p_{s^*-1}}^{t} f(x)\,dx \\ &= \int_0^t f(x)\,dx \end{aligned}$$

For each player with some valuation function f as defined above, we can compute the functions F^{P_s}, $s \in [k]$ by using gates $G_\zeta, G_{*\zeta}, G_-, G_*, G_{min}, G_{max}$. Then $F(t)$ can be computed by using G_+ gates. The arithmetic circuits that compute the functions $F(t)$ (one for each agent $j \in [n]$) constitute a proper Consensus Halving instance. This completes the proof of Lemma 19.

6.5.3 (n, n)-Consensus Halving is FIXP-hard

We show that (n, n)-Consensus Halving is FIXP-hard by reducing from the problem of finding a Nash equilibrium in a d-player game for $d \geq 3$, which is known to be FIXP-complete [66]. As shown in [66], this problem can be reduced to the Brouwer fixed point

problem: given an arithmetic circuit computing a function $F : [0,1]^n \to [0,1]^n$, find a point $x \in [0,1]^n$ such that $F(x) = x$. In a similar way to [70], we take this circuit, with the outputs looped back to the inputs, and embed it into a consensus halving instance. Since Lemma 19 implies that our implementation of the circuit is correct, this means that any solution to the consensus halving problem must encode a point x satisfying $F(x) = x$.

One difficulty is that we must ensure that the arithmetic circuit that we build falls into the class permitted by Lemma 19. To do this, we carefully analyse the circuits produced in [66], and we modify them so that all of the preconditions of Lemma 19 hold. This gives us the following result.

Theorem 24. (n, n)-Consensus Halving *is* `FIXP`*-hard.*

6.5.3.1 Proof of Theorem 24

In [66] it is shown that the problem of finding a Nash equilibrium of a d-player normal form game with $d \geq 3$ ("d-player Nash equilibrium" problem) is `FIXP`-complete. Given an instance of this problem, we will construct a polynomial-time reduction to (n, n)-Consensus Halving. We will start from an arbitrary instance of "d-player Nash equilibrium" and, according to it, design a circuit using only the gates $G_\zeta, G_+, G_-, G_*, G_{\max}, G_{\min}$ with $\zeta \in \mathbb{Q}$. This step is done by a straightforward application of the procedure described in the proofs of Lemma 4.5 and Lemma 4.6 in [66]. This circuit computes a function whose fixed points correspond precisely to the Nash equilibria of the initial game. Then, we create an equivalent circuit by "breaking down" the initial gates to some more suitable ones (by introducing "special gates", see Table 6.2), whose inputs and outputs are guaranteed to be in $[0, 1]$. From this, we will create a cyclic circuit, introduce consensus halving players on the "wires" of the circuit, and show that a consensus halving solution with at most as many cuts as the number of players in this instance can be efficiently translated back to a Nash equilibrium of the initial game.

6.5.3.1.1 Expressing the game as a circuit without division gates

Here, given an arbitrary d-player game, we will create a function whose fixed points are precisely the Nash equilibria of that game. Consider a given instance I of the "d-player Nash equilibrium" problem, i.e. a d-player normal form game where each player i has a set S_i of pure strategies. We will use the following notation similar to the one in [66]: $N_i := |S_i|$, $N := \sum_i^d N_i$ and v_i is the payoff function of player i with domain $D_I :=$

$\times_{i=1}^{d}\Delta_{N_i}$, where Δ_{N_i} is the unit (N_i-1)-simplex. Define the *mixed strategy profile* $x := (x_{11},\ldots,x_{1N_1},x_{21},\ldots,x_{2N_2},\ldots,x_{d1},\ldots,x_{dN_d})$ to be a N-dimensional vector with the entry x_{ij} being the probability that player $i\in[d]$ plays pure strategy $j\in S_i$. Also, $v(x)$ is an N-dimensional vector with entries indexed as in x, with $v_{ij}(x) := v_i(j,x_{-i})$, the latter being the expected payoff of player i when she plays the pure strategy $j\in S_i$ against the partial profile x_{-i} of the rest of the players. The payoff function of each player is normalized by a standard scaling in $[0,1]$ so that the Nash equilibria of the game are precisely the same. Thus, $v_{ij}(x)\in[0,1]$. Finally, let $h(x) := x + v(x)$.

Now, define for each player i the function $f_{i,x}(t) := \sum_{j\in S_i}\max(h_{ij}(x)-t,0)$ with parameter x. This function is defined in $\mathbb{R}$ and it is continuous, piecewise linear, strictly decreasing with values from 0 to $+\infty$, thus there is a unique value $t_i\in\mathbb{R}$ such that $f_{i,x}(t_i)=1$. The required function whose set of fixed points is identical to the set of Nash equilibria of instance I is $G_I(x)_{ij} := \max(h_{ij}(x)-t_i,0)$ for $i\in[d]$, $j\in S_i$. The function G_I takes as input the n-dimensional vector x and outputs an N-dimensional vector $G_I(x)$ with entries defined as above. By definition of G_I and choice of t_i, it is $\sum_{j\in S_i}G_I(x)_{ij}=1$ for every $i\in[d]$, and therefore G_I is a mapping of the domain D_I to itself.

Lemma 20 (LEMMA 4.5, [66])**.** *The fixed points of the function G_I are precisely the Nash equilibria of the game I.*

In fact, the structure of function G_I allows for it to be efficiently constructed using only the required types of gates.

Lemma 21 (LEMMA 4.6, [66])**.** *We can construct in polynomial time a circuit with basis $\{+,-,*,\max,\min\}$ (no division) and rational constants that computes the function G_I.*

For the proofs of the above lemmata the reader is referred to the indicated work by Etessami and Yannakakis.

In the proof of the latter lemma in [66] it is shown how to construct an arithmetic circuit C_I that computes the function G_I using only gates of type $G_\zeta, G_+, G_-, G_*, G_{\max}, G_{\min}$, where $\zeta\in\mathbb{Q}$. The construction of C_I is the following: Compute the function $y=h(x)=x+v(x)$ using only G_+, G_* type of gates, allowed by the definition of $v(x)$. Vector y has d sub-vectors, where $y_i=(y_{i1},y_{i2},\ldots,y_{iN_i})$. Then, each y_i is sorted using a sorting network Z_i thus creating a vector $z_i=(z_{i1},z_{i2},\ldots,z_{iN_i})$ with sorted entries $z_{i1}\geq z_{i2}\geq\cdots\geq z_{iN_i}$; sorting networks can be implemented in arithmetic circuits using only gates $G_{\max}, G_{\min}$ (for more see e.g. [92]). Using z_{ij}'s the function $t_i := \max_{l\in[N_i]}\left\{(1/l)*\left(\left(\sum_{j=1}^{l}z_{ij}\right)-1\right)\right\}$ is

computed and the final output of the whole circuit is

$$x'_{ij} := \max\{y_{ij} - t_i, 0\} \quad \text{for each } i \in [d], j \in S_i. \tag{6.3}$$

6.5.3.1.2 A circuit with gates whose inputs/outputs are in $[0,1]$

One can easily observe that some of the gates of circuit C_I may have inputs and outputs outside of $[0,1]$. For example, the G_+ gate that computes $y_{ij} = x_{ij} + v(x)_{ij}$ can be 2 and the arguments of $G_{\max}$ in t_i can be negative. We will transform this circuit into an equivalent one that guarantees its gates' inputs and outputs to be in $[0,1]$, using only gates $G_\zeta, G_+, G_-^{[0,1]}, G_*, G_{*2}^{[0,1]}, G_{\max}, G_{\min}$, where $\zeta \in \mathbb{Q} \cap (0,1]$, $G_-^{[0,1]}$ is a special subtraction gate that outputs 0 in case the subtraction results to a negative number, and $G_{*2}^{[0,1]}$ is a special multiplication gate that multiplies a single input in $[0, \frac{1}{2}]$ with 2 and its output is in $[0,1]$.

In particular, instead of constructing the circuit C_I as described in the previous paragraph, we will construct an equivalent one, called C'_I, whose input and output are the same as that of C_I, namely x_{ij} and x'_{ij}, $i \in [d]$, $j \in [N_i]$ respectively, but its gates have inputs/outputs in $[0,1]$. We do this by manipulating the formula for the required function G_I under computation, by suitably scaling up or down the input values of each gate, using additional gates $G_\zeta, G_+, G_-^{[0,1]}, G_*$.

We construct C'_I as follows: First, we compute the vector $p := h(x)/2 = x * \frac{1}{2} + v(x) * \frac{1}{2}$ using only G_+, G_* gates. Note that $x_{ij}, v_{ij}(x), p_{ij} \in [0,1]$, $\forall i \in [d], j \in S_i$ (recall that the payoff function is normalized in $[0,1]$). Then, we sort each of the sub-vectors p_i, $i \in [d]$ via a sorting network Q_i that can be constructed using $G_{\max}$ and $G_{\min}$ gates, thus computing the sorted vectors $q_i = (q_{i1}, q_{i2}, \ldots, q_{iN_i})$ with sorted entries $q_{i1} \geq q_{i2} \geq \cdots \geq q_{iN_i}$. Now, for every $i \in [d]$ and $l \in [N_i]$ we compute the following sub-function

$$t''_{il} := \frac{1}{2} * \frac{1}{l} * \sum_{j=1}^{l} q_{ij} + \frac{1}{2} - \frac{1}{4} * \frac{1}{l},$$

by using $l+1$ G_+ gates, 3 G_+ gates and 1 $G_-^{[0,1]}$ gate, where the subtraction gate is the last to take place. One should observe that since $\sum_{j=1}^{N_i} x_{ij} = 1$ and $\sum_{j=1}^{N_i} v_{ij}(x) \leq 1$ (by definition of normalized payoff function), it is $\sum_{j=1}^{N_i} q_{ij} \leq \frac{1}{2} \cdot (1+1) = 1$, therefore none of the individual computations of t''_{il} is outside $[0,1]$. Moreover, in the subtraction, the value of the subtrahend is at most the value of the minuend so the subtraction is precise (not capped at 0).

Now, for each $i \in [d]$ we compute the sub-function

$$t''_i := \max_{l \in [N_i]} \{t''_{il}\},$$

by using $N_i - 1$ $G_{\max}$ gates, and consequently compute

$$t'_i := \left(t''_i - \frac{1}{2}\right) * 2,$$

by using one $G_{-}^{[0,1]}$ and one special $G_{*2}^{[0,1]}$ gate and the computations happen from left to right. Note that $t''_i \geq 1/2$, therefore the subtraction is precise (not capped at 0). Also, note that, by definition of t''_{il}, it is $t''_i \leq 1$, therefore $t''_i - 1/2 \leq 1/2$ and the output of the $G_{*2}^{[0,1]}$ gate of t'_i is in $[0, 1]$. Finally, the output of the circuit C'_I is computed by

$$x'_{ij} := \max\{p_{ij} - t'_i, 0\} * 2, \quad \text{for each } i \in [d], j \in S_i, \tag{6.4}$$

using one $G_{-}^{[0,1]}$ and one special $G_{*2}^{[0,1]}$ gate.

Lemma 22. *Circuit C'_I is equivalent to C_I, i.e. it computes the function G_I.*

Proof. We will show that for every $i \in [d], j \in S_i$, the value x_{ij} of (6.4) is the same as that of (6.3), i.e. the output of the circuits C'_I and C_I is the exact same. Using the formulas for t'''_{il}, t''_i and t'_i, we can re-write algebraically x_{ij} by substituting the circuit's operations with the regular mathematical ones, i.e. $G_+, G_{-}^{[0,1]}, G_{*2}^{[0,1]}, G_*, G_{\max}, G_{\min}$ translate to $+, -, \cdot 2, \cdot, \max, \min$ respectively. Observe that this is possible since the $G_{-}^{[0,1]}$ gate, excluding the one in (6.4), actually performs subtraction without capping the output to 0.

Thus, starting from (6.4) we have

$$
\begin{aligned}
x'_{ij} &= \max\{p_{ij} - t'_i, 0\} \cdot 2 \\
&= \max\{2 \cdot p_{ij} - 2 \cdot t'_i, 0\} \\
&= \max\left\{y_{ij} - 4 \cdot \left(t''_i - \frac{1}{2}\right), 0\right\} \quad (y_{ij} \text{ from construction of } C_I) \\
&= \max\left\{y_{ij} - 4 \cdot \left(\max_{l \in [N_i]}\{t''_{il}\} - \frac{1}{2}\right), 0\right\} \\
&= \max\left\{y_{ij} - 4 \cdot \left(\max_{l \in [N_i]}\left\{\frac{1}{2l} \cdot \left(\sum_{j=1}^{l} q_{ij}\right) + \frac{1}{2} - \frac{1}{4l}\right\} - \frac{1}{2}\right), 0\right\} \\
&= \max\left\{y_{ij} - 4 \cdot \max_{l \in [N_i]}\left\{\frac{1}{2l} \cdot \left(\sum_{j=1}^{l} q_{ij}\right) - \frac{1}{4l}\right\}, 0\right\} \\
&= \max\left\{y_{ij} - \max_{l \in [N_i]}\left\{\frac{1}{l} \cdot \left(\sum_{j=1}^{l} 2 \cdot q_{ij}\right) - \frac{1}{l}\right\}, 0\right\} \\
&= \max\left\{y_{ij} - \max_{l \in [N_i]}\left\{\frac{1}{l} \cdot \left(\left(\sum_{j=1}^{l} z_{ij}\right) - 1\right)\right\}, 0\right\} \quad (z_{ij} \text{ from construction of } C_I) \\
&= \max\{y_{ij} - t_i, 0\} \quad (t_i \text{ from construction of } C_I),
\end{aligned}
$$

which is by definition equal to the output x'_{ij} of (6.3). □

The circuit C'_I we constructed that computes the function G_I uses gates of type in the set $\{G_\zeta, G_+, G_*, G_{\max}, G_{\min}, G_-^{[0,1]}, G_{*2}^{[0,1]}\}$, where $\zeta \in \mathbb{Q} \cap (0, 1]$.

6.5.3.1.3 The (n, n)-Consensus Halving instance

At this point we are ready to construct the (n, n)-Consensus Halving instance. The final circuit C'_I computes the function G_I, where $G_I : D_I \to D_I$, whose fixed points are precisely the Nash equilibria of the initial instance I of the d-player game, due to Lemma 20. The output of C'_I is the N-dimensional vector x' with entries x'_{ij} computed from (6.4). Let us close the circuit by connecting the output x'_{ij} with the input x_{ij} for every $i \in [d]$, $j \in S_i$. This new circuit, called C_I^o, is *cyclic*, meaning that it has no input and no output.

The cyclic circuit C_I^o (like C'_I) uses only gates in $\{G_\zeta, G_+, G_*, G_{\max}, G_{\min}, G_-^{[0,1]}, G_{*2}^{[0,1]}\}$, where $\zeta \in \mathbb{Q} \cap (0, 1]$. In Section 6.5.1 we describe how to turn such circuits into Consensus

HALVING instances. Suppose that C_I^o uses l gates. Then, by the procedure of Section 6.5.1 let us turn C_I^o into a special circuit $C_I^{o'}$ with $r = linear(l)$ gates which uses only the required gates by Lemma 19. Finally, still following that procedure, let us turn $C_I^{o'}$ into a CONSENSUS HALVING instance with $n := 4r$ agents.

We can now prove Theorem 24

Proof. In Section 6.5.2 it was proven that a solution to the above (n, n)-CONSENSUS HALVING instance, i.e. a solution with n cuts, in linear time can be translated back to a tuple $z^* := (z_1^*, z_2^*, \ldots, z_r^*)$ of satisfying values for the nodes of $C_I^{o'}$. Recall that $C_I^{o'}$ was created by another cyclic equivalent circuit C_I^o which was also created by merging the input and output nodes of an acyclic circuit C_I'.

Let us denote by $v_1, v_2, \ldots, v_N$ and $v_1', v_2', \ldots, v_N'$ the input and output nodes respectively of C_I' and denote by $V_1, V_2, \ldots, V_N$ the merged nodes in C_I^o and $C_I^{o'}$. Let us denote by $x^* := (x_1^*, x_2^*, \ldots, x_N^*)$ the N entries of z^* that correspond to the values of nodes $(V_1, V_2, \ldots, V_N)$. Since the procedure in Section 6.5.1 which turns C_I^o into $C_I^{o'}$ preserves the computation of the values of $V_1, V_2, \ldots, V_N$, it follows that x^* satisfies C_I^o. Consequently, if the values x^* are copied as values of both input $(v_1, v_2, \ldots, v_N)$ and output $(v_1', v_2', \ldots, v_N')$ nodes of C_I' then C_I' is satisfied, since these nodes of C_I' compute the same values as those that $V_1, V_2, \ldots, V_N$ compute in C_I^o.

As it was shown in Lemma 22, the output of C_I' computes the same output as C_I, which computes the function G_I. Thus, for x^* it holds that $G_I(x^*) = x^*$, i.e. it is a fixed point of G_I. Recall now that the fixed points of G_I are precisely the Nash equilibria of instance I of the initial "d-player Nash equilibrium" problem. Since, due to [66], "d-player Nash equilibrium" is FIXP-complete, it follows that (n, n)-CONSENSUS HALVING is FIXP-hard. □

Theorem 24, along with Theorem 21 give the following corollary.

Corollary 13. FIXP $\subseteq$ BU.

6.5.4 $(n, n-1)$-CONSENSUS HALVING is ETR-complete

We will show the ETR-hardness of $(n, n-1)$-CONSENSUS HALVING by reducing from the following problem FEASIBLE, which is known to be ETR-complete [125].

Definition 18 (FEASIBLE, FEASIBLE$_{[0,1]}$)**.** *Let $p(x_1, \ldots, x_m)$ be a polynomial.* FEASIBLE *asks whether there exists a point $(x_1, \ldots, x_m) \in \mathbb{R}^m$ that satisfies $p(x_1, \ldots, x_m) = 0$.* FEASIBLE$_{[0,1]}$ *asks whether there exists a point $(x_1, \ldots, x_m) \in [0,1]^m$ that satisfies $p(x_1, \ldots, x_m) = 0$.*

The idea is to turn the polynomial into a circuit, and then embed that circuit into a consensus halving instance using Lemma 19. As before, the main difficulty is ensuring that the preconditions of Lemma 19 are satisfied. To do this, we must ensure that the inputs to the circuit take values in $[0, 1]$, which is not the case if we reduce directly from FEASIBLE. Instead, we first consider the problem FEASIBLE$_{[0,1]}$, in which x is constrained to lie in $[0, 1]^n$ rather than $\mathbb{R}^n$, and we show the following result.

Lemma 23. FEASIBLE$_{[0,1]}$ *is* ETR*-complete.*

6.5.4.1 Proof of Lemma 23

Let us define the constrained version of ETR, denoted ETR$_{[0,1]}$, where the polynomials are over $[0, 1]^n$ (while in ETR they are over $\mathbb{R}^n$). It is easy to see that ETR$_{[0,1]} \subseteq$ ETR; an arbitrary ETR$_{[0,1]}$ instance $\exists(X_1, \ldots, X_m) \in [0, 1]^m \cdot \Phi$, where Φ is the ETR$_{[0,1]}$ formula, can be written as the following ETR instance $\exists(X_1, \ldots, X_m) \in \mathbb{R}^m \cdot \Phi \bigwedge_{i=1}^{m} ((X_i \geq 0) \wedge (X_i \leq 1))$.

We present a polynomial-time reduction from the ETR-complete problem FEASIBLE to an intermediate problem CONJUCTION$_{[0,1]}$, which belongs to ETR$_{[0,1]}$, thus showing that ETR$_{[0,1]}$ = ETR. Then we reduce a typical complete problem of ETR$_{[0,1]}$ to another intermediate problem called FEASIBLE$_{[0,1]}$, and finally we reduce the latter to $(n, n-1)$-CONSENSUS HALVING, thus showing that the latter is ETR-hard. A straightforward corollary is that CONJUCTION$_{[0,1]}$ and FEASIBLE$_{[0,1]}$ are ETR-complete, results that, together with the equivalence of classes ETR$_{[0,1]}$ = ETR, we believe are of independent interest.

6.5.4.1.1 ETR$_{[0,1]}$ = ETR

In this section we prove that ETR $\subseteq$ ETR$_{[0,1]}$, hence ETR$_{[0,1]}$ = ETR. To this end, we present a polynomial time reduction from FEASIBLE to CONJUCTION$_{[0,1]}$. Let us first define the problem CONJUCTION$_{[0,1]}$.

Definition 19 (CONJUCTION$_{[0,1]}$)**.** *Let $p_1, \ldots, p_k : [0, 1]^n \to \mathbb{R}$ be a family of polynomials, where each one of them is given as a sum of monomials with integer coefficients.* CONJUCTION$_{[0,1]}$ *asks whether the polynomials have a common zero.*

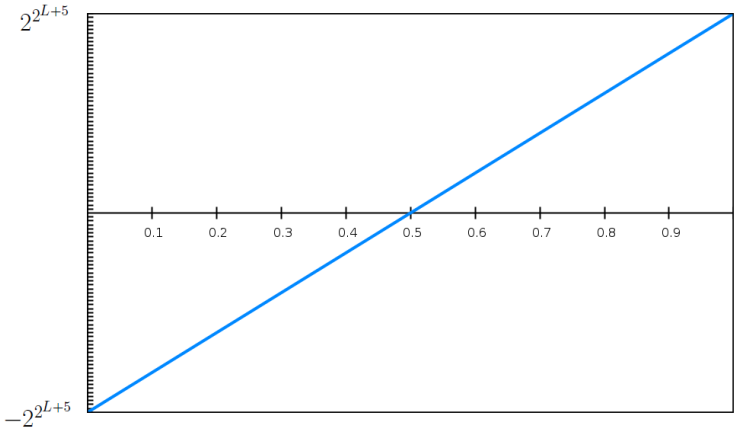


Figure 6.4: Function $x(y) = 2^{2^{L+5}} \cdot (2 \cdot y - 1)$.

Suppose we are asked to decide an arbitrary instance $(\exists X \in \mathbb{R}^n)\,(p(X) = 0)$ of Feasible. We wish to map every $X \in \mathbb{R}^n$ to a $Y \in [0,1]^n$ so that there is a solution $Y^* \in [0,1]^n$ if and only if there exists a solution $X^* \in \mathbb{R}^n$. To this end, we will need the following result by Schaefer and Štefanckovič [125]. A *semialgebraic set* is a subset of $\mathbb{R}^n$ described by a finite Boolean formula whose atoms are equalities and inequalities of multivariate polynomials over the reals. We borrow the terminology of [125] and by *(bit-)complexity* of a semialgebraic set we call the shortest length of any formula defining the set.

Proposition 2 ([125])**.** *If a bounded semialgebraic set in $\mathbb{R}^n$ has complexity at most $L \geq 5n$, then all its points have distance at most $2^{2^{L+5}}$ from the origin.*

The above proposition implies that if X_i is in a solution of Feasible then $|X_i|$ is upper bounded by $2^{2^{L+5}}$. Therefore, there is no need to map *all* real numbers to $[0,1]$, just the interval $\left[-2^{2^{L+5}}, 2^{2^{L+5}}\right]$. So, we will use the linear function

$$x(y) := 2^{2^{L+5}} \cdot (2 \cdot y - 1), \quad y \in [0,1] \tag{6.5}$$

in order to scale the solutions of $p(X)$ in $[0,1]^n$, (see Figure 6.4). First, we need to create the number $2^{2^{L+5}}$ or the number $2^{-2^{L+5}}$. These numbers have exponential in the input

bit-representation, that is why we will use the trick of "repeated squaring" to create the required number by introducing auxiliary variables which we repeatedly square $L+5$ times. Since we need our variables to be in $[0,1]$, we will create the number $2^{-2^{L+5}}$, and we do this by introducing the variables $S_1, \ldots, S_{L+6}$ and including the following conjunctions in our formula:

$$(2 \cdot S_1 = 1) \wedge \left(S_2 = (S_1)^2\right) \ldots \wedge \left(S_{L+6} = (S_{L+5})^2\right).$$

Now, using (6.5), we have

$$X_i \cdot S_{L+6} = 2 \cdot Y_i - 1,$$

where we have introduced a tuple of variables $Y := (Y_1, \ldots, Y_n) \in [0,1]^n$.

In order to make the substitution of variables, we multiply the given equation $p(X) = 0$ with $(S_{L+6})^d$, where $d := \sum_{i=1}^{n} d_i$ and d_i is the maximum degree of X_i in $p(X)$. Subsequently, in the new equation we substitute each product $S_{L+6} \cdot X_i$ with $2 \cdot Y_i - 1$, and we get a polynomial equation $q(Y) = 0$.

Eventually, the instance of the problem $\textsc{Conjuction}_{[0,1]}$ that we create is

$$\left(\exists Y, S \in [0,1]^{n+L+6}\right)(q(Y) = 0) \wedge (2 \cdot S_1 = 1) \bigwedge_{j=2}^{L+6} \left(S_j = (S_{j-1})^2\right),$$

where we denote by S the tuple $(S_1, \ldots, S_{L+6})$ and L is the total complexity of the initial Feasible instance. Note that the above instance has a solution if and only if the initial instance of Feasible has one. In fact, the stronger property holds that we can get any solution X from a solution Y in polynomial time, through (6.5), although this property is not necessary for our reduction since we are dealing with "yes/no problems". Also, note that the complexity of the new formula is at most $O(L^2) + O(L+6) = O(L^2)$.

Hence, we have proven that $\texttt{ETR} \subseteq \texttt{ETR}_{[0,1]}$. Since $\texttt{ETR}_{[0,1]} \subseteq \texttt{ETR}$ the following theorem follows:

Theorem 25. $\texttt{ETR}_{[0,1]} = \texttt{ETR}$.

6.5.4.1.2 FEASIBLE$_{[0,1]}$ is ETR-complete

We will show that FEASIBLE$_{[0,1]}$ is ETR$_{[0,1]}$-complete, so it is ETR-complete according to Theorem 25. Clearly, FEASIBLE$_{[0,1]}$ is in ETR$_{[0,1]}$, since it is a special case of a problem in ETR with constrained variables in $[0,1]^n$ where the boolean formula consists only of a single equation. We will now show that any problem in ETR$_{[0,1]}$ can be reduced to a FEASIBLE$_{[0,1]}$ instance.

Suppose we are asked to decide an arbitrary existential sentence of ETR$_{[0,1]}$:

$$(\exists(X_1,\ldots,X_n)\in[0,1]^n)\,\Phi \tag{6.6}$$

where Φ is a boolean formula with atoms $\Phi_i := f_i \diamond 0$ where each f_i, $i \in [m]$ is a polynomial function of $X_1,\ldots,X_n$ written in the standard form (a sum of monomials with integer coefficients) and $\diamond \in \{\leq, >\}$. This is without loss of generality, since we can turn every equality to a conjunction of two inequalities, and also, we can always move all monomials of an inequality to the left or right side appropriately.

The formula Φ consists of atoms Φ_i, $i \in [m]$ connected with $\wedge, \vee$ and $\neg$. Let us transform Φ in polynomial time into its equivalent one without $\neg$, by employing De Morgan's laws, and thus the negation of an "$\leq$-inequality" becomes an "$>$-inequality" and vice versa. As a first step, we would like to eliminate all the $\leq$ and $>$ symbols, so that our formula contains only atoms with $=$.

Consider an arbitrary atom $f_i \leq 0$. For brevity, in the following we will denote $(X_1,\ldots,X_n)$ by X and always imply that f_i depends on X. The sentence

$$(\exists X\in[0,1]^n)\,(f_i\leq 0)$$

is equivalent to the following,

$$\left(\exists X, R_i\in[0,1]^{n+1}\right)\left(f_i+\frac{R_i}{1-R_i}=0\right), \tag{6.7}$$

where an additional variable R_i is introduced. In an ETR formula division is not allowed, so in order to eliminate the division operation, we further transform (6.7) to the equivalent

$$\left(\exists X, R_i\in[0,1]^{n+1}\right)\left(f_i\cdot(1-R_i)+R_i=0\right), \tag{6.8}$$

where allowing $R_i = 1$ still does not allow $f_i > 0$, since (6.8) has no solution for $R_i = 1$.

Now, consider an arbitrary atom $f_i > 0$ of complexity L, i.e. the sentence

$$(\exists X \in [0,1]^n)\,(f_i > 0) \tag{6.9}$$

We will use the following result by Schaefer and Štefankovič. We remind that by *(bit-)complexity* of a semialgebraic set we call the shortest length of any formula defining the set.

Proposition 3 ([125])**.** *If two semialgebraic sets in $\mathbb{R}^n$ each of complexity at most $L \geq 5n$ have positive distance (for example, if they are disjoint and compact), then that distance is at least $2^{-2^{L+5}}$.*

According to the above proposition, sentence (6.9) is equivalent to the following,

$$(\exists X \in [0,1]^n)\left(f_i \geq 2^{-2^{L+5}}\right), \tag{6.10}$$

where, since the bit-length of $2^{-2^{L+5}}$ is exponential, we can create it using the "repeated squaring" trick. That is we introduce $L+6$ more variables $S_1, \ldots, S_{L+6} \in [0,1]$ whose tuple we denote by S and add the following conjunction of atoms in the formula:

$$\begin{aligned} &(2 \cdot S_1 = 1) \\ \wedge &\left(S_2 = (S_1)^2\right) \\ &\vdots \\ \wedge &\left(S_{L+6} = (S_{L+5})^2\right). \end{aligned}$$

Then, (6.10) is equivalent to

$$\left(\exists X, S \in [0,1]^{n+L+6}\right)(S_{L+6} - f_i \leq 0), \tag{6.11}$$

which we know how to transform to an equality (see (6.8)). Therefore, by introducing a variable T_i, (6.11) is equivalent to,

$$\left(\exists X, S, T_i \in [0,1]^{n+L+7}\right)((S_{L+6} - f_i) \cdot (1 - T_i) + T_i = 0), \tag{6.12}$$

Now our boolean formula consists of m atoms that are polynomials equal to 0. We

will proceed using the arsenal introduced in [99] where they prove a similar result for the unconstrained ETR case. First, let us introduce an additional "boolean" variable W_i, $i \in [m]$, one for each atom, with value 1 if the atom initially had $\leq$, and 0 if the atom initially had $>$. That is, for an arbitrary atom i, one of (6.8) or (6.12) is true. So, we can add in our formula the following sub-formula and the conjunction of them for every $i \in [m]$:

$$\begin{aligned} &((f_i \cdot (1 - R_i) + R_i = 0) \wedge (1 - W_i = 0)) \\ \vee &(((S_{L+6} - f_i) \cdot (1 - T_i) + T_i = 0) \wedge (W_i = 0)) \,. \end{aligned} \tag{6.13}$$

Next, we will eliminate the $\vee$ and $\wedge$ operators using the following trick: $(p = 0) \vee (q = 0)$ is equivalent to $p \cdot q = 0$ and $(p = 0) \wedge (q = 0)$ is equivalent to $p^2 + q^2 = 0$. We will start from the latter conjunction of sub-formulas. For every sub-formula $i \in [m]$, as in (6.13), we have a single polynomial $h_i = 0$, where

$$h_i := \Big((f_i \cdot (1 - R_i) + R_i)^2 + (1 - W_i)\Big) \cdot \Big(((S_{L+6} - f_i) \cdot (1 - T_i) + T_i)^2 + W_i\Big) \,,$$

thus replacing the conjunction of sub-formulas as in (6.13) with $\bigwedge_{i=1}^{m}(h_i = 0)$. Note that we have not squared $(1 - W_i)$ and W_i because we know they are in $[0, 1]$. Now, let us substitute the initial formula Φ (after removing the $\neg$ operators) with its equivalent, using W_i's. That is, if the i-th atom of the initial formula is a "$\leq$-inequality" we substitute it with the atom $(1 - W_i = 0)$, and if it is a "$>$-inequality" we substitute it with the atom $(W_i = 0)$. Therefore, we can now apply the aforementioned trick of multiplication to eliminate the $\vee$ operators and thus have a formula with just $\wedge$, i.e.

$$\bigwedge_{i=1}^{m'}(g_i = 0),$$

where $m' \leq m$ is the number of atoms in the resulting formula that represents Φ.

The whole formula, that is, together with the sub-formulas for the "boolean" variables is

$$\bigwedge_{i=1}^{m'}(g_i = 0) \bigwedge_{i=1}^{m}(h_i = 0).$$

What is left is to transform this into a single polynomial using the trick of sum of squares

(squares are not needed for g_i's because our polynomials are in [0,1]),

$$\sum_{i=1}^{m'} g_i + \sum_{i=1}^{m} h_i^2 = 0.$$

Let us denote by W the tuple of "boolean variables" $(W_1, \ldots, W_m) \in [0,1]^m$, and similarly, $R := (R_1, \ldots, R_m) \in [0,1]^m$ and $T := (T_1, \ldots, T_m) \in [0,1]^m$. Then the existential sentence that we have to decide is

$$\left(\exists X, R, S, T \in [0,1]^{n+2m+L+6}\right) \left(\sum_{i=1}^{m'} g_i + \sum_{i=1}^{m} h_i^2 = 0 \right), \tag{6.14}$$

where L is the maximum complexity of an ">-inequality" in Φ. Since the number of atoms m of Φ is at most twice the total complexity L' of Φ (because we substituted each equation with two inequalities), the number of variables is $O(L')$. Also, since $m' \leq m$ and the complexity of h_i is at most $8 \cdot 16 = 128$ times the complexity of the i-th atom in Φ, the complexity of the resulting formula of Feasible$_{[0,1]}$ is $O(L')$.

We have proven that the formulas (6.6) and (6.14) are equivalent. Therefore, one is true if and only if the other is, hence Feasible$_{[0,1]}$ is ETR$_{[0,1]}$-complete. Finally, from Theorem 25 we get that Feasible$_{[0,1]}$ is ETR-complete.

Theorem 26. $(n, n-1)$-Consensus Halving *is* ETR*-complete.*

6.5.4.2 Proof of Theorem 26

As we show in Theorem 22, (n,k)-Consensus Halving is in ETR. In this section we prove that $(n, n-1)$-Consensus Halving is ETR-hard, implying that it is complete for ETR. This extends the results of [70], where it was established that $(n, n-1)$-Consensus Halving is NP-hard even when a solution is required to be $1/poly(n)$-approximately correct, i.e. it allows that $|F_i(A_+) - F_i(A_-)| \leq \epsilon$ for every agent i, where $\epsilon = 1/poly(n)$.

We present a polynomial time reduction from the ETR-complete problem Feasible$_{[0,1]}$ to $(n, n-1)$-Consensus Halving. Suppose we are asked to decide an arbitrary instance of Feasible$_{[0,1]}$, i.e. the existential sentence

$$\left(\exists X \in [0,1]^N\right) (p(X) = 0), \tag{6.15}$$

where $X := (X_1, \ldots, X_N) \in [0,1]^N$ and p, is a polynomial function of $X_1, \ldots, X_N$ written

in the standard form (a sum of monomials with integer coefficients). Consider the positive integer coefficients $C_1, \ldots, C_l$ of p, where the number of terms of the polynomial is l. These coefficients are positive without loss of generality, since we can replace a negative coefficient C that follows after a $+(-)$ in the polynomial, with $-C$ that follows a $-(+)$. Also, let us normalize the coefficients and create new ones $c_1, \ldots, c_l$, where

$$c_j := \frac{C_j}{l \cdot C_{max}}, \quad j \in [l],$$

where $C_{max} := \max_j C_j$. Note that our new polynomial $q(X)$ which uses the new coefficients has exactly the same roots as $p(X)$. Also, note that $c_j \in (0, \frac{1}{l}]$ for every $j \in [l]$, a fact that will play an important role at the last steps of our reduction.

Now, let us split polynomial q into two polynomials q_1 and q_2, such that

$$q(X) := q_1(X) - q_2(X),$$

and both q_1 and q_2 are sums of *positive* terms; l_1 and l_2 terms of q_1 and q_2 respectively, where $l = l_1 + l_2$. In particular,

$$q_1(X) := \sum_{j=1}^{l_1} r_j(X),$$
$$q_2(X) := \sum_{j=l_1+1}^{l_2} r_j(X),$$

where $r_j(X) := c_j \cdot X_1^{d_{1j}} \cdot \cdots \cdot X_N^{d_{Nj}}$ is the term $j \in [l]$ and d_{ij} is the exponent of variable X_i, $i \in [N]$, in the j-th term. Eventually, the existential sentence, equivalent to (6.15), that we ask to decide is

$$\left(\exists X \in [0,1]^N\right) \left(q_1(X) = q_2(X)\right).$$

Let us construct the algebraic circuit that takes as input the tuple X and computes the value of $q_1(X)$. This circuit needs only to use gates in $\{G_\zeta, G_+, G_{*\zeta}, G_*, G_{()^2}\}$, where $\zeta \in \mathbb{Q} \cap (0, 1]$. To see why, observe that since every $X_i \in [0, 1]$, $i \in [N]$, any multiplication between them by a G_* gate is done properly (the gate's inputs/output are in $[0, 1]$), and obviously the same holds for $G_{()^2}$. Also, note that due to our downscaled coefficients c_j, it

is $c_j \leq 1/2$ for every j, and also

$$\sum_{j=1}^{l_1} r_j(X) \leq l_1/l \leq 1. \tag{6.16}$$

Therefore, we guarantee that any of the $l_1 - 1$ additions of the terms r_j of q_1 by a G_+ gate is done properly, (inputs in $[0, 1/2]$ and output in $[0, 1]$). Similarly, we construct a circuit that computes q_2.

Example: Consider the following instance of FEASIBLE$_{[0,1]}$:

$$\big(\exists X := (X_1, X_2, X_3) \in [0,1]^3\big)(p(X) = 0),$$
$$\text{where} \quad p(X) := 6X_1^3X_2 - 4X_2^2X_3^2 - X_1X_3^4 + 8X_2X_3^2 + 3.$$

Let us create an equivalent existential sentence by replacing $p(X)$ with the polynomial $q(X)$, where $q(X) := \frac{p(X)}{40}$, so

$$\big(\exists X := (X_1, X_2, X_3) \in [0,1]^3\big)(q(X) = 0),$$
$$\text{where} \quad q(X) := \frac{6}{40}X_1^3X_2 - \frac{4}{40}X_2^2X_3^2 - \frac{1}{40}X_1X_3^4 + \frac{8}{40}X_2X_3^2 + \frac{3}{40}.$$

We proceed by splitting $q(X)$ into the following polynomials,

$$q_1(X) := \frac{6}{40}X_1^3X_2 + \frac{8}{40}X_2X_3^2 + \frac{3}{40}$$
$$q_2(X) := \frac{4}{40}X_2^2X_3^2 + \frac{1}{40}X_1X_3^4.$$

The circuit that computes these polynomials is presented in Figure 6.5.

At this point we are ready to prove Theorem 26.

Proof. Let us construct a $(n, n-1)$-CONSENSUS HALVING instance, where n is to be defined later. In Section 6.5.1 we have shown how to construct an equivalent circuit to the one that computes q_1, q_2, called "special circuit", that

- uses only gates $G_\zeta, G_+, G_{*\zeta}, G_{()^2}, G_-^{[0,1]}, G_{*2}^{[0,1]}$,

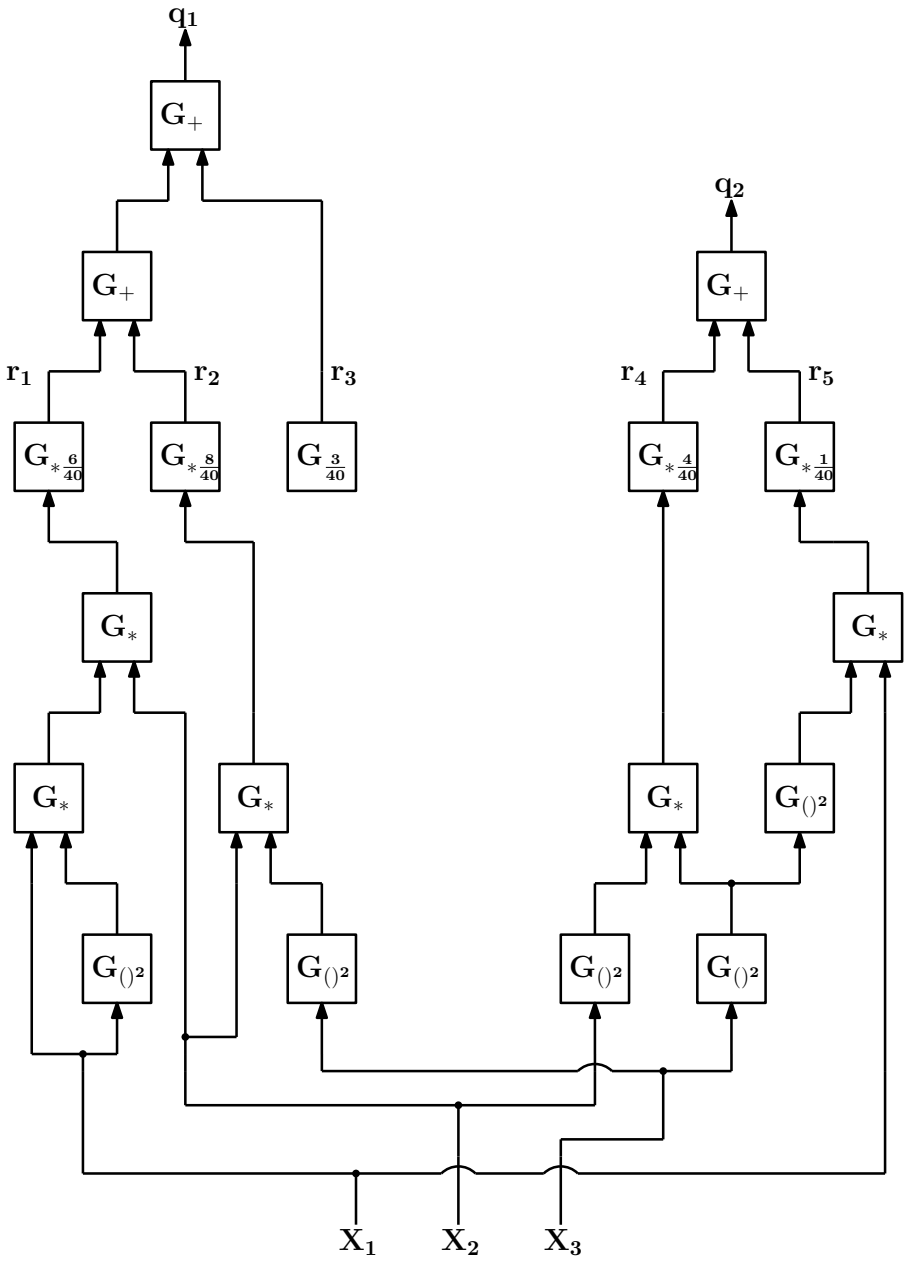


Figure 6.5: An example of the circuit that computes q_1 and q_2. Each G_* gate that multiplies two variable inputs is replaced by the structure shown in Figure 6.6.

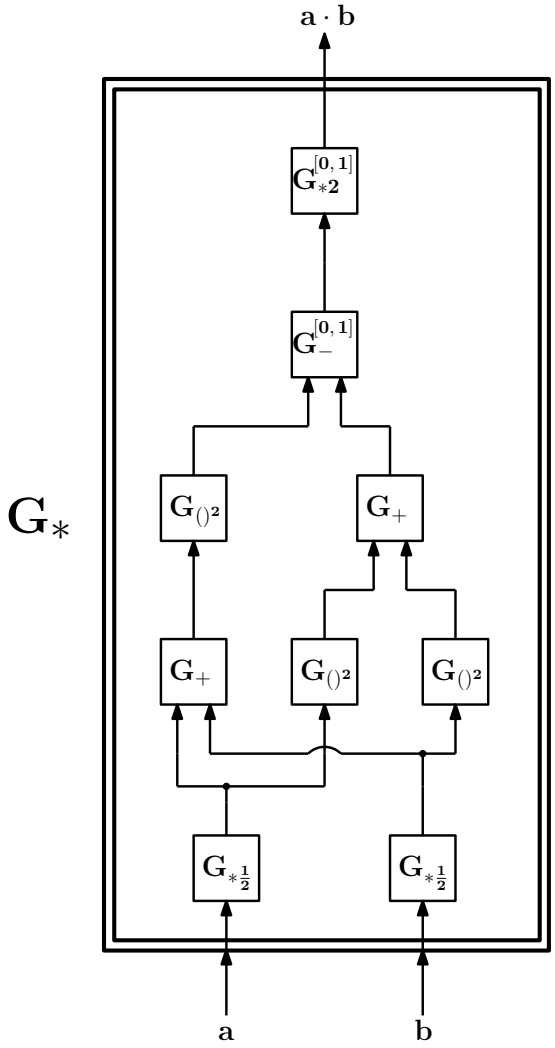


Figure 6.6: The internal components of G_* (see Section 6.5.1 for the mathematical formula that prescribes this transformation).

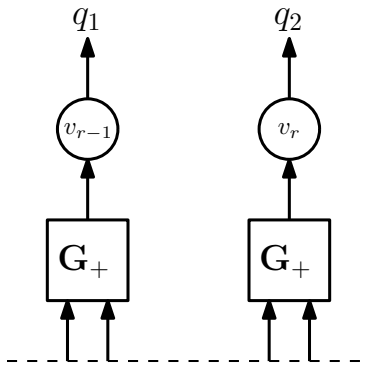


Figure 6.7: The last two nodes of the special circuit.

- every G_ζ and $G_{*\zeta}$ has $\zeta \in \mathbb{Q} \cap (0, 1]$,
- for every input $x \in [0, 1]^N$, all intermediate values computed by the circuit lie in $[0, 1]$.

For the constraints of the above types of gates, see Tables 6.1, 6.2.

Let the number of gates in that special circuit be $r := poly(N)$. Consider the last two nodes of the special circuit whose outgoing edges are q_1 and q_2 respectively. Without loss of generality, we name them v_{r-1} and v_r (see Figure 6.7).

By Lemma 19 and the construction described in its proof (Section 6.5.2), we embed the special circuit in a Consensus Halving instance. This instance now consists of $4r$ agents, since to each node $i \in [r]$ correspond 4 agents: ad_i, mid_i, cen_i and ex_i with valuation functions described by (6.1).

According to the embedding described in Section 6.5.2, a tuple $(z_1^*, \ldots, z_r^*)$ of values that satisfies the special circuit, corresponds to a $(4r, 4r)$-Consensus Halving solution, i.e. a tuple $(t_1^*, \ldots, t_{4r}^*)$ with $0 \leq t_1^* \leq \cdots \leq t_{4r}^* \leq 12r$, of the Consensus Halving instance we constructed, and vice versa. As shown in detail in Section 6.5.2, a circuit every value z_i^* in a solution can be translated to 4 cuts $t_{4i-3}^*, t_{4i-2}^*, t_{4i-1}^*, t_{4i}^*$ in the Consensus Halving solution by the transformation in paragraph Section 6.5.2.2. Conversely, a 4-tuple $(t_{4i-3}^*, t_{4i-2}^*, t_{4i-1}^*, t_{4i}^*)$ of cuts in a Consensus Halving solution can be translated to a single value z_i^* by the simple transformation $z_i^* = t_{4i}^* - v_{i,l}^+$ in Section 6.5.2.2.

Let us now introduce a $(4r + 1)$-st additional agent, named *finis* (from the Latin word for "end") who does not correspond to any node. The valuation function of this agent is non-zero only in the intervals v_{r-1}^+ and v_r^- and, in particular is the following,

$$finis(t) = \begin{cases} 1, & t \in v_{r-1}^+ \cup v_r^- \\ 0, & \text{otherwise.} \end{cases} \tag{6.17}$$

Eventually, the number of agents in the embedding is $n := 4r + 1$.

We will show that the answer to the arbitrary $\text{Feasible}_{[0,1]}$ instance (6.15) is "yes", if and only if the answer to the $(n, n-1) -$ Consensus Halving problem is "yes", i.e. there exists a $(n-1)$-cut that satisfies n agents.

Suppose that there exists a solution $X^* := (X_1^*, \ldots, X_N^*) \in [0, 1]^N$ of (6.15), which equivalently means that $q_1(X^*) = q_2(X^*)$. Then, by the correct construction of our special circuit (following the procedure in Section 6.5.1) which uses r gates and computes q_1 and q_2, there is a tuple $z^* := (z_1^*, \ldots, z_r^*)$ that satisfies it. Let, without loss of generality,

$(z_1^*, \ldots, z_N^*) := (X_1^*, \ldots, X_N^*)$. Then it holds that $q_1(z_1^*, \ldots, z_N^*) = q_2(z_1^*, \ldots, z_N^*)$, therefore $z_{r-1}^* = z_r^*$.

According to the aforementioned translation to cuts, in the Consensus Halving instance there will be a cut $t_{4(r-1)}^* = v_{r-1,l}^+ + z_{r-1}^*$ in interval v_{r-1}^+ (i.e. a positive cut), and another one in $t_{4r-1}^* = v_{r,l}^- + z_r^*$ in interval v_r^- (i.e. a negative cut). From the valuation function (6.17) of agent $finis$, we can see that her positive total valuation equals her negative total valuation, since $z_{r-1}^* \cdot 1 + (1 - z_r^*) \cdot 1 = (1 - z_{r-1}^*) \cdot 1 + z_r^* \cdot 1$ holds from $z_{r-1}^* = z_r^*$. Therefore $finis$ is satisfied. Also, the agents ad_i, mid_i, cen_i, ex_i for all $i \in [r]$ are satisfied as argued in Section 6.5.2, and the answer to $(n, n-1)$ − Consensus Halving is "yes", since we have $4r + 1$ agents satisfied by $4r$ cuts.

Suppose now that there exists a $4r$-cut $(t_1^*, \ldots, t_{4r}^*)$ with $0 \leq t_1^* \leq \cdots \leq t_{4r}^* \leq 12r$ that is a solution of the $(n, n-1)$-Consensus Halving instance we constructed, where $n := 4r + 1$. As argued in Section 6.5.2, if the ad_i, mid_i, cen_i, ex_i agents for $i \in [r]$ are satisfied then each of cen_i, ex_i agents imposes a cut in interval v_i^- and v_i^+ respectively. The cuts in intervals v_i^+, for all $i \in [r]$ can be translated back to values z_i^*, which successfully compute the values of the circuit, i.e. they satisfy the circuit. There are also two interesting cuts $t_{4(r-1)}^*$ and t_{4r-1}^* imposed by ex_{r-1} and cen_r respectively which satisfy agent $finis$. Since this agent is satisfied with no additional cut, it holds that $z_{r-1}^* \cdot 1 + (1 - z_r^*) \cdot 1 = (1 - z_{r-1}^*) \cdot 1 + z_r^* \cdot 1$, or equivalently $z_{r-1}^* = z_r^*$. Since z_{r-1}^* and z_r^* correspond to the value of the circuit at q_1 and q_2 respectively, for the circuit's inputs $(z_1^*, \ldots, z_N^*)$ it holds that $q_1(z_1^*, \ldots, z_N^*) = q_2(z_1^*, \ldots, z_N^*)$. Equivalently, $q(z_1^*, \ldots, z_N^*) = 0$, and equivalently $p(z_1^*, \ldots, z_N^*) = 0$. Therefore, we have found values that satisfy (6.15), and the answer to Feasible$_{[0,1]}$ is "yes". □

6.6 A Discussion on ETR and Other Complexity Classes

As shown in Sections 6.4.2 and 6.5.4, the decision problem $(n, n-1)$-Consensus Halving that asks to decide whether a $(n-1)$-cut solution exists for n agents is ETR-complete. This result is analogous to the result in [70], where it is shown that in its approximate version, $(n, n-1, \epsilon)$-Consensus Halving is NP-complete, for inverse-polynomial ϵ. Furthermore, the exact problem (n, n)-Consensus Halving we considered here was proved to be in BU and FIXP-hard (see Sections 6.4.1 and 6.5.3), while its approximate counterpart (n, n, ϵ)-Consensus Halving was shown to be PPA-complete in [71]. Moreover, our result that LinearBU =PPA gives an indication that (n, n)-Consensus Halving is BU-complete. The

aforementioned results, if indeed the latter is also a truth, confirm a remarkable correspondence between the approximate and exact computation worlds that has been suspected due to many previous results. An example of such results is the ones on the problem of finding a Nash equilibrium in a strategic game: for the m-player case, with $m \geq 3$, the exact version was proven to be FIXP-complete [66], while the approximate version is PPAD-complete [50]. It has also been shown in [66] that LinearFIXP =PPAD. Finally, in a slightly different correspondence than the one mentioned for Consensus Halving, various *decision versions* of the m-player Nash equilibrium problem with $m \geq 3$, are ETR-complete [27,75,80], while for $m = 2$ they are NP-complete [24,25,76].

It seems that ETR is a class that captures decision problems that are a lot harder than these in NP (under standard complexity assumptions) because either they do not have truth certificates of polynomial length or because the certificate cannot be checked in polynomial time. However, it seems that ETR is the analogue of NP in the Blum-Shub-Smale model of computation [28], in which computing functions over real numbers is as costly as is computing functions over rational numbers in Turing machines. In this thesis, we also make an attempt to define the analogues of FNP and TFNP, namely FETR and TFETR (see Section 2.4). According to previous results and the ones shown in this thesis, it holds that PPAD $\subseteq$ PPA $\subseteq$ TFNP $\subseteq$ FNP and FIXP $\subseteq$ BU $\subseteq$ TFETR $\subseteq$ FETR.

In the next chapter we provide a general framework for approximation schemes, a framework designed for problems in a subclass of ETR. In particular, since some optimization problems in TFNP or, in general, FNP (whose corresponding decision problems are in NP), have polynomial or quasi-polynomial time approximation schemes (PTAS/QPTAS), we study harder problems in TFETR or FETR, and seek similar approximation schemes. In a beautiful turn of events, in Chapter 7, we show that PTASs and QPTASs exist for a wide class of problems in ETR (or more precisely, FETR). By extending a well-known technique that yields the best possible algorithm (under standard complexity assumptions) for computing approximate Nash equilibria in strategic games, we provide a general framework that gives in a standardized way, approximation algorithms of the same quality as the state of the art for some problems, while for some other problems these algorithms are the first to achieve an efficient approximation. Interestingly, approximation techniques that work inside FNP, transcend it, and reach FETR.

Part IV

Approximation Algorithms

Chapter 7

Approximating the Existential Theory of the Reals

The Existential Theory of the Reals (ETR) consists of existentially quantified Boolean formulas over equalities and inequalities of polynomial functions of variables in $\mathbb{R}$. In this chapter we propose and study the approximate existential theory of the reals (ϵ-ETR), in which the constraints only need to be satisfied approximately. We first show that when the domain of the variables is $\mathbb{R}$ then ϵ-ETR = ETR under polynomial-time reductions, and then study the constrained ϵ-ETR problem when the variables are constrained to lie in a given bounded convex set. Our main theorem is a sampling theorem, similar to those that have been proved for approximate equilibria in normal form games. It discretizes the domain in a grid-like manner whose density depends on various properties of the formula. A consequence of our theorem is that we obtain a quasi-polynomial time approximation scheme (QPTAS) for a fragment of constrained ϵ-ETR. We use our theorem to create several new PTAS and QPTAS algorithms for problems from a variety of fields.

The results of this chapter have been published in the Proceedings of the 14th International Conference on Web and Internet Economics (WINE 2018) [57] (co-authored with Deligkas, Fearnley and Spirakis).

7.1 Overview

7.1.1 Sampling techniques

The Lipton-Markakis-Mehta algorithm (LMM) is a well known method for computing approximate Nash equilibria in normal form games [96]. The key idea behind their technique is to prove that there exist approximate Nash equilibria where all players use *simple* strategies.

Suppose that we have a convex set $C = \text{conv}(c_1, c_2, \ldots, c_l)$ defined by vectors c_1 through c_l. A vector $x \in C$ is *k-uniform* if it can be written as a sum of the form $(\beta_1/k) \cdot c_1 + (\beta_2/k) \cdot c_2 + \cdots + (\beta_l/k) \cdot c_l$, where each β_i is a non-negative integer and $\sum_{i=1}^{l} \beta_i = k$.

Since there are at most $l^{O(k)}$ k-uniform vectors, we can enumerate all k-uniform vectors in $l^{O(k)}$ time. For approximate equilibria in $n \times n$ bimatrix games, Lipton, Markakis, and Mehta showed that for every $\epsilon > 0$ there exists an ϵ-Nash equilibrium where both players use k-uniform strategies where $k \in O(\log n/\epsilon^2)$, and so they obtained a quasi-polynomial time approximation scheme (QPTAS) for finding an ϵ-Nash equilibrium.

Their proof of this fact uses a sampling argument. Every bimatrix game has an exact Nash equilibrium (NE), and each player's strategy in this NE is a probability distribution. If we sample from each of these distributions k times, and then construct new k-uniform strategies using these samples, then when $k \in O(\log n/\epsilon^2)$ there is a positive probability the new strategies form an ϵ-NE. So by the probabilistic method, there must exist a k-uniform ϵ-NE.

The sampling technique has been widely applied. It was initially used by Althöfer [8] in zero-sum games, before being applied to non-zero sum games by Lipton, Markakis, and Mehta [96]. Subsequently, it was used to produce algorithms for finding approximate equilibria in normal form games with many players [19], sparse bimatrix games [20], tree polymatrix [21], and Lipschitz games [62]. It has also been used to find constrained approximate equilibria in polymatrix games with bounded treewidth [60].

At their core, each of these results uses the sampling technique in the same way as the LMM algorithm: first take an exact solution to the problem, then sample from this solution k times, and finally prove that with positive probability the sampled vector is an approximate solution to the problem. The details of the proofs, and the value of k, are often tailored to the specific application, but the underlying technique is the same.

7.1.2 The existential theory of the reals

In this chapter we ask the following question: *is there a broader class of problems to which the sampling technique can be applied?* We answer this by providing a sampling theorem for the existential theory of the reals. The existential theory of the reals consists of existentially quantified formulae using the connectives $\{\wedge, \vee, \neg\}$ over polynomials compared with the operators $\{<, \leq, =, \geq, >\}$. For example, each of the following is a formula in the existential theory of the reals.

$$\exists x \exists y \exists z \cdot (x = y) \wedge (x > z) \qquad \exists x \cdot (x^2 = 2)$$
$$\exists x \exists y \cdot \neg(x^{10} = y^{100}) \vee (y \geq 4) \qquad \exists x \exists y \exists z \cdot (x^2 + y^2 = z^2)$$

Given a formula in the existential theory of the reals, we must decide whether the formula is *true*, that is, whether there do indeed exist values for the variables that satisfy the formula. Throughout this chapter we will use the Turing model of computation (also known as bit model). In this model, the inputs of our problems will be polynomial functions represented by tensors with rational entries which are encoded as a string of binary bits.

`ETR` is defined as the class that contains every problem that can be reduced in polynomial time to the typical `ETR` problem: Given a Boolean formula F, decide whether F is a true sentence in the existential theory of the reals. It is known that in the Turing model `ETR` $\subseteq$ `PSPACE` [35], and `NP` $\subseteq$ `ETR` since the problem can easily encode Boolean satisfiability. However, the class is not known to be equal to either `PSPACE` or `NP`, and it seems to be a distinct class of problems between the two. Many problems are now known to be `ETR`-complete, including various problems involving constrained equilibria in normal form games with at least three players [24–27,75].

7.1.3 Our contribution

In this chapter we propose the *approximate* existential theory of the reals (ϵ-ETR), where we seek a solution that approximately satisfies the constraints of the formula. We show a subsampling theorem for a large fragment of ϵ-ETR, which can be used to obtain PTASs and QPTASs for the problems that lie within it. We believe that this will be useful for future research: instead of laboriously reproving subsampling results for specific games, it now suffices to simply write a formula in ϵ-ETR and then apply our theorem to immediately get the desired result. To exemplify this, we prove several new QPTAS and PTAS results

using our theorem.

Our first result is actually that, in the computational complexity world, ϵ-ETR = ETR, meaning that the problem of computing an approximate solution to an ETR formula is as hard as finding an exact solution. However, this result crucially relies on the fact that ETR formulas can have solutions that are doubly-exponentially large. This motivates the study of *constrained* ϵ-ETR, where the solutions are required to lie within a given bounded convex set.

Our main theorem (Theorem 29) gives a subsampling result for constrained ϵ-ETR. It states that if the formula has an exact solution, then it also has a k-uniform approximate solution, where the value of k depends on various parameters of the formula, such as the number of constraints and the number of variables. The theorem allows for the formula to be written using *tensor* constraints, which are a type of constraint that is useful in formulating game-theoretic problems.

The consequence of the main theorem is that, when various parameters of the formula are up to polylogarithmic in specific parameters (see Corollary 14), we are able to obtain a QPTAS for approximating the existential theory of the reals. Specifically, this algorithm either finds an approximate solution of the constraints, or verifies that no exact solution exists. In many game theoretic and fair division applications an exact solution always exists, and so this algorithm will always find an approximate solution.

We should mention here also that our technique allows approximation of optimization problems whose objective function does not need to be described using the grammar of ETR formulas. For a discussion on this, see Remark 1. Also, we are not just applying the well-known subsampling techniques in order to derive our main theorem. Our main theorem incorporates a new method for dealing with polynomials of degree d, which prior subsampling techniques were not able to deal with.

Our theorem can be applied to a wide variety of problems. In the game theoretic setting, we prove new results for constrained approximate equilibria in normal form games, and approximating the value vector of a Shapley game. Then we move to the fair division setting, and we show how a special case of the Consensus Halving problem admits a QPTAS. We also show optimization results. Specifically, we give approximation algorithms for optimizing polynomial functions over a bounded convex set, subject to polynomial constraints. We also give algorithms for approximating eigenvalues and eigenvectors of tensors. Finally, we apply the theorem to some problems from computational geometry.

7.2 The Existential Theory of the Reals

Let x_1, x_2, ..., $x_q \in \mathbb{R}$ be distinct variables, which we will treat as a vector $x \in \mathbb{R}^q$. A *term* of a multivariate polynomial is a function $T(x) := a \cdot x_1^{d_1} \cdot x_2^{d_2} \cdot \cdots \cdot x_q^{d_q}$, where a is non negative rational and $d_1, d_2, \ldots, d_q$ are non negative integers. A multivariate polynomial is a function $p(x) := T_1(x) + T_2(x) + \cdots + T_t(x) + c$, where each T_i is a term as defined above, and $c \in \mathbb{Q}_{\geq 0}$ is a constant.

We now define *Boolean formulae* over multivariate polynomials. The atoms of the formula are polynomials compared with $\{<, \leq, =, \geq, >\}$, and the formula itself can use the connectives $\{\wedge, \vee, \neg\}$.

Definition 20. *The* existential theory of the reals *consists of every true sentence of the form* $\exists x_1 \exists x_2 \ldots \exists x_q \cdot F(x)$*, where* F *is a Boolean formula over multivariate polynomials of* x_1 *through* x_q*.*

ETR is defined as the class that contains every problem that can be reduced in polynomial time to the typical ETR problem: Given a Boolean formula F, decide whether F is a true sentence in the existential theory of the reals. We will say that F has m constraints if it uses m operators from the set $\{<, \leq, =, \geq, >\}$ in its definition.

7.2.1 The approximate ETR

In the *approximate* existential theory of the reals, we replace the operators $\{<, \leq, \geq, >\}$ with their approximate counterparts. We define the operators $<_\epsilon$ and $>_\epsilon$ with the interpretation that $x <_\epsilon y$ holds if and only if $x < y + \epsilon$ and $x >_\epsilon y$ if and only if $x > y - \epsilon$ for some given $\epsilon > 0$. The operators $\leq_\epsilon$ and $\geq_\epsilon$ are defined analogously.

We do not allow equality tests in the approximate ETR. Instead, we require that every constraint of the form $x = y$ should be translated to $(x \leq y) \wedge (y \leq x)$ before being weakened to $(x \leq_\epsilon y) \wedge (y \leq_\epsilon x)$.

We also do not allow negation in Boolean formulas. Instead, we require that all negations are first pushed to atoms, using De Morgan's laws, and then further pushed into the atoms by changing the inequalities. So the formula $\neg((x \leq y) \wedge (a > b))$ would first be translated to $(x > y) \vee (a \leq b)$ before then being weakened to $(x >_\epsilon y) \vee (a \leq_\epsilon y)$.

Definition 21. *The approximate existential theory of the reals consists of every true sentence of the form* $\exists x_1 \exists x_2 \ldots \exists x_q \cdot F(x)$*, where* F *is a negation-free Boolean formula using the operators* $\{<_\epsilon, \leq_\epsilon, \geq_\epsilon, >_\epsilon\}$ *over multivariate polynomials of* x_1 *through* x_q*.*

Given a Boolean formula F, the ϵ-ETR problem asks us to decide whether F is a true sentence in the approximate existential theory of the reals, where the operators $\{<_\epsilon, \leq_\epsilon, \geq_\epsilon, >_\epsilon\}$ are used.

7.2.1.1 Unconstrained ϵ-ETR

Our first result is that if no constraints are placed on the value of the variables, that is if each x_i can be arbitrarily large, then ϵ-ETR = ETR for *all* values of $\epsilon > 0$. We show this via a two way polynomial-time reduction between ϵ-ETR and ETR. The reduction from ϵ-ETR to ETR is trivial, since we can just rewrite each constraint $x <_\epsilon y$ as $x < y + \epsilon$, and likewise for the other operators.

For the other direction, we show that the ETR-complete problem Feasible, which asks us to decide whether a system of multivariate polynomials $(p_i)_{i=1,\ldots,k}$ has a shared root, can be formulated in ϵ-ETR. We will prove this by modifying a technique of Schaefer and Stefankovic [125].

Definition 22 (Feasible)**.** *Given a system of k multi-variate polynomials $p_i : \mathbb{R}^n \to \mathbb{R}^n$, $i = 1, \ldots, k$, decide whether there exists an $x \in \mathbb{R}^n$ such that $p_i(x) = 0$ for all i.*

Schaefer and Stefankovic showed that this problem is ETR-complete.

Theorem 27 ([125])**.** Feasible *is* ETR*-complete.*

We will reduce Feasible to ϵ-ETR. Let $P = (p_i)_{i=1,\ldots,k}$ be an instance of Feasible, and let L be the number of bits needed to represent this instance. We define $\text{gap}(P) = 2^{-2^{L+5}}$. The following lemma was shown by Schaefer and Stefankovic.

Lemma 24 ([125])**.** *Let $P = (p_i)_{i=1,\ldots,k}$ be an instance of* Feasible*. If there does not exist an $x \in \mathbb{R}^n$ such that $p_i(x) = 0$ for all i, then for every $x \in \mathbb{R}^n$ there exists an i such that $|p_i(x)| > \text{gap}(P)$.*

In other words, if the instance of Feasible is not solvable, then one of the polynomials will always be bounded away from 0 by at least $\text{gap}(P)$.

The reduction The first task is to build an ϵ-ETR formula that ensures that a variable $t \in \mathbb{R}$ satisfies $t \geq \epsilon / \text{gap}(P)$. This can be done by the standard trick of repeated squaring, but we must ensure that the ϵ-inequalities do not interfere with the process. We define

the following formula over the variables $t, g_1, g_2, \ldots, g_{L+6} \in \mathbb{R}^n$, where all of the following constraints are required to hold.

$$\begin{aligned} g_1 &\geq_\epsilon 2 + \epsilon, \\ g_j &\geq_\epsilon g_{j-1}^2 + \epsilon && \text{for all } j \in \{2, 3, \ldots, L+6\}. \\ t &\geq_\epsilon \epsilon \cdot g_{L+6} + \epsilon \end{aligned}$$

In other words, this requires that $g_1 \geq 2$, and $g_j \geq g_{j-1}^2$. So we have $g_{L+6} \geq 2^{2^{L+5}}$, and hence $t \geq \epsilon / \operatorname{gap}(P)$. Note that the size of this formula is polynomial in the size of P.

Given an instance $P = (p_i)_{i=1,\ldots,k}$ of Feasible we create the following ϵ-ETR instance ψ, where all of the following are required to hold.

$$t \cdot p_i(x) \leq_\epsilon 0 \qquad \text{for all } i, \tag{7.1}$$

$$t \cdot p_i(x) \geq_\epsilon 0 \qquad \text{for all } i, \tag{7.2}$$

$$t \geq \epsilon / \operatorname{gap}(P), \tag{7.3}$$

where the final inequality is implemented using the construction given above.

Lemma 25. *ψ is satisfiable if and only if P has a solution.*

Proof. First, let us assume that P has a solution. This means that there exists an $x \in \mathbb{R}^n$ such that $p_i(x) = 0$ for all i. Note that x clearly satisfies inequalities (7.1) and (7.2), while inequality (7.3) can be satisfied by fixing t to be any number greater than $\epsilon / \operatorname{gap}(P)$. So we have proved that ψ is satisfiable.

On the other hand, now we will assume that $x \in \mathbb{R}^n$ satisfies ψ. Note that we must have

$$p_i(x) \leq \epsilon / t \leq \operatorname{gap}(P)$$

and likewise

$$p_i(x) \geq -\epsilon / t \geq -\operatorname{gap}(P),$$

and hence $|p_i(x)| \leq \operatorname{gap}(P)$ for all i. But Lemma 24 states that this is only possible in the case where P has a solution. □

This completes the proof of the following theorem.

Theorem 28. *ϵ-ETR = ETR for all $\epsilon \geq 0$.*

7.2.1.2 Constrained ϵ-ETR

In our negative result for unconstrained ϵ-ETR, we abused the fact that variables could be arbitrarily large to construct the doubly-exponentially large number t. So, it makes sense to ask whether ϵ-ETR gets easier if we *constrain* the problem so that variables cannot be arbitrarily large.

In this chapter, we consider ϵ-ETR problems that are constrained by a bounded convex set in $\mathbb{R}^q$. For vectors $c_1, c_2, \ldots, c_l \in \mathbb{R}^q$ we use $\text{conv}(c_1, c_2, \ldots, c_l)$ to denote the set containing every vector that lies in the convex hull of c_1 through c_l. In the *constrained* ϵ-ETR, we require that the solution of the ϵ-ETR problem should also lie in the convex hull of c_1 through c_l.

Definition 23. *Given vectors $c_1, c_2, \ldots, c_l \in \mathbb{R}^q$ and a Boolean formula F that uses the operators $\{<_\epsilon, \leq_\epsilon, \geq_\epsilon, >_\epsilon\}$, the constrained ϵ-ETR problem asks us to decide whether*

$$\exists x_1 \exists x_2 \ldots \exists x_q \cdot \big(x \in \text{conv}(c_1, c_2, \ldots, c_l) \wedge F(x)\big).$$

Note that, unlike the constraints used in F, the convex hull constraints are not weakened. So the resulting solution $x_1, x_2, \ldots, x_q$, must actually lie in the convex set.

7.3 Approximating Constrained ϵ-ETR

7.3.1 Polynomial classes

To state our main theorem, we will use a certain class of polynomials where the coefficients are given as a tensor. This will be particularly useful when we apply our theorem to certain problems, such as normal form games. To be clear though, this is not a further restriction on the constrained ϵ-ETR problem, since all polynomials can be written down in this form.

In the sequel, we use the term *variable* to refer to a p-dimensional vector; for example, in Definition 23, the q-dimensional vector x would be called variable under this new naming. The variables of the polynomials we will study will be p-dimensional vectors denoted as $x_1, x_2, \ldots, x_n$, where $x_j(i)$ will denote the i-th element ($i \in [p]$) of vector x_j. The coefficients of the polynomials will be a captured by tensor denoted by A. Given a $\times_{j=1}^{n} p$ tensor A, we denote by $a(i_1, \ldots, i_n)$ its element with coordinates $(i_1, \ldots, i_n)$ on the tensor's dimensions $1, \ldots, n$, respectively, and by α we denote the maximum absolute value of these elements. We define the following two classes of polynomials.

- **Simple tensor multivariate.**

  We will use $\text{STM}(A, x_1^{d_1}, \ldots, x_n^{d_n})$ denote an STM polynomial with n variables where each variable x_j, $j \in [n]$ is applied d_j times on tensor A that defines the coefficients. Tensor A has $\sum_{j=1}^{n} d_j$ dimensions with p indices each. We will say that an STM polynomial is of maximum degree d, if $d = \max_j d_j$. Here is an example of a degree 2 simple tensor polynomial with two variables:

$$\text{STM}(A, x^2, y) = \sum_{i=1}^{p} \sum_{j=1}^{p} \sum_{k=1}^{p} x(i) \cdot x(j) \cdot y(k) \cdot a(i,j,k) + 10.$$

  This polynomial itself is written as follows.

$$\text{STM}(A, x_1^{d_1}, \ldots, x_n^{d_n}) = \sum_{i_{1,1} \in [p]} \cdots \sum_{i_{n,d_n} \in [p]} (x_1(i_{1,1})) \cdot \ldots \cdot (x_1(i_{1,d_1})) \cdot \ldots \cdot (x_n(i_{n,1})) \cdot \ldots \cdot (x_n(i_{n,d_n})) \cdot \cdot a(i_{1,1}, \ldots, i_{1,d_1} \ldots, i_{n,1}, \ldots, i_{n,d_n}) + a_0.$$

- **Tensor multivariate.** A tensor multivariate (TMV) polynomial is the sum over a number of simple tensor multivariate polynomials. We will use $\text{TMV}(x_1, \ldots, x_n)$ to denote a tensor multivariate polynomial with n vector variables, which is formally defined as

$$\text{TMV}(x_1, \ldots, x_n) = \sum_{i \in [t]} \text{STM}(A_i, x_1^{d_{i1}}, \ldots, x_n^{d_{in}}),$$

  where the exponents $d_{i1}, \ldots, d_{in}$ depend on i, and t is the number of simple multivariate polynomials. We will say that $\text{TMV}(x_1, \ldots, x_n)$ has length t if it is the sum of t STM polynomials, and that it is of degree d if $d = \max_{i \in [t], j \in [n]} d_{ij}$. Observe that $t \leq (d+1)^n$; it could be the case that a TMV polynomial is a sum of STM polynomials, each of which has a distinct combination of exponents $d_{i1}, \ldots, d_{in}$ in its variables, where each $d_{ij} \in \{0, 1, \ldots, d\}$.

7.3.2 ϵ-ETR with tensor constraints

We focus on ϵ-ETR instances F where all constraints are of the form $\text{TMV}(x_1, \ldots, x_n) \bowtie 0$, where $\bowtie$ is an operator from the set $\{<_\epsilon, \leq_\epsilon, >_\epsilon, \geq_\epsilon\}$. Recall that each TMV constraint

considers vector variables. We consider the number of variables used in F (denoted as n) to be the number of vector variables used in the TMV constraints. So the value of n used in our main theorem may be constant if only a constant number of vectors are used, even if the underlying ϵ-ETR instance actually has a non-constant number of variables. For example, if x and y and w are p-dimensional probability distributions and A_1 and A_2 are $p \times p$ tensors, the TMV constraint $x^T A_1 y + w^T A_2 x > 0$ has three variables, degree 1, length two; though the underlying problem has $3 \cdot p$ variables.

Note that every ϵ-ETR constraint can be written as a TMV constraint, because all multivariate polynomials can be written down as a TMV polynomial. Every term of a TMV can be written as a STM polynomial where the tensor entry is non zero for exactly the combination of variables used in the term, and 0 otherwise. Then a TMV polynomial can be constructed by summing over the STM polynomial for each individual term.

7.3.2.1 The main theorem

Given an ϵ-ETR formula F, we define exact(F) to be a Boolean formula in which every approximate constraint is replaced with its exact variant, meaning that every instance of $x \leq_\epsilon y$ is replaced with $x \leq y$, and likewise for the other operators.

Our main theorem is as follows.

Theorem 29. *Let F be an ϵ-ETR instance with n vector variables and m multivariate-polynomial constraints each one of maximum length t and maximum degree d, constrained by the convex hull defined by $c_1, c_2, \ldots, c_l \in \mathbb{R}^{np}$. Let α be the maximum absolute value of the coefficients of constraints of F, and let $\gamma = \max_i \|c_i\|_\infty$. If* exact($F$) *has a solution in* $\mathrm{conv}(c_1, c_2, \ldots, c_l)$*, then F has a k-uniform solution in* $\mathrm{conv}(c_1, c_2, \ldots, c_l)$ *where*

$$k = \frac{512 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^6 \cdot d^6 \cdot t^5 \cdot \ln(2 \cdot \alpha' \cdot \gamma' \cdot d \cdot n \cdot m)}{\epsilon^5},$$

where $\alpha' := \max(\alpha, 1), \gamma' := \max(\gamma, 1)$.

7.3.2.2 Consequences of the main theorem

Our main theorem gives a QPTAS for approximating a fragment of ϵ-ETR. The total number of k-uniform vectors in a convex set $C = \mathrm{conv}(c_1, c_2, \ldots, c_l)$ is $l^{O(k)}$. So, if the parameters α, γ, d, t, and n are all polylogarithmic in m, then our main theorem tells us that the total number of k-uniform vectors is $l^{O(\mathrm{poly}\log m)}$, where m is the number of constraints. So if

we enumerate each k-uniform vector x, we can check whether F holds, and if it does, we can output x. If no k-uniform vector exists that satisfies F, then we can determine that exact(F) has no solution. This gives us the following result.

Corollary 14. *Let F be an ϵ-ETR instance constrained by the convex hull defined by $c_1, c_2, \ldots, c_l$. If α, γ, n, d, and t are polylogarithmic in m, then we have an algorithm and runs in time $l^{O\left(\frac{poly \log m}{\epsilon^5}\right)}$ that either finds a solution to F, or determines that* exact(F) *has no solution.*

Let N be the input size of the given problem. If m is constant and l is polynomial in N then this gives a PTAS, while if m and l are polynomial in N, then this gives a QPTAS.

In Section 7.5 we will show that the problem of approximating the best social welfare achievable by an approximate Nash equilibrium in a two-player normal form game can be written down as a constrained ϵ-ETR formula where α, γ, d, and n are constant (and recall that $t \leq (d+1)^n$). It has been shown that, assuming the exponential time hypothesis, this problem cannot be solved faster than quasi-polynomial time [33,61], so this also implies that constrained ϵ-ETR where α, γ, d, and n are constant cannot be solved faster than quasi-polynomial time unless the exponential time hypothesis is false.

Many ϵ-ETR problems are naturally constrained by sets that are defined by the convex hull of exponentially many vectors. The cube $[0,1]^p$ is a natural example of one such set. Brute force enumeration does not give an efficient algorithm for these problems, since we need to enumerate $l^{O(k)}$ vectors, and l is already exponential in the dimension parameter p. However, our main theorem is able to provide non-deterministic polynomial time algorithms for these problems.

This is because each k-uniform vector is, by definition, the convex combination of at most k of the vectors in the convex set, and this holds even if l is exponential. So, provided that k is polynomial in the input size, we can guess the subset of vectors that are used, and then verify efficiently that the formula holds. This is particularly useful for problems where exact(F) always has a solution, which is often the case in game theory applications, since it places the approximation problem in NP, whereas deciding the existence of an exact solution may be ETR-complete.

Corollary 15. *Let F be an ϵ-ETR instance constrained by the convex hull defined by $c_1, c_2, \ldots, c_l$. If α, γ, d, t, n, are polynomial in the input size, then there is a non-deterministic polynomial time algorithm that either finds a solution to F, or determines*

that exact(F) *has no solution. Moreover, if* exact(F) *is guaranteed to have a solution, then the problem of finding an approximate solution for* F *is in* NP.

7.3.2.3 Approximation notions

According to the relaxation procedure for ETR that we have described, each atom A_i of the ETR formula is relaxed additively by a positive quantity ϵ. The main theorem (Theorem 29) and the intermediate resluts, give a sufficiently fine discretization (distance at most $1/k$ for some $k \in \mathbb{N}^*$) of the domain of the ETR instance's variables, such that if there exists an exact solution $x^* = (x_1^*, \ldots, x_n^*)$ of the formula then there exists a k-uniform solution in the discretized domain that ϵ-satisfies every A_i. In particular we prove that if $A_i = (p(x) \bowtie 0)$, where $p(x)$ is a multivariate polynomial and $\bowtie \in \{<, \leq, =, \geq, >\}$ then there exists a k-uniform vector x' such that $|p(x') - p(x^*)| \leq \epsilon$. This implies the ϵ-satisfaction of each A_i by the triangle inequality.

In fact, by this work we do not aim to reply an "approximate yes/no" to an ETR instance, i.e. to give a yes/no answer to the relaxed ETR instance, but instead to output an *approximate solution* (if an exact solution exists) to the ETR instance. Therefore, more accurately we should refer to this approximation of ETR as an approximation of Function ETR (FETR), where FETR is the function problem extension of the decision problem complexity class ETR. As ETR is the analogue of NP, FETR is the analogue of FNP in the Blum-Shub-Smale computation model [28].

Definition 24 (ϵ-approximation)**.** *Consider a given* ETR *instance with domain* D *and formula* F*. If* x^* *is a solution to the instance and* x' *is a solution to the respective* ϵ*-ETR instance for a given* $\epsilon > 0$*, then* x' *is called an* ϵ*-approximation of* x^**.*

Definition 25 (PTAS/QPTAS)**.** *Consider a function problem* P *with input size* N*, whose objective is to output a solution* x^**. An algorithm that computes an* ϵ*-approximation* x' *of* P *in time polynomial in* N *for any fixed* $\epsilon > 0$ *is a Polynomial Time Approximation Scheme (PTAS). An algorithm that computes* x' *in time* $O(N^{poly \log N})$ *is a Quasi-Polynomial Time Approximation Scheme (QPTAS).*

Remark 1. *Our technique that finds an* x' *such that* $|p(x') - p(x^*)| \leq \epsilon$ *provides one with more power than showing that polynomial inequalities weakened by* ϵ *hold for* x'*. In fact, it allows for approximation of solutions that need not be described by an* ETR *formula. A simple example of such a case is the one presented in Section 7.4.1 where we seek an*

approximation of the maximum of the quadratic function in the simplex. The maximization objective does not need to be written in an ETR *formula. Instead, we show that any point $f(x)$ of the quadratic function, for x in the simplex, can be approximated by a point $f(x')$ where x' is in a discrete simplex with a small number of points. Then we find the maximum of $f(x')'s$ which is smaller than $\max(f(x))$ by at most ϵ.*

The fact that operation "max" can be executed in time linear in the number of points of the discretized simplex allows us to use our method for expressions with "max" which is forbidden in the grammar of ETR*. More generally, the following theorem shows that even more complicated objectives, such as "$\max_{x_1} \min_{x_2}$" can be treated by a modification of the algorithm described in Section 7.3.2.2.*

Theorem 30. *Let F be a multi-objective optimization instance whose objective functions are multivariate polynomials, with variables constrained by the convex hull defined by $c_1, c_2, \ldots, c_l$. Let k be the quantity specified in Theorem 29 with m being the number of polynomial functions in the instance, meaning the ones in the objectives and constraints. If every objective on the functions has a polynomial time algorithm to be performed on a discrete domain, then there is an algorithm that runs in time $l^{O(k)}$, and either finds a solution which satisfies every objective of F within additive ϵ, or determines that F has no solution.*

Proof. As explained at the beginning of this section, our technique discretizes the domain of the variables with a density sufficient to approximate any point of any of the polynomial functions that are given as part of the atoms of an ETR formula. That is, for any x^* in the continuous domain it guarantees the existence of a discrete x' such that for every polynomial p in the atoms, it is $|p(x') - p(x^*)| \leq \epsilon$. Note now that the technique works for any given set of polynomials when we require that for every polynomial in the set, every point x^* has a discrete x'. This is regardless of what the atoms' operators from $\{<, \leq, =, \geq, >\}$ are or with what logical operators from $\{\wedge, \vee\}$ the atoms connect to each other.

In view of the above, observe that any objective (with the properties of the statement of the theorem) on functions, takes time polynomial in the size of the discretized space, therefore it does not change asymptotically the total running time of the algorithm described at the beginning of Section 7.3.2.2. That is because first, the aforementioned algorithm will brute-force through all of the points in the discretized domain and for these points it will check if all of the constraints of F are satisfied. Now the algorithm we propose will deviate from the aforementioned algorithm and for the points that satisfy the constraints of F (feasible points), for each objective it will run the efficient respective algorithm of

the objective on the feasible points and check whether all objectives of the relaxed by ϵ instance are satisfied for some point. This can be done in time polynomial in the size of the discretized domain, i.e. $l^{O(k)}$. If a discrete point is found that ϵ-satisfies F, then the algorithm returns it, otherwise there no point in the continuous domain that satisfies F according to Theorem 29. □

7.3.3 A theorem for non-tensor constraints

One downside of Theorem 29 is that it requires that the formula is written down using tensor constraints. We have argued that every ETR formula can be written down in this way, but the translation introduces a new vector-variable for each variable in the ETR formula. When we apply Theorem 29 to obtain PTASs or QPTASs we require that the number of vector variables is at most polylogarithmic, and so this limits the application of the theorem to ETR formulas that have at most polylogarithmically many variables.

Theorem 32 is a sampling result for ϵ-ETR with non-tensor constraints, which is proved via some intermediate results. First, we will use the following theorem of Barman.

Theorem 31 ([20])**.** *Let $c_1, c_2, \ldots, c_l \in \mathbb{R}^q$ with $\max_i \|c_i\|_\infty \leq 1$. For every $x \in conv(c_1, c_2, \ldots, c_l)$ and every $\epsilon > 0$ there exists a $O(\log l/\epsilon^2)$-uniform vector $x' \in \mathrm{conv}(c_1, c_2, \ldots, c_l)$ such that $\|x - x'\|_\infty \leq \epsilon$.*

The following lemma shows that if we take two vectors x and x' that are close in the L_∞ norm, then for all polynomials p the value of $|p(x) - p(x')|$ cannot be too large.

We denote by $consts(p)$ the maximum absolute coefficient in polynomial p, and by $terms(p)$ the number of terms of p.

Lemma 26. *Let $p(x)$ be a multivariate polynomial over $x \in \mathbb{R}^q$ with degree d and let $\epsilon \in (0, \gamma]$ for some constant $\gamma > 0$. For every pair of vectors $x, x' \in [0, \gamma]^q$ with $\|x - x'\|_\infty \leq \epsilon$ we have:*

$$|p(x) - p(x')| \leq \gamma^{d-1} \cdot (2^d - 1) \cdot consts(p) \cdot terms(p) \cdot \epsilon.$$

Proof. Consider a term of $p(x)$, which can without loss of generality be written as $t(x) = c \cdot \prod_{\substack{i \in [q] \\ \sum_i = d}} x_i$, where it could be the case that any number of x_i's are the same. We have

$$
\begin{aligned}
|t(x) - t(x')| &= \left| c \cdot \prod_{\substack{i \in [q] \\ \sum_i = d}} x_i - c \cdot \prod_{\substack{i \in [q] \\ \sum_i = d}} x'_i \right| \\
&= c \cdot \left| \prod_{\substack{i \in [q] \\ \sum_i = d}} x_i - \prod_{\substack{i \in [q] \\ \sum_i = d}} x'_i \right| \\
&\leq c \cdot \left| \prod_{\substack{i \in [q] \\ \sum_i = d}} x_i - \prod_{\substack{i \in [q] \\ \sum_i = d}} (x_i + \epsilon) \right| \\
&\leq c \cdot \left| \prod_{\substack{i \in [q] \\ \sum_i = d}} x_i - \left[\prod_{\substack{i \in [q] \\ \sum_i = d}} x_i + \binom{d}{1} \gamma^{d-1} \epsilon + \binom{d}{2} \gamma^{d-2} \epsilon^2 + \cdots + \binom{d}{d} \gamma^0 \epsilon^d \right] \right| \\
&\leq c \cdot \left| \epsilon \cdot \sum_{k=1}^{d} \binom{d}{k} \gamma^{d-1} \right| \\
&\leq c \cdot \epsilon \cdot \gamma^{d-1} \cdot \sum_{k=1}^{d} \binom{d}{k} \\
&= \epsilon \cdot c \cdot \gamma^{d-1} \cdot (2^d - 1),
\end{aligned}
$$

where the second to last four lines use the fact that x_i's, and ϵ are all less than or equal to γ.

Next consider a term $t(x)$ of $p(x)$ of degree $d' \leq d$. This can be written similarly to the aforementioned term. Then $|t(x) - t(x')| \leq c \cdot \epsilon \cdot \gamma^{d-1} \cdot (2^{d'} - 1) \leq c \cdot \epsilon \cdot \gamma^{d-1} \cdot (2^d - 1)$. Since there are $terms(p)$ many terms in p, we therefore have that

$$|p(x) - p(x')| \leq \gamma^{d-1} \cdot (2^d - 1) \cdot consts(p) \cdot terms(p) \cdot \epsilon.$$

□

We now apply this to prove the following theorem.

Theorem 32. *Let F be an ϵ-ETR instance constrained over the convex hull defined by $c_1, c_2, \ldots, c_l \in \mathbb{R}^q$. Let m be the number of constraints used in F, Let $\gamma = \max_i \|c_i\|_\infty$, let α be the largest constant coefficient used in F, let t be the number of terms used in total in all polynomials of F, and let d be the maximum degree of the polynomials in F. If* $\text{exact}(F)$ *has a solution in* $\text{conv}(c_1, c_2, \ldots, c_l)$*, then F has a k-uniform solution in* $\text{conv}(c_1, c_2, \ldots, c_l)$ *where*

$$k = \alpha^2 \cdot \gamma^{2d-2} \cdot (2^d - 1)^2 \cdot t^2 \cdot \log l/\epsilon^2.$$

Proof. Let x be the solution to $\text{exact}(F)$. First we apply Theorem 31 to find a point y that is k-uniform, where $k = \alpha^2 \cdot \gamma^{2d-2} \cdot (2^d - 1)^2 \cdot t^2 \cdot \log l/\epsilon^2$, such that

$$\|x - y\|_\infty \leq \epsilon/(\alpha \cdot \gamma^{d-1} \cdot (2^d - 1) \cdot t).$$

Next we can apply Lemma 26 to argue that, for each polynomial p used in F, we have

$$\begin{aligned} |p(x) - p(y)| &\leq \alpha \cdot \gamma^{d-1} \cdot (2^d - 1) \cdot t \cdot \left(\frac{\epsilon}{\alpha \cdot \gamma^{d-1} \cdot (2^d - 1) \cdot t} \right) \\ &= \epsilon. \end{aligned}$$

Since all constraints of F have a tolerance of ϵ, and since x satisfies $\text{exact}(F)$, we can conclude that $F(y)$ is satisfied. □

The key feature here is that the number of variables does not appear in the formula for k, which allows the theorem to be applied to some formulas for which Theorem 29 cannot. However, since the theorem does not allow tensor constraints, its applicability is more limited because the number of terms t will be much larger in non-tensor formulas. For example, as we will see in Section 7.5, we can formulate bimatrix games using tensor constraints over constantly many vector variables, and this gives a result using Theorem 29. No such result can be obtained via Theorem 32, because when we formulate the problem without tensor constraints, the number of terms t used in the inequalities becomes polynomial in the dimension.

7.4 The Proof of the Main Theorem

In this section we prove Theorem 29. Before we proceed with the technical results, let us illustrate via an example the crucial idea for proving that the special vectors we have defined (i.e. the k-uniform vectors for some $k \in \mathbb{N}^*$) inside a discretized convex hull can be used to approximate not only multilinear polynomials, but also multivariate polynomials of degree $d \geq 2$. At the same time, we show that the discretization of the domain (points in distance at most $1/k$ from each other) does not need to be very fine in order to achieve an additive approximation ϵ at any point of such a function. Our example is in approximating the quadratic polynomial over the simplex.

Let us provide a roadmap for this section. We begin by the detailed aforementioned example. Then we proceed by considering two special cases, namely Lemma 28 and Lemma 30, which when combined will be the backbone of the proof of the main theorem.

Firstly, we will show how to deal with problems where every constraint of the Boolean formula is a *multilinear polynomial*, which we will define formally later. We deal with this kind of problems using Hoeffding's inequality and the union bound, which is similar to how such constraints have been handled in prior work.

Then, we study problems where the Boolean formula consists of a *single* degree d polynomial constraint. We reduce this kind of problems to a constrained $\epsilon/2$-ETR problem with multilinear constraints, so we can use our previous result to handle the reduced problem. Sampling techniques in degree d polynomial problems have not been considered in previous work, and so this reduction is a novel extension of sampling based techniques to a broader class of ϵ-ETR formulas.

Finally, we deal with the main theorem: we reduce the original ETR problem with multivariate constraints to a set of ϵ'-ETR problems with a single standard degree d constraint, and then we use the last result to derive a bound on k.

As a byproduct of our main result one can get the same result as that of [54] in which a PTAS for fixed degree polynomial minimization over the simplex was presented. Even though the PTAS that follows from our result on the same optimization problem has roughly the same running time as that of [54], the proof presented in this chapter (which is independent of the aforementioned work) is significantly simpler. Nevertheless, the result in the current chapter generalizes previous results on polynomial optimization over the simplex, by providing a universal algorithm for multi-objective optimization problems, and showing how its running time depends on the parameters of the problem (see Theorem 30).

7.4.1 Example: A simple PTAS for quadratic polynomial optimization over the simplex

Definition 26 (Standard quadratic optimization problem (SQP)). *Given a $p \times p$ matrix A with entries normalized in $[0,1]$, find the value*

$$v^* := \max_{x \in \Delta_p} x^T A x, \quad \textit{where } \Delta_p \textit{ is the } (p-1) - \textit{simplex.}$$

SQP is a strongly NP-hard problem, even for the case where A has entries in $\{0,1\}$; in a theorem of Motzkin and Straus [111] it is shown that if matrix A is the adjacency matrix of a graph on p vertices whose maximum clique has c vertices, then $v^* = 1 - 1/c$. The problem of finding the size of the maximum clique in a general graph is known to be (strongly) NP-hard since its decision version is one of Karp's 21 NP-complete problems [88]. Therefore, unless P = NP there is no Fully Polynomial Time Approximation Scheme for SQP and the best thing we can hope for the problem is a PTAS. We present a PTAS for SQP (Corollary 16), which has almost the same running time as that of [29], but we claim that our proof is significantly simpler.

Let $x^* := \arg(v^*)$. Consider the set $\Delta_p(k)$ of all k-uniform vectors, for $k = 16\ln(3/\epsilon)/\epsilon^2$, with items $x^{(i)} \in \Delta_p(k)$, for $i = 1, 2, \ldots, |\Delta_p(k)|$.

Lemma 27. *There exists a multiset $\mathcal{X}$ of $\Delta_p(k)$ with $|\mathcal{X}| = 2/\epsilon$ such that for every $x^{(i)}, x^{(j)} \in \mathcal{X}$ with $i \neq j$, it is*

$$x^{*T} A x^* - x^{(i)T} A x^{(j)} < \epsilon/2.$$

Proof. Note that although $i \neq j$, $x^{(i)}$ could be equal to $x^{(j)}$ since the two k-uniform vectors belong to a multiset of $\Delta_p(k)$. The proof is by the probabilistic method. Let us create the events

$$\begin{aligned}
E_i &= \left\{ x^{*T} A x^* - x^{(i)T} A x^* < \epsilon/4 \right\}, && \forall i \text{ for which } x^{(i)} \in \mathcal{X}, \\
F_{i,j} &= \left\{ x^{(i)T} A x^* - x^{(i)T} A x^{(j)} < \epsilon/4 \right\}, && \forall i,j \text{ with } i \neq j, \text{ for which } x^{(i)}, x^{(j)} \in \mathcal{X}, \\
G_{i,j} &= \left\{ x^{*T} A x^* - x^{(i)T} A x^{(j)} < \epsilon/2 \right\}, && \forall i,j \text{ with } i \neq j, \text{ for which } x^{(i)}, x^{(j)} \in \mathcal{X}.
\end{aligned}$$

Observe that $E_i \cap F_{i,j} \subseteq G_{i,j}$. Now, let each of k i.i.d. random variables be drawn from x^*. The sample space for each is $[p]$. For any $x^{(i)}, x^{(j)} \in \Delta_p(k)$, the expectation of $x^{(i)T} A x^*$

is $x^{*T}Ax^*$, and the expectation of $x^{(i)T}Ax^{(j)}$ (for fixed $x^{(i)}$) is $x^{(i)T}Ax^*$. Let us denote $r := |\mathcal{X}| = 2/\epsilon$. By using a Höffding bound [86], we get

$$\begin{aligned}\Pr\{\overline{E_i}\} &\leq e^{-k\epsilon^2/8}, \quad \forall i \text{ for which } x^{(i)} \in \mathcal{X}, \text{ and}\\ \Pr\{\overline{F_{i,j}}\} &\leq e^{-k\epsilon^2/8}, \quad \forall i,j \text{ with } i \neq j, \text{ for which } x^{(i)}, x^{(j)} \in \mathcal{X}.\end{aligned}$$

Consider now the event H that captures the condition that needs to be satisfied by the lemma. It is

$$H = \bigcap_{\substack{i,j\in\mathcal{X}\\ i\neq j}} G_{i,j}.$$

Therefore

$$\overline{H} = \bigcup_{\substack{i,j\in\mathcal{X}\\ i\neq j}} \overline{G_{i,j}} \subseteq \bigcup_{i\in\mathcal{X}} \overline{E_i} \bigcup_{\substack{i,j\in\mathcal{X}\\ i\neq j}} \overline{F_{i,j}}.$$

Hence

$$\begin{aligned}\Pr\{\overline{H}\} &\leq re^{-k\epsilon^2/8} + r(r-1)e^{-k\epsilon^2/8}\\ &= r^2e^{-k\epsilon^2/8}\\ &< 1.\end{aligned}$$

□

The above strict inequality means that $\Pr\{H\} > 0$, therefore, there exists a set $\mathcal{X}$ that satisfies the statement of the lemma.

The following theorem corresponds to the general Lemma 30, for the case $\alpha = \gamma = 1$, $d = 2$.

Theorem 33. *There exists a $\frac{32\ln(3/\epsilon)}{\epsilon^3}$-uniform vector x, such that $v^* - x^TAx < \epsilon$.*

Proof. Consider the multiset $\mathcal{X}$ of $\Delta_p(k)$ of Lemma 27, and recall that $r := |\mathcal{X}| = 2/\epsilon$. Let us create the vector

$$x := \frac{1}{r}\sum_{i\in\mathcal{X}} x^{(i)}.$$

Then, it is

$$\begin{aligned}
x^{*T}Ax^* - x^TAx &= x^{*T}Ax^* - \left(\frac{1}{r}\sum_{x^{(i)}\in\mathcal{X}} x^{(i)T}\right) A \left(\frac{1}{r}\sum_{x^{(i)}\in\mathcal{X}} x^{(i)}\right) \\
&= x^{*T}Ax^* - \frac{1}{r^2}\sum_{x^{(i)},x^{(j)}\in\mathcal{X}} x^{(i)T}Ax^{(j)} \\
&= x^{*T}Ax^* - \frac{1}{r^2}\left(\sum_{\substack{x^{(i)},x^{(j)}\in\mathcal{X}\\ i\neq j}} x^{(i)T}Ax^{(j)} + \sum_{x^{(i)}\in\mathcal{X}} x^{(i)T}Ax^{(i)}\right) \\
&= \frac{1}{r^2}\left(r(r-1)x^{*T}Ax^* - \sum_{\substack{x^{(i)},x^{(j)}\in\mathcal{X}\\ i\neq j}} x^{(i)T}Ax^{(j)} + rx^{*T}Ax^* - \sum_{x^{(i)}\in\mathcal{X}} x^{(i)T}Ax^{(i)}\right) \\
&< \frac{1}{r^2}\left(r(r-1)\frac{\epsilon}{2} + r\right) \\
&\leq \frac{\epsilon}{2} + \frac{1}{r} \\
&= \epsilon,
\end{aligned}$$

where the second to last inequality is implied from Lemma 27 which applies for every $x^{(i)}, x^{(j)} \in \mathcal{X}$ when $i \neq j$, and from the fact that $x^{*T}Ax^* - x^{(i)T}Ax^{(i)}$ is upper bounded by 1 for every $x^{(i)} \in \mathcal{X}$ (recall that the entries of A are in $[0, 1]$).

The proof is concluded by observing that the vector x we created is a kr-uniform vector, for $k = 16\ln(3/\epsilon)/\epsilon^2$ and $r = 2/\epsilon$. □

Corollary 16. *There is a PTAS for* SQP.

Proof. By Theorem 33, since the desired probability vector x that is suitable for the approximation is the mean of r many k-uniform vectors, x is kr-uniform. Therefore, it can be found by exhaustively searching through all possible multisets of $[p]$ created by sampling with replacement $kr = 32\ln(3/\epsilon)/\epsilon^3$ times. The number of all those possible multisets is $\binom{p+kr-1}{kr} \in O(p^{kr})$. For each multiset, i.e. vector x that the search algorithm takes into account, it picks the one that makes x^TAx maximum. This value is guaranteed to be ϵ-close to v^* by Theorem 33.

Hence, if we desire a $(1-\epsilon)$-approximation of SQP *in the weak sense* according to

Definition 2.2 of [52], the described algorithm runs in time $O\left(p^{\ln(\frac{3}{\epsilon})/\epsilon^3}\right)$. □

7.4.2 The general proof

7.4.2.1 Problems with multilinear constraints

We begin by considering constrained ϵ-ETR problems where the Boolean formula F consists of tensor-multilinear polynomial constraints. We will use $\text{TML}(A, x_1, \ldots, x_n)$ to denote a tensor-multilinear polynomial with n variables and coefficients defined by tensor A of size $\times_{j=1}^{n} p$. Formally,

$$\text{TML}(A, x_1, \ldots, x_n) = \sum_{i_1 \in [p]} \cdots \sum_{i_n \in [p]} x_1(i_1) \cdot \ldots \cdot x_n(i_n) \cdot a(i_1, \ldots, i_n) + c.$$

We will use α to denote the maximum entry of tensor A in the absolute value sense and γ to denote the infinite norm of the convex set that constrains the variables.

Lemma 28. *Let F be a Boolean formula with n variables and m tensor-multilinear polynomial constraints and let $\mathcal{Y}$ be a convex set in the variables space. If the constrained* ETR *problem defined by* exact(F) *and $\mathcal{Y}$ has a solution, then the constrained ϵ-ETR problem defined by F and $\mathcal{Y}$ has a k uniform solution where*

$$k = \frac{2 \cdot \alpha^2 \cdot \gamma^2 \cdot n^2 \cdot \ln(3 \cdot n \cdot m)}{\epsilon^2}.$$

Proof. Let $(x_1^*, x_2^*, \ldots, x_n^*) \in \mathcal{Y}$ be a solution for exact(F). Since we assume the $\mathcal{Y}$ is the convex hull of $c_1, \ldots, c_l$ any $x \in \mathcal{Y}$ can be written as a convex combination of the c_i's, i.e., $x = \sum_{i \in [l]} a_i \cdot c_i$, where $a_i \geq 0$ for every $i \in [l]$, and $\sum_{i \in [l]} a_i = 1$. Observe, $a = (a_1, \ldots, a_l)$ corresponds to a probability distribution over $c_1, \ldots, c_l$, where vector c_i is drawn with probability a_i, and x can be thought of as the mean of a. So, we can "sample" a point by sampling over c_i's according to the probability that defines this point.

For every $i \in [n]$, let x_i' be a k-uniform vector sampled independently from x_i^*. To prove the lemma, we will show that, because of the choice of k, with positive probability the sampled vectors satisfy every constraint of the ϵ-ETR problem. Then, by the probabilistic method the lemma will follow.

Let $\text{TML}_j(A_j, x_1, \ldots, x_n)$ be a multilinear polynomial that defines a constraint of F.

For every $j \in [m]$ we define the following event

$$|\operatorname{TML}_j(A_j, x'_1, \ldots, x'_n) - \operatorname{TML}_j(A_j, x^*_1, \ldots, x^*_n)| \leq \epsilon. \tag{7.4}$$

Observe that if $x'_1, \ldots, x'_n$ satisfy inequality (7.4) for every $j \in [m]$, then the lemma follows.

For every $j \in [m]$, we replace the corresponding event (7.4) with n events that are *linear* in each variable. For notation simplicity, let us denote by ML^i_j the multilinear polynomial $TML_j(A_j, x_1, \ldots, x_n)$ in which we have additionally set $x_1 = x'_1, x_2 = x'_2, \ldots, x_i = x'_i$ and $x_{i+1} = x^*_{i+1}, x_{i+2} = x^*_{i+2}, \ldots, x_n = x^*_n$. Furthermore, let $ML^0_j = ML_j(A_j, x^*_1, \ldots, x^*_n)$. Then, for every $i \in [n]$ consider the event

$$|ML^i_j - ML^{i-1}_j| \leq \frac{\epsilon}{n}. \tag{7.5}$$

Observe that, if for a given $j \in [m]$ all n events defined in (7.5) are satisfied, then by the triangle inequality, the corresponding event (7.4) is satisfied as well.

Consider now ML^i_j. This can be seen as a random variable that depends on the choice of x'_i and takes values in $[-\gamma \cdot \alpha, \gamma \cdot \alpha]$. But recall that the x'_i's are sampled from x^*_i using k samples, and that they are mutually independent, so $\mathbb{E}\big[ML^i_j\big] = ML^{i-1}_j$. Thus, we can bound the probability that a constraint (7.5) is not satisfied, i.e. bound the probability that $|ML^i_j - ML^{i-1}_j| > \frac{\epsilon}{n}$, using Hoeffding's inequality [86]. So,

$$\begin{aligned}
\Pr\left(\left|ML^i_j - ML^{i-1}_j\right| > \frac{\epsilon}{n}\right) &= \Pr\left(\left|ML^i_j - \mathbb{E}\big[ML^i_j\big]\right| > \frac{\epsilon}{n}\right) \\
&\leq 2 \cdot exp\left(-\frac{2 \cdot k^2 \cdot \left(\frac{\epsilon}{n}\right)^2}{4 \cdot k \cdot \gamma^2 \cdot \alpha^2}\right) \\
&= 2 \cdot exp\left(-\frac{k \cdot \epsilon^2}{2 \cdot n^2 \cdot \gamma^2 \cdot \alpha^2}\right).
\end{aligned} \tag{7.6}$$

Recall, that we have $n \cdot m$ events of the form (7.5). We can bound the probability that any of those events is violated, via the union bound. So, using (7.6) and the union bound, the probability that any of these events is violated is upper bounded by

$$2 \cdot m \cdot n \cdot exp\left(-\frac{k \cdot \epsilon^2}{2 \cdot n^2 \cdot \gamma^2 \cdot \alpha^2}\right). \tag{7.7}$$

Hence, if the value of (7.7) is strictly less than 1, then there are $x'_1, \ldots, x'_m$ such that all of

the $n \cdot m$ events of (7.5) are realized with positive probability, therefore the events of (7.4) are realized with positive probability and thus the lemma follows. By requiring (7.7) to be strictly less than 1, and solving for k we get

$$k > \frac{2 \cdot \alpha^2 \cdot \gamma^2 \cdot n^2 \cdot \ln(2 \cdot n \cdot m)}{\epsilon^2}$$

which holds, by our choice of k.

□

7.4.2.2 Problems with a standard degree d constraint

We now consider constrained ϵ-ETR problems with *exactly one* tensor polynomial constraint of standard degree d. We will use $\mathrm{TSD}(A, x, d)$ to denote a standard degree d tensor-polynomial with coefficients defined by the $\times_{j=1}^{d} p$ tensor A. Here, d identical vectors x are applied on A. Formally,

$$\mathrm{TSD}(A, x, d) = \sum_{i_1 \in [p]} \cdots \sum_{i_d \in [p]} x(i_1) \cdot \ldots \cdot x(i_d) \cdot a(i_1, \ldots, i_d) + c.$$

To prove the following lemma we consider the variable x to be defined as the average of $r = O(\frac{\alpha^2 \cdot \gamma^d \cdot d^2}{\epsilon})$ variables. This allows us to "break" the standard degree d tensor polynomial to a sum of multilinear tensor polynomials and to a sum of not-too-many multivariate polynomials. Then, the choice of r allows us to upper bound the error occurred by the multivariate polynomials by $\frac{\epsilon}{2}$. Then, we observe that in order to prove the lemma we can write the sum of multilinear tensor polynomials as an $\frac{\epsilon}{2}$-ETR problem with r variables and roughly r^d multilinear constraints. This allows us to use Lemma 28 to complete the proof.

Lemma 29. *Let F be a Boolean formula with one variable and one tensor-polynomial constraint of standard degree d, let $\mathcal{Y}$ be a bounded convex set, and let $r = \frac{2 \cdot \alpha^2 \cdot \gamma^d \cdot d^2}{\epsilon}$. If the constrained* ETR *problem* exact(F) *has a solution in $\mathcal{Y}$, then there exists a satisfiable constrained $\frac{\epsilon}{2}$-ETR problem Π_{ML} with r variables, where each variable is a k-uniform vector for $k = \frac{16 \cdot \alpha^4 \cdot \gamma^d \cdot d^4}{\epsilon^3}$. The Boolean formula of Π_{ML} is the conjunction of $\prod_{i=0}^{d-1}(r-i)$ tensor multilinear constraints, and every solution of Π_{ML} in $\mathcal{Y}$ can be transformed to a solution for the constrained ϵ-ETR problem defined by F and $\mathcal{Y}$.*

Proof. Assume that $x^* \in \mathcal{Y}$ is a solution for F. Let $\mathrm{TSD}(A, x, d)$ denote the tensor polynomial of standard degree d used in F. For notation simplicity, let $\mathrm{TSD}(A, x, d) = A(x^d)$. Cre-

ate r new k-uniform variables $x_1, \ldots, x_r \in \mathcal{Y}(k)$ by sampling each one from x^*, where $\mathcal{Y}(k)$ is the discretized set made from $\mathcal{Y}$ by using k-uniform vectors, and set $x = \frac{1}{r}(x_1 + \ldots + x_r)$. Let $\mathcal{X} = \bigcup_{i=1}^{r} \{x_i\}$ be a *multiset* of $\mathcal{Y}(k)$ with cardinality r, meaning that multiple copies of an element of $\mathcal{Y}(k)$ are allowed in $\mathcal{X}$. In the sequel we will treat the elements of $\mathcal{X}$ as distinct, even though some might correspond to the same element of $\mathcal{Y}(k)$. Then, note that $A(x^d)$ can be written as a sum of simple tensor-multivariate polynomials where some of them are multilinear and have as variables $x_1, \ldots, x_r$. Now, let $\mathcal{S}$ be the set of all ordered d-tuples that can be made by drawing d elements from $\mathcal{X}$ with replacement. Formally, $\mathcal{S} = \{(\hat{x}_1, \ldots, \hat{x}_d) : \hat{x}_1, \ldots, \hat{x}_d \in \mathcal{X}\}$. Let us also define $\mathcal{S}_d$ to be the set of all ordered d-tuples that can be made by drawing d elements from $\mathcal{X}$ without replacement. Formally, $\mathcal{S}_d = \{(\hat{x}_1, \ldots, \hat{x}_d) : \hat{x}_1, \ldots, \hat{x}_d \in \mathcal{X}, \quad \hat{x}_1, \ldots, \hat{x}_d \text{ are pairwise different}\}$, and observe that $|S_d| = \prod_{i=0}^{d-1}(r-i)$. So, any element of $\mathcal{S}_d$, combined with tensor A, produces a multilinear polynomial. Hence, using the notation introduced, we get that $|A(x^d) - A(x^{*^d})|$ is less than or equal to the sum of the following two sums

$$\frac{1}{r^d} \sum_{(\hat{x}_1,\ldots,\hat{x}_d) \in \mathcal{S}_d} \left| A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d}) \right| \qquad \text{and} \tag{7.8}$$

$$\frac{1}{r^d} \sum_{(\hat{x}_1,\ldots,\hat{x}_d) \in \mathcal{S} - \mathcal{S}_d} \left| A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d}) \right|. \tag{7.9}$$

Observe, $|\mathcal{S} - \mathcal{S}_d| = r^d - |\mathcal{S}_d|$ and that $\left| A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d}) \right| \leq \gamma^d \cdot \alpha$ for every

$A(\hat{x}_1, \ldots, \hat{x}_d)$. Then, for the sum given in (7.9) we get

$$\begin{aligned}
\frac{1}{r^d} \sum_{(\hat{x}_1,\ldots,\hat{x}_d)\in\mathcal{S}-\mathcal{S}_d} & \left|A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d})\right| \\
&\leq \left(1 - \frac{r \cdot (r-1) \cdots (r-d+1)}{r^d}\right) \cdot \gamma^d \cdot \alpha \\
&\leq \left(1 - \left(1 - \frac{1}{r}\right)\left(1 - \frac{2}{r}\right) \cdot \left(1 - \frac{d-1}{r}\right)\right) \cdot \gamma^d \cdot \alpha \\
&\leq \left(1 - \left(1 - \frac{d-1}{r}\right)^{d-1}\right) \cdot \gamma^d \cdot \alpha \\
&\leq \left(1 - \left(1 - \frac{(d-1)^2}{r}\right)\right) \cdot \gamma^d \cdot \alpha \quad \text{(Bernoulli's inequality)} \\
&= \frac{(d-1)^2}{r} \cdot \gamma^d \cdot \alpha \\
&\leq \frac{\epsilon}{2}.
\end{aligned}$$

Hence, in order for the original constraint to be satisfied, it suffices to satisfy the constraint

$$\frac{1}{r^d} \sum_{(\hat{x}_1,\ldots,\hat{x}_d)\in\mathcal{S}_d} \left|A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d})\right| \leq \frac{\epsilon}{2}. \tag{7.10}$$

Observe that $|\mathcal{S}_d| = \prod_{i=0}^{d-1}(r-i) < r^d$, therefore, instead of the constraint (7.10), it suffices to satisfy the following $|S_d|$ constraints (we introduce one constraint for every $(\hat{x}_1, \ldots, \hat{x}_d) \in \mathcal{S}_d$)

$$\left|A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d})\right| \leq \frac{\epsilon}{2}. \tag{7.11}$$

Note that each constraint (7.11) is the relaxed by $\epsilon/2$ version of a constraint with a multilinear function equal to 0; multilinearity is due to the fact that $\hat{x}_1, \ldots, \hat{x}_d$ are pairwise different by definition of the set S_d. The proof is completed by using Lemma 28 for $n = d$, $m = |S_d|$ and $\epsilon/2$ instead of ϵ to show that indeed there exists a collection S_d of tuples $\hat{x}_1, \ldots, \hat{x}_d$, where each $\hat{x}_i$, $i \in [d]$ is a k-uniform vector with $k \geq \frac{8 \cdot \alpha^2 \cdot \gamma^2 \cdot d^2 (d+2) \cdot \ln r}{\epsilon^2}$ such that all $|S_d|$ constraints of (7.11) are satisfied. The latter inequality is true by our choice of k and r.

□

Now we can prove the following lemma.

Lemma 30. *Let F be a Boolean formula with variable x and one tensor-polynomial constraint of standard degree d, and let $\mathcal{Y}$ be a bounded convex set. If the constrained* ETR *problem defined by* exact(F) *and $\mathcal{Y}$ has a solution, then the constrained ϵ-ETR problem defined by F and $\mathcal{Y}$ has a k-uniform solution where*

$$k = \frac{32 \cdot \alpha^6 \cdot \gamma^{2d} \cdot d^6}{\epsilon^4}.$$

Proof. First, we use Lemma 29 to construct the constrained $\frac{\epsilon}{2}$-ETR problem Π_{ML} with tensor-multilinear constraints. Recall that Π_{ML} has $r = \frac{2 \cdot \alpha^2 \cdot \gamma^d \cdot d^2}{\epsilon}$ variables and if Π_{ML} is satisfiable, then there exist $\frac{k}{r}$-uniform vectors $\hat{x}_1 \in \mathcal{Y}, \ldots, \hat{x}_r \in \mathcal{Y}$ that $\epsilon/2$-satisfy Π_{ML}. Then, let us construct the k-uniform vector $\hat{x} = \frac{1}{r} \cdot (\hat{x}_1 + \ldots + \hat{x}_r)$. Note that, according to Lemma 29, it is

$$\begin{aligned}
|A(\hat{x}^d) - A(x^{*^d})| &\leq \frac{1}{r^d} \sum_{(\hat{x}_1,\ldots,\hat{x}_d) \in \mathcal{S}_d} \left| A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d}) \right| \\
&\quad + \frac{1}{r^d} \sum_{(\hat{x}_1,\ldots,\hat{x}_d) \in \mathcal{S} - \mathcal{S}_d} \left| A(\hat{x}_1, \ldots, \hat{x}_d) - A(x^{*^d}) \right| \\
&\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \\
&= \epsilon.
\end{aligned}$$

This completes the proof of the lemma. □

7.4.2.3 Problems with simple multivariate constraints

We now assume that we are given a constraint-ϵ-ETR problem defined by a Boolean formula F of tensor simple multilinear polynomial constraints and a bounded convex set $\mathcal{Y}$. As before $\gamma = \|\mathcal{Y}\|_\infty$ and let α be the maximum absolute value of the coefficients of the constraints. We will say that the constraints are of maximum degree d if d is the maximum degree among all variables. The main idea of the proof of the following lemma is to rewrite the problem as an equivalent problem with standard degree d constraints and then apply Lemmas 30 and 28 to derive the bound for k.

Lemma 31. *Let F be a Boolean formula with n variables and m simple tensor-multivariate polynomial constraints of maximum degree d and let $\mathcal{Y}$ be a bounded convex set in the*

variables space. If the constrained ETR *problem defined by* exact(F) *and* $\mathcal{Y}$ *has a solution, then the constrained* ϵ*-ETR problem defined by* F *and* $\mathcal{Y}$ *has a* k*-uniform solution where*

$$k = \frac{512 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^6 \cdot d^6 \cdot \ln(2 \cdot \alpha' \cdot \gamma' \cdot d \cdot n \cdot m)}{\epsilon^5},$$

where $\alpha' := \max(\alpha, 1), \gamma' := \max(\gamma, 1)$.

Proof. Let $x_1^*, \ldots, x_n^*$ be a solution for exact(F) and let x_i', $i \in [n]$ be a variable k-uniform vector sampled from x_i^*. We will prove that if k equals at least the quantity of the statement of the lemma, then there exist vectors $x_1', \ldots, x_n'$ that constitute a solution to the constrained ϵ-ETR problem defined by F and $\mathcal{Y}$.

Consider the j-th constraint where $j \in [m]$ defined by the simple tensor-multivariate polynomial $\mathrm{STM}(A_j, x_1^{d_{j1}}, \ldots, x_n^{d_{jn}})$. We will use the same technique we used in Lemma 28 to create n constraints, where constraint $i \in [n]$ is defined via a simple degree d_{ji} polynomial. Again, for notation simplicity for every $i \in [m]$ we use STM_j^i to denote the polynomial $\mathrm{STM}(A_j, x_1^{d_{j1}}, \ldots, x_n^{d_{jn}})$ where we set $x_1 = x_1', \ldots, x_i = x_i'$ and $x_{i+1} = x_{i+1}^*, \ldots, x_n = x_n^*$. Let $\mathrm{STM}_j^0 := \mathrm{STM}(A_j, (x_1^*)^{d_{j1}}, \ldots, (x_n^*)^{d_{jn}})$. Then, for every $j \in [m]$ we define the following n constraints

$$|\,\mathrm{STM}_j^i - \mathrm{STM}_j^{i-1}\,| \leq \frac{\epsilon}{n}. \tag{7.12}$$

Observe that for some $j \in [m]$, every constraint i of the form (7.12) defines a simple degree d_{ji} polynomial with respect to variable x_i'. Furthermore, observe that if every such constraint is satisfied, then the initial constraint defined by $\mathrm{STM}(A_j, x_1^{d_{j1}}, \ldots, x_n^{d_{jn}})$ is satisfied too. Then, we convert each such constraint to a set of $\prod_{i=0}^{d-1}(r-i)$ multilinear constraints with $r = \frac{2 \cdot \alpha^2 \cdot \gamma^d \cdot d^2}{\epsilon}$ variables, using Lemma 29 where we demand that every multilinear constraint is $\frac{\epsilon}{2n}$-satisfied (we restrict the current $\frac{\epsilon}{n}$ to half of it in order to use Lemma 29). The proof is then completed by using Lemma 28 where we observe that we have $r \cdot n = \frac{2 \cdot \alpha^2 \cdot \gamma^d \cdot d^2 \cdot n}{\epsilon}$ variables and $\prod_{i=0}^{d-1}(r-i) \cdot n \cdot m < r^d \cdot n \cdot m$ constraints and we set ϵ to $\frac{\epsilon}{2n}$.

To arrive to the actual size k of the required uniform vector, we start from the size k' prescribed by Lemma 28 and sequentially set proper values for the parameters as dictated by our method of for transforming the constraints. We have

$$
\begin{aligned}
k' &= \frac{2 \cdot \alpha^2 \cdot \gamma^2 \cdot n^2 \cdot \ln(3 \cdot n \cdot m)}{\epsilon^2} \\
&= \frac{8 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^2 \cdot d^4 \cdot \ln(6 \cdot \alpha^2 \cdot \gamma^d \cdot d^2 \cdot n \cdot m/\epsilon)}{\epsilon^4} && (n \leftarrow n \cdot r) \\
&= \frac{8 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^2 \cdot d^4 \cdot \ln(6 \cdot \alpha^{2d+2} \cdot \gamma^{d^2+d} \cdot d^{2d+2} \cdot n^2 \cdot m/\epsilon^{d+1})}{\epsilon^4} && (m \leftarrow r^d \cdot n \cdot m) \\
&= \frac{128 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^6 \cdot d^4 \cdot \ln(6 \cdot 2^{d+1} \cdot \alpha^{2d+2} \cdot \gamma^{d^2+d} \cdot d^{2d+2} \cdot n^{d+3} \cdot m/\epsilon^{d+1})}{\epsilon^4} && (\epsilon \leftarrow \frac{\epsilon}{2n}) \\
&\leq \frac{128 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^6 \cdot d^4 \cdot \ln(2 \cdot \max(\alpha, 1) \cdot \max(\gamma, 1) \cdot d \cdot n \cdot m/\epsilon)^{4d^2}}{\epsilon^4} && (\text{for any } d \geq 1) \\
&= \frac{512 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^6 \cdot d^6 \cdot \ln(2 \cdot \max(\alpha, 1) \cdot \max(\gamma, 1) \cdot d \cdot n \cdot m/\epsilon)}{\epsilon^4} \\
&\leq \frac{512 \cdot \alpha^6 \cdot \gamma^{2d+2} \cdot n^6 \cdot d^6 \cdot \ln(2 \cdot \max(\alpha, 1) \cdot \max(\gamma, 1) \cdot d \cdot n \cdot m)}{\epsilon^5}.
\end{aligned}
$$

We want $k \geq k'$, therefore it suffices to bound from below k by the upper bound of k'. This completes the proof. □

7.4.2.4 Putting everything together

Proof. For the final step of the proof of Theorem 29, assume that $x_1^*, \ldots, x_n^* \in \mathcal{Y}$ is a solution for exact(F). Consider now a multivariate constraint $i \in [m]$ of F defined by $TMV_i(x_1, \ldots, x_n)$. Firstly, we replace this constraint by

$$|TMV_i(x_1, \ldots, x_n) - TMV_i(x_1^*, \ldots, x_n^*)| \leq \epsilon. \tag{7.13}$$

Then, replace constraint (7.13) by t constraints of the form

$$|STM_{i,j}(x_1, \ldots, x_n) - STM_{i,j}(x_1^*, \ldots, x_n^*)| \leq \frac{\epsilon}{t} \tag{7.14}$$

where $STM_{i,1}(x_1, \ldots, x_n), \ldots, STM_{i,t}(x_1, \ldots, x_n)$ are the simple tensor multivariate polynomials $TMV_i(x_1, \ldots, x_n)$ consists of. By the triangle inequality we get that if all t constraints given by (7.14) hold, then constraint (7.13) holds as well. Hence, we can reduce the problem to an equivalent problem with the same n variables and $m \cdot t$ constraints that all of them are simple tensor multivariate polynomials. So, we can apply Lemma 31 where we replace m with $m \cdot t$ and ϵ with $\frac{\epsilon}{t}$. This completes the proof of the theorem. □

7.5 Applications

We now show how our theorems can be applied to derive new approximation algorithms for a variety of problems. In order to conclude that Corollary 14 provides a PTAS or QPTAS for some given problem, one has to carefully determine the actual input size of the problem and show that the running time of the corollary's algorithm satisfies the PTAS or QPTAS definition.

7.5.1 Constrained approximate Nash equilibria

A *constrained* Nash equilibrium is a Nash equilibrium that satisfies some extra constraints, like specific bounds on the payoffs of the players. Constrained Nash equilibria attracted the attention of many authors, who proved NP-completeness for two-player games [24,48,76] and ETR-completeness for three-player games [24–27,75] for constrained *exact* Nash equilibria.

Constrained approximate equilibria have been studied, but so far only lower bounds have been derived [16,33,60,61,84]. It has been observed that sampling methods can give QPTASs for finding constrained approximate Nash equilibria for certain constraints in two player games [61].

By applying Theorem 29, we get the following result for games with number of players up to polylogarithmic in the number of pure strategies (here n is the number of players): *Any property of an approximate equilibrium that can be formulated in ϵ-ETR where α, γ, d, t and n are up to polylogarithmic in the number of pure strategies has a QPTAS.* This generalises past results to a much broader class of constraints, and provides results for games with more than two players, which had not previously been studied in this setting.

A game is defined by the set of players, the set of actions for every player, and the payoff function of every player. In normal form games, the payoff function is given by a multilinear function on a tensor of appropriate size. Consider an n-player game where every player has l-actions, and let A_j denote the payoff tensor of player j with elements in $[0, 1]$; A_j has size $\times_{i=1}^{n} l$. The interpretation of the tensor A_j is the following: the element $A_j(i_1, \ldots, i_n)$ of the tensor corresponds to the payoff of player j when Player 1 chooses action i_1, Player 1 chooses action i_2, and so on. To play the game, every player j chooses a probability distribution $x_j \in \Delta^l$, a.k.a. a *strategy*, over their actions. A collection of strategies is called *strategy profile*. The expected payoff of player j under the strategy profile $(x_1, \ldots, x_n)$ is given by $ML(A_j, x_1, \ldots, x_n)$. For notation simplicity, let $u_j(x_j, x_{-j}) := ML(A_j, x_1, \ldots, x_n)$, where

x_{-j} is the strategy profile of all players except player j. A strategy profile $(x_1^*, \ldots, x_n^*)$ is a Nash equilibrium if for every player j it holds that $u_j(x_j^*, x_{-j}^*) \geq u_j(x_j, x_{-j}^*)$ for every $x_j \in \Delta^l$, or equivalently $u_j(x_j^*, x_{-j}^*) \geq u_j(s_p, x_{-j}^*)$ for every possible s_p, where s_p denotes the case where player j chooses their action p with probability 1.

Our framework formally describes a broad family of constrained Nash equilibrium problems for which we can get a QPTAS.

Theorem 34. *Let Γ be an n-player l-action normal form game Γ. Furthermore, let F be a Boolean formula with $c \in poly(l)$ TMV constraints of degree d. If $n, d \in poly\log(l)$, then in quasi-polynomial time we can compute an approximate NE of Γ constrained by F, or decide that no such constrained approximate NE exists.*

Proof. Observe that we can write the problem of the existence of a constrained Nash equilibrium as an `ETR` problem. The constraints of the problem will be the constraints of F plus the constraint

$$u_j(s_l, x_{-j}) - u_j(x_j, x_{-j}) \leq 0$$

for every player $i \in [m]$ and every action s_l of player j.

Thus, we can use Theorem 29 and complete the proof since we produced an ϵ-ETR problem with $m = c + n \cdot l = \text{poly}(l)$ constraints, which is polynomial in the input size; d and t are polylogarithmic in l by assumption (it always holds that $t \leq d$); $\gamma = 1$ since every variable is a probability distribution; $\alpha = 1$ by the definition of normal form games. □

7.5.2 Shapley games

Shapley's stochastic games [129] describe a two-player infinite-duration zero-sum game. The game consists of N *states*. Each state specifies a two-player $M \times M$ bimatrix game where the players compete over: (1) a reward (which may be negative) that is paid by player two to player one, and (2) a probability distribution over the next state of the game. So each round consists of the players playing a bimatrix game at some state s, which generates a reward, and the next state s' of the game. The reward in round i is discounted by λ^{i-1}, where $0 < \lambda < 1$ is a *discount factor*. The overall payoff to player 1 is the discounted sum of the infinite sequence of rewards generated during the course of the game.

Shapley showed that these games are determined, meaning that there exists a value vector v, where v_s is the value of the game starting at state s. A polynomial-time algorithm has been devised for computing the value vector of a Shapley game when the number of

states N is constant [81]. However, since the values may be irrational, this algorithm needs to deal with algebraic numbers, and the *degree* of the polynomial is $O(N)^{N^2}$, so if N is even mildly super-constant, then the algorithm is not polynomial.

Furthermore, Shapley showed that the value vector is the unique solution of a system of polynomial optimality equations, which can be formulated in ETR. Any approximate solution of these equations gives an approximation of the value vector, and applying Theorem 29 gives us a QPTAS. This algorithm works when $N \in O(\sqrt[6]{\log M})$, which is a value of N that prior work cannot handle. The downside of our algorithm is that, since we require the solution to be bounded by a convex hull defined by finitely many points, the algorithm only works when the value vector is reasonably small. Specifically, the algorithm takes a constant bound $B \in \mathbb{R}$, and either finds the approximate value of the game, or verifies that the value is strictly greater than B.

To formally define a Shapley game, we use N to denote the number of states, and M to denote the number of actions. The game is defined by the following two functions.

- For each $s \leq N$ and $j, k \leq M$ the function $r(s, j, k)$ gives the reward at state s when player one chooses action j and player two chooses action k.

- For each $s, s' \leq N$ and $j, k \leq M$ the function $p(s, s', j, k)$ gives the probability of moving from state s to state s' when player one chooses action j and player two chooses action k. It is required that $\sum_{s'=1}^{N} p(s, s', j, k) = 1$ for all s, j, and k.

The game begins at a given starting state. In each round of the game the players are at a state s, and play the matrix game at that state by picking an action from the set $\{1, 2, \ldots, M\}$. The players are allowed to use randomization to make this choice. Supposing that the first player chose action j and the second player chose the action k, the first player receives the reward $r(s, j, k)$, and then a new state s' is chosen according to the probability distribution given by $p(s, \cdot, j, k)$.

The reward in future rounds is *discounted* by a factor of λ where $0 < \lambda < 1$ in each round. So if r_1, r_2, ... is the infinite sequence of rewards, the total reward paid by player two to player one is $\sum_{i=1}^{\infty} \lambda^{i-1} \cdot r_i$, which, due to the choice of λ, is always a finite value.

The two players play the game by specifying a probability distribution at each state, which represents their strategy for playing at that state. Let Δ^M denote the M-dimensional simplex, which represents the strategy space for both players at a single state. For each

$x, y \in \Delta^M$, we overload notation by defining the expected reward and next state functions.

$$r(s,x,y) = \sum_{j=1}^{M}\sum_{k=1}^{M} x(j)\cdot y(k)\cdot r(s,i,j),$$
$$p(s,s',x,y) = \sum_{j=1}^{M}\sum_{k=1}^{M} x(j)\cdot y(k)\cdot p(s,s',i,j).$$

Shapley showed that these games are *determined* [129], meaning that there is a unique vector $v \in \mathbb{R}^N$ such that v_s is the *value* of the game starting at state s: player one has a strategy to ensure that the expected reward is at least $v(s)$, while player two has a strategy to ensure that the expected reward is at most $v(s)$. Furthermore, Shapley showed that this value vector is the unique solution of the following *optimality equations* [129]. For each state s we have the equation

$$v(s) = \min_{x\in\Delta^M}\max_{y\in\Delta^M}\left(r(s,x,y) + \lambda\cdot\sum_{s'=1}^{N} p(s,s',x,y)\cdot v_{s'}\right). \tag{7.15}$$

In other words, v_s must be the value of the one-shot zero-sum game at s, where the payoffs of this zero-sum game are determined by the values of the other states given by $v_{s'}$.

Theorem 35. *Let Γ be a Shapley game with $N \in O(\sqrt[6]{\log M})$, unbounded number of actions per state, and rewards in $[-c,c]$ for every state-action combination, where c is a constant. Furthermore, let s be the starting state of the game. Let $B \in \mathbb{R}$ be a constant. In quasi-polynomial time we can approximately compute the value of Γ starting from s, if the value of every state is less than or equal to B, or decide that at least one of these values is greater than or equal to B.*

Proof. Let $v = (v(1), v(2), \ldots, v(N))$, and for every state s let x_s and y_s denote the strategy player one and player two choose at state s respectively. Observe that $r(s, x_s, y_s)$ is an STM polynomial with variables x and y of the form

$$\mathrm{STM}(A_{s1}, x_s, y_s) = \sum_{j=1}^{M}\sum_{k=1}^{M} x_s(j)\cdot y_s(k)\cdot a_{s1}(j,k)$$

where $a_{s1}(i,j,k) = r(s,j,k)$.

Observe also that $\lambda\cdot\sum_{s'=1}^{N} p(s,s',x_s,y_s)\cdot v_{s'}$ can be written as an STM polynomial

with variables x, y and v of the form

$$\mathrm{STM}(A_{s2}, x, y, v) = \sum_{j=1}^{M} \sum_{k=1}^{M} \sum_{l=1}^{N} x_s(j) \cdot y_s(k) \cdot v(l) \cdot a_{s2}(j,k,l)$$

where $a_{s2}(i,j,k) = \lambda \cdot p(s,l,j,k)$.

Let us define $\mathrm{TMV}_s(x_s, y_s, v) = \mathrm{STM}(A_{s1}, x_s, y_s) + \mathrm{STM}(A_{s2}, x_s, y_s, v)$; $\mathrm{TMV}_s(x_s, y_s, v)$ has length 2 and degree 1.

Note that we can replace equation (7.15) with the following $2 \cdot M$ TMV polynomial constraints

$$\begin{aligned} \mathrm{TMV}(x_s, y_s, v) - \mathrm{TMV}(j, y_s, v) \leq 0 \qquad & \text{for every action } j \leq M \text{ of player one} \\ \mathrm{TMV}(x_s, k, v) - \mathrm{TMV}(x_s, y_s, v) \geq 0 \qquad & \text{for every action } k \leq M \text{ of player two.} \end{aligned}$$

So, to approximate $v(s)$ it suffices to solve the ϵ-ETR problem defined by the $2 \cdot M \cdot N$ constraints defined as above for every state $s \leq N$. Observe, the ϵ-ETR problem has: $2N + 1$ variables (x_1 through x_N, y_1 through y_N, and v); $2 \cdot M \cdot N$ TMV constraints; $\gamma = \max\{1, \max_s v(s)\}$; $\alpha = \max\{c, \lambda \cdot \max_{s,s',j,k} p(s,s',j,k)\} = \max\{c, 1\}$, since $\lambda < 1$ and $\max_{s,s',j,k} p(s,s',j,k) < 1$. So, if $N \in O(\sqrt[6]{\log m})$, $\max_s v(s)$ is constant, and c is a constant, we can use Theorem 29 and derive a QPTAS for (7.15).

Finally, we note that an approximate solution to (7.15) gives an approximation of the value vector itself. This is because Shapley has shown that, when v is treated as a variable, the optimality equation given in (7.15) is a *contraction map*. The value vector is a fixed point of this contraction map, and the uniqueness of the value vector is guaranteed by Banach's fixed point theorem. Our algorithm produces an approximate fixed point of the optimality equations. It is easy to show, using the contraction map property, that an approximate fixed point must be close to an exact fixed point. □

7.5.3 Approximate consensus halving

In this section we show that an approximate solution to the consensus halving problem can be found in quasi-polynomial time when each agent's valuation function is a single polynomial of constant or even polylogarithmic degree. We will do so by formulating the problem as a constrained ϵ-ETR instance, and then applying Theorem 29.

This result first appeared in [58,59] and implies that these instances can be solved

approximately using a polylogarithmic number of cuts. We note that this is one of the most general classes of instances for which we could hope to prove such a result: any instance in which n agents desire completely disjoint portions of the object can only be solved by an n-cut, and piecewise linear functions are capable of producing such a situation. So in a sense, we are exploiting the fact that this situation cannot arise when the agents have non-piecewise polynomial valuation functions.

Lemma 32. *For every* Consensus Halving *instance with n agents, and every $\epsilon > 0$, if each agent's valuation function F_i is a single polynomial of degree at most $O(poly \log n)$, then there exists a k-cut, where $k := O(poly \log n)/\epsilon^5$, and pieces A_+ and A_- such that:*

- *every cut point is a multiple of $1/k = \frac{\epsilon^5}{O(poly \log n)}$;*
- *$|F_i(A_+) - F_i(A_-)| \leq \epsilon$, for every agent i.*

Proof. Since each agent i has a polynomial valuation function, there is a $d \in O(\log n)$ and constants $a_0, a_1, \ldots, a_l$ such that each function F_i can be written as $F_i(t) = \sum_{j=0}^{d} a_j \cdot t^j$.

To prove the theorem, we will formulate the problem as a constrained ϵ-ETR instance, and apply Theorem 29, which proves the claim. We first write a simple ETR formula for consensus halving with polynomial valuation functions. If a consensus halving instance has a solution, then it also has one in which the cuts are *strictly alternating*, meaning that

$$F_i(A_+) = \sum_{j=1}^{\lfloor n/2 \rfloor} \big(F_i(t_{2j}) - F_i(t_{2j-1})\big),$$
$$F_i(A_-) = \sum_{j=1}^{\lceil n/2 \rceil} \big(F_i(t_{2j-1}) - F_i(t_{2j-2})\big),$$

where the cut is the tuple $(t_1, t_2, \ldots, t_n)$, with $0 \leq t_1 \leq \cdots \leq t_n \leq 1$ and $t_0 := 0$, $t_{n+1} := 1$.

In this encoding, we have no need to encode which set a particular cut belongs to, and so we can encode a n-cut as an element of the n-simplex $x := (x_1, x_2, \ldots, x_{n+1}) \in \Delta^{n+1}$, where $x_i := t_i - t_{i-1}$. From the latter, it is easy to see that

$$t_i := \sum_{j=1}^{i} x_j. \tag{7.16}$$

For $j \in \{0, 1, \ldots, n\}$, let us denote by 1^j and 0^j a j-tuple of 1's and 0's respectively. Let us also define the n-dimensional vector $v_j := (0^j, 1^{n-j})$. Now observe that any n-cut

$t := (t_1, t_2, \ldots, t_n)$ can be represented by a n-dimensional point which is in fact a convex combination of the $n+1$ vectors v_j, $j \in \{0, 1, \ldots, n\}$. In particular, from (7.16) it is easy to see that

$$t := (t_1, t_2, \ldots, t_n) = \sum_{j=1}^{n+1} x_j \cdot v_{j-1}.$$

Hence, we can encode the problem as an ETR formula

$$\exists t \cdot \left(\bigwedge_{i=1}^{n} F_i(A_+) = F_i(A_-) \right) \wedge t \in C,$$

where C is the convex hull of the vectors $v_0, v_1, \ldots, v_n$. This formula has n constraints, one for each agent, and a single constraint bounding the variables in the convex set C which can be expressed by $n+1$ vectors, namely v_j, $j \in \{0, 1, \ldots, n\}$.

The main theorem of [57] allows us to leave the constraint $t \in C$ unchanged, but insists that we weaken the others. Specifically each constraint is weakened so that only $F_i(A_+) - F_i(A_-) \leq \epsilon$ and $F_i(A_+) - F_i(A_-) \geq -\epsilon$ are enforced, which implies that $|F_i(A_+) - F_i(A_-)| \leq \epsilon$. This is sufficient to encode an approximate solution to the problem.

The constructed ϵ-ETR instance has one vector-variable $t \in C$ and $2n$ constraints. Let us now study one of the constraints of the ϵ-ETR instance.

$$\sum_{j=1}^{\lfloor n/2 \rfloor} \big(F_i(t_{2j}) - F_i(t_{2j-1})\big) - \sum_{j=1}^{\lceil n/2 \rceil} \big(F_i(t_{2j-1}) - F_i(t_{2j-2})\big) \leq \epsilon.$$

Using the representation of F_i, we can write down a constraint as $\sum_{k=0}^{d} a_k \cdot h_k(t_1, t_2, \ldots, t_n) \leq \epsilon$, where $h_k(t_1, t_2, \ldots, t_n)$ is a sum of monomials, each one of degree d. F_i depends on t_0 and t_{n+1} as well, but recall that these are 0 and 1 respectively.

The term $a_k \cdot h_k(t_1, t_2, \ldots, t_n)$ is a simple tensor multivariate polynomial with one variable of degree k, which we will denote by $STM(H_k, t^k)$. Under this notation H_k is a k-dimensional tensor where vector t is applied k times. Hence, every constraint is a sum of $d+1$ simple tensor multivariate polynomials, i.e. a TMV polynomial of maximum degree d constructed by $d+1$ STM polynomials. Furthermore, $||v_j||_\infty \leq 1$ for all $j \in \{0, 1, \ldots, n\}$, and for every constraint, the maximum absolute coefficient is constant by definition, and the degree d is $O(poly \log n)$. Hence, we can apply Theorem 29 and get the claimed result. □

As a consequence, we can perform a brute force search over all possible k-cuts to find an approximate solution, which can be carried out in $n^{O(poly \log n/\epsilon^5)}$ time.

Theorem 36. Consensus Halving *admits a QPTAS when the valuation function of every agent is a single polynomial of degree* $O(poly \log n)$.

7.5.4 Optimization problems

Our framework can provide approximation schemes for optimization problems with one vector variable $x \in \mathbb{R}^p$ with polynomial constraints over bounded convex sets. Formally,

$$\begin{aligned} \max \quad & h(x) \\ s.t. \quad & h_1(x) \geq 0, \ldots, h_m(x) \geq 0 \\ & x \in \text{conv}(c_1, \ldots, c_l) \end{aligned}$$

where $h(x), h_1(x), \ldots, h_m(x)$ are polynomials with respect to vector x; for example $h(x) = x^T Ax$, where A is an $p \times p$ matrix, subject to $h_1(x) = x^T x - \frac{1}{10} \geq 0$ and $x \in \Delta^p$. We will call the polynomials h_i *solution-constraints*. Optimization problems of this kind received a lot of attention over the years [53,55,56,68].

For optimization problems, we sample from the solution that achieves the maximum when we apply Theorem 29, in order to prove that there is a k-uniform solution that is close to the maximum. Our algorithm enumerates all k-uniform profiles, and outputs the one that maximizes the objective function. Using this technique, Theorem 29 gives the following results.

1. There is a PTAS if $h(x)$ is a STM polynomial of maximum degree independent of p, the number of solution-constraints is independent of p, and $l = \text{poly}(p)$.

2. There is QPTAS if $h(x)$ is a STM polynomial of maximum degree up to $\text{poly} \log p$, the number of solution-constraints is $\text{poly}(p)$, and $l = \text{poly}(p)$.

To the best of our knowledge, the second result is new. The first result was already known, however it was proven using completely different techniques: in [29] it was proven for the special case of degree two, in [68] it was extended to any fixed degree, and alternative proofs of the fixed degree case were also given in [55,56]. We highlight that in all of the aforementioned results solution constraints were not allowed. Note that unless NP = ZPP

there is no FPTAS for quadratic programming even when the variables are constrained in the simplex [53]. Hence, our results can be seen as a partial answer to the important question posed in [53]: *What is a complete classification of functions that allow a PTAS?*

Furthermore, as shown in Theorem 30 this technique yields a generalized algorithm for multi-objective optimization problems which, to the best of our knowledge, is a completely new result.

7.5.5 Tensor problems

Our framework provides quasi-polynomial time algorithms for deciding the existence of approximate eigenvalues and approximate eigenvectors of tensors in $\mathbb{R}^{p\times p\times p}$, where the elements are bounded by a constant, where the solutions are required to be in a bounded convex set. In [85] it is proven that there is no PTAS for these problems when the domain is unrestricted. To the best of our knowledge this is the first positive result for the problem even in this, restricted, setting.

Definition 27. *The nonzero vector $x \in \mathbb{R}^p$ is an* eigenvector *of tensor $A \in \mathbb{R}^{p\times p\times p}$ if there exists an* eigenvalue *$\lambda \in \mathbb{R}$ such that for every $k \in [p]$ it holds that*

$$\sum_{i}^{n}\sum_{j}^{n} a(i,j,k)\cdot x(i)\cdot x(j) = \lambda \cdot x(k). \tag{7.17}$$

Theorem 37. *Let A be an $\mathbb{R}^{p\times p\times p}$ tensor with entries in $[-c,c]$, where c is a constant. Furthermore, let $B \in \mathbb{R}$ be a constant and let $\mathcal{Y}$ be a bounded convex set where $\|\mathcal{Y}\|_\infty$ is a constant. In a quasi-polynomial time we can compute an eigenvalue-eigenvector pair (λ, x) that approximately satisfy* (7.17) *such that $\lambda \leq B$ and $x \in \mathcal{Y}$, or decide that no such pair exists.*

Proof. Observe that $\sum_i^n \sum_j^n a(i,j,k)\cdot x(i)\cdot x(j)$ can be written as an STM polynomial $\mathrm{STM}(A_1, x^2)$ where $a_1(i,j) = a(i,j,k)$. Furthermore, let ℓ be a p-dimensional vector. Then, $\lambda \cdot x(k)$ can be written as an STM polynomial $\mathrm{STM}(A_2, x, \ell)$, where $a_2(k,1) = 1$ and zero otherwise.

So, Equation (7.17) can be written as an TMV polynomial constraint of degree 2 and length 2, with two vector variables, x and ℓ. So, the problem of computing an eigenvalue-eigenvector pair that approximately satisfy (7.17) can be written as an ϵ-ETR problem with p TMV polynomial constraints of degree 2 and length 2 and two vector variables. Hence,

we can use Theorem 29 with $\gamma = \|\mathcal{Y}\|_\infty$ which is constant, $\alpha = c$, $n = 2$, $t = 2$, $d = 2$, and $m = p$ to find a solution if exists, or decide that no such solution exists. □

7.5.6 Computational geometry

Finally, we note that our theorem can be applied to problems in computational geometry, although the results are not as general as one may hope. Many problems in this field are known to be ETR-complete, including, for example, the Steinitz problem for 4-polytopes, inscribed polytopes and Delaunay triangulations, polyhedral complexes, segment intersection graphs, disk intersection graphs, dot product graphs, linkages, unit distance graphs, point visibility graphs, rectilinear crossing number, and simultaneous graph embeddings. We refer the reader to the survey of Cardinal [38] for further details.

All of these problems can be formulated in ϵ-ETR, and indeed our theorem does give results for these problems. However, our requirement that the bounding convex set be given explicitly limits their applicability. Most computational geometry problems are naturally constrained by a cube, so while Corollary 15 does give NP algorithms, we do not get QPTASs unless we further restrict the convex set. Here we formulate QPTASs for the segment intersection graph and the unit disk intersection graph problems when the solutions are restricted to lie in a simplex. While it is not clear that either problem has natural applications that are restricted in this way, we do think that future work may be able to derive sampling theorems that are more tailored towards the computational geometry setting.

7.5.6.1 Segment intersection graphs

Definitions Let G be an undirected graph with vertex set $\{v_1, v_2, \ldots, v_n\}$. We say that G is a *segment graph* if there are straight segments $s_1, s_2, \ldots, s_n$ in the plane such that, for every $i, j, 1 \leq i < j \leq n$, the segments s_i and s_j have a common point if and only if $\{v_i, v_j\} \in E(G)$.

By a suitable rotation of the co-ordinate system we can achieve that none of the segments is vertical. Then the segment s_i representing vertex v_i can be algebraically described as the set $\{(x, y) \in \mathbb{R}^2 : y = a_i x + b_i, c_i \leq x \leq d_i\}$ for some real numbers a_i, b_i, c_i, d_i. We say that G is a *simplex K segment graph* if the real numbers a_i, b_i, c_i, d_i, $i = 1, 2, \ldots n$ are under the

constraints

$$a_i, b_i, c_i, d_i \geq 0, \text{ for every } i = 1, 2, \ldots n, \text{ and}$$
$$\sum_{i=1}^{n}(a_i + b_i + c_i + d_i) = K, \quad \text{where } K > 0 \text{ is a given constant.}$$

We let SIM-K-SEG denote the class of all simplex K segment graphs with parameter $K > 0$.

The problem ϵ-RECOG(SIM-K-SEG) is defined as follows. Given an abstract undirected graph G, does it belong with tolerance ϵ to SIM-K-SEG?

Formulation of ϵ-RECOG(SIM-K-SEG) We first give a description for the problem with $\epsilon = 0$ and then we generalize for arbitrary $\epsilon \geq 0$. The formulation is taken from [99].

Letting l_i be the line containing s_i, we note that $s_i \cap s_j \neq \emptyset$ if l_i and l_j intersect in a single point whose x-coordinate lies in both the intervals $[c_i, d_i]$ and $[c_j, d_j]$. It is easy to see that the x-coordinate equals $\frac{b_j - b_i}{a_i - a_j}$.

Now we turn to the general case where $\epsilon \geq 0$. Let us introduce variables A_i, B_i, C_i, D_i representing the unknown quantities a_i, b_i, c_i, d_i, $i = 1, 2, \ldots, n$. By the problem's definition we require the vector $(A_1, B_1, C_1, D_1, \ldots, A_n, B_n, C_n, D_n)$ to be in the $(4n-1)$-simplex with parameter K. Then $s_i \cap s_j \neq \emptyset$ can be expressed by the following predicate:

$$\begin{aligned}\text{INTS}(A_i, B_i, C_i, D_i, A_j, B_j, C_j, D_j) = \\ (A_i >_\epsilon A_j \wedge C_i(A_i - A_j) \leq_\epsilon B_j - B_i \leq_\epsilon D_i(A_i - A_j) \\ \wedge C_j(A_i - A_j) \leq_\epsilon B_j - B_i \leq_\epsilon D_j(A_i - A_j)) \\ \vee (A_i <_\epsilon A_j \wedge C_i(A_i - A_j) \geq_\epsilon B_j - B_i \geq_\epsilon D_i(A_i - A_j) \\ \wedge C_j(A_i - A_j) \geq_\epsilon B_j - B_i \geq_\epsilon D_j(A_i - A_j))\end{aligned}$$

(this is only correct if we "globally" assume that $C_i \leq_\epsilon D_i$ for all i). The existence of a

SEG-representation of G can then be expressed by the formula

$$(\exists A_1B_1C_1D_1 \dots A_nB_nC_nD_nK) \left(\bigwedge_{i=1}^{n} C_i \leq_\epsilon D_i \right)$$
$$\wedge \left(\bigwedge_{\{i,j\}\in E} \text{INTS}(A_i, B_i, C_i, D_i, A_j, B_j, C_j, D_j) \right)$$
$$\wedge \left(\bigwedge_{\{i,j\}\notin E} \neg\text{INTS}(A_i, B_i, C_i, D_i, A_j, B_j, C_j, D_j) \right)$$

Theorem 38. *There is an algorithm that runs in time $n^{O(K^2 \cdot \log n/\epsilon^2)}$ and either finds a vector $(A_1, B_1, C_1, D_1, \dots, A_n, B_n, C_n, D_n)$ that is a solution to ϵ-RECOG(SIM-K-SEG), or determines that there is no solution to 0-RECOG(SIM-K-SEG).*

Proof. We set $x = (A_1, B_1, C_1, D_1, \dots, A_n, B_n, C_n, D_n)$ and $F(x)$ to be the above formula that we constructed. Their combination makes an ϵ-ETR instance. Vector x is constrained over the convex hull defined by the vertices of the $(4n-1)$-simplex, i.e. vectors $v_i \in \mathbb{R}^{4n}$, $i \in \{1, 2, \dots 4n\}$ with their i-th element equal to K and the rest equal to 0. Therefore the cardinality of our convex set is $m = 4n$, and $\gamma = K$. By looking at the formula we can conclude that $a = 1$, $t = 4$, and $d = 2$. By Theorem 32 the result follows. □

7.5.6.2 Unit disk intersection graphs

Definitions Let G be an undirected graph with vertex set $\{v_1, v_2, \dots, v_n\}$. We say that G is a *unit disk intersection graph* or *unit disk graph* if there are disks $d_1, d_2, \dots, d_n$ (in the plane) with radius 1 such that, for every $i, j, 1 \leq i < j \leq n$, the disks d_i and d_j have more than one points common (i.e. an area) if and only if $\{v_i, v_j\} \in E(G)$.

The disk d_i representing vertex v_i can be algebraically described as the set $\{(x, y) \in \mathbb{R}^2 : (x - x_i)^2 + (y - y_i)^2 \leq 1\}$ for some real numbers x_i, y_i that determine the centre of the disk. We say that G is a *simplex K unit disk graph* if the real numbers x_i, y_i, $i = 1, 2, \dots n$ are under the constraints

$$x_i, y_i \geq 0, \text{ for every } i = 1, 2, \dots n, \text{ and}$$
$$\sum_{i=1}^{n} (x_i + y_i) = K, \quad \text{where } K > 0 \text{ is a given constant.}$$

We let SIM-K-UDG denote the class of all simplex K unit disk graphs with parameter $K > 0$.

The problem ϵ-RECOG(SIM-K-UDG) is defined as follows. Given an abstract undirected graph G, does it belong with tolerance ϵ to SIM-K-UDG?

Formulation of ϵ-RECOG(SIM-K-UDG) Let us introduce variables X_i, Y_i representing the unknown quantities x_i, y_i, $i = 1, 2, \ldots, n$. We require the vector $(X_1, Y_1, \ldots, X_n, Y_n)$ to be in the $(2n-1)$-simplex with parameter K. Then we consider an ϵ-intersection $d_i \cap_\epsilon d_j \neq \emptyset$ to happen if:

$$\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} < 2 + \epsilon$$

and an ϵ-non-intersection $d_i \cap_\epsilon d_j = \emptyset$ to happen if:

$$\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \geq 2 - \epsilon$$

The existence of a UDG-representation of G can then be expressed by the formula

$$\begin{aligned}(\exists X_1 Y_1 \ldots X_n Y_n)& \\ &\left(\bigwedge_{\{i,j\} \in E} (X_i - X_j) \cdot (X_i - X_j) + (Y_i - Y_j) \cdot (Y_i - Y_j) < 4 + 2\epsilon + \epsilon^2 \right) \\ \wedge &\left(\bigwedge_{\{i,j\} \notin E} (X_i - X_j) \cdot (X_i - X_j) + (Y_i - Y_j) \cdot (Y_i - Y_j) \geq 4 - 2\epsilon + \epsilon^2 \right)\end{aligned}$$

Theorem 39. *There is an algorithm that runs in time $n^{O(K^2 \cdot \log n/\epsilon^2)}$ and either finds a vector $(X_1, Y_1, \ldots, X_n, Y_n)$ that is a solution to ϵ-RECOG(SIM-K-UDG), or determines that there is no solution to 0-RECOG(SIM-K-UDG).*

Proof. We set $x = (X_1, Y_1, \ldots, X_n, Y_n)$ and $F(x)$ to be the above formula that we constructed. Their combination makes an ϵ-ETR instance. Vector x is constrained over the convex set defined by the vertices of the $(2n-1)$-simplex, i.e. vectors $v_i \in \mathbb{R}^{2n}$, $i \in \{1, 2, \ldots 2n\}$ with their i-th element equal to K and the rest equal to 0. Therefore the cardinality of our convex set is $m = 2n$, and $\gamma = K$. By looking at the formula we can conclude that $a = 2$, $t = 7$, and $d = 2$. By Theorem 32 the result follows. □

Chapter 8

Conclusions

In this thesis we extend results regarding computational complexity and efficient algorithms for problems in various strategic settings. Here we give an overview of how our results relate to previous work and present interesting questions that remain to be answered.

8.1 Evolutionary Games

Our results extend the work of Etessami and Lochbihler [67] which showed that deciding the existence of an ESS in a given game is coNP-hard. In particular, their hardness reduction is from the complement of the Clique problem to a particular instance of a game. We extend this reduction to be valid from the same coNP-complete problem to a family of games whose values can be arbitrarily picked within specific intervals. This reveals that infinitely many instances of the problem are coNP-hard, and that arbitrary perturbations of payoff values around those of the hard instances do not suffice to make the problem easier.

On the other hand, the work of Hart and Rinott [127] implies that, for almost all games whose payoffs are sampled from common probability distributions, not only it is easy to decide existence of ESS, but the answer is "yes". This indicates that there should be some instance "between" the ones we find hard, and the ones of [127] for which the problem becomes easier to decide, perhaps in polynomial time. In view of this, an interesting open problem is to categorize families of games according to the complexity of the ESS existence problem.

8.2 Games between Rational and Intelligent Entities

8.2.1 Strategic contention resolution

We have extended the results of Christodoulou et al. [43] and Fiat et al. [69] for the acknowledgement-based and ternary feedback, respectively. These works provide equilibrium protocols with desirable properties of anonymity and time-efficiency for the case of a single channel. In this thesis we studied the same setting for multiple channels and showed that there are more, and easier to find equilibrium protocols with the same properties.

However, this work leaves open some interesting problems. One of them is to find equilibria for arbitrary number of players in the multiple-channel setting with acknowledgement-based feedback for the general, history-dependent case or the special, history-independent case. This will probably require a characterization of equilibria which will also give a great amount of information about how the equilibria look like, similarly to how the characterization we provide for history-independent, ternary feedback protocols in Theorem 10 indicates the exact (asymptotic) behaviour of the transmission probability and the expected latency.

Another important open problem is to prove or disprove that there exists a FIN-EQ protocol that is efficient in the multiple-channels setting. This could be a deadline protocol or it might use some other key idea to impose a heavy latency on the players as a threat, so that they auto-restrain themselves from frequently attempting transmission. Proving that there is no efficient deadline FIN-EQ for the multiple-channel setting would be an interesting "paradox", since an efficient deadline FIN-EQ is found in [69] for the single-channel setting with ternary feedback. In view of Theorem 11 we conjecture that the "paradox" is there.

8.2.2 Connected Subgraph Defense Games

Here our results extend the line of work of Mavronicolas et al. [100] on defense games in graphs. In these games, we have generalized the pure strategy of the defender to be a connected induced subgraph of the underlying graph of size λ instead of two adjacent nodes. We termed these new games Connected Subgraph Defense (CSD) games and studied the structure of equilibria and the complexity of finding one, depending on the power of the defender λ. We also extended the notion of *Price of Defense*, as termed in [100], for any λ and found almost tight bounds for its value.

An interesting open problem is the following. For λ that is both more than constantly away from 1 and n, our LP-based algorithm for computing a Nash equilibrium is not ef-

ficient. That is because in that case, our algorithm considers the strategy space of the defender to have cardinality $\binom{n}{\lambda}$ and brute forces through all of that space. Is there a polynomial time algorithm for computing a Nash equilibrium when $\lambda \in \omega(1) \cap o(n)$? Another open problem is to determine the complexity of deciding whether a general graph is defense-optimal. Here we conjecture that it is NP-hard.

8.3 Fair Division

We studied the "exact" counterpart of the recent work [71,72] which has shown that the approximate version of the Consensus Halving problem is PPA-complete. We proved that the exact version is much harder (under standard complexity assumptions), namely FIXP-hard, and deciding whether there exists a solution with fewer than n cuts is ETR-complete. En route we defined a new complexity class BU, which captures the search problems whose solution is proven via the Borsuk-Ulam theorem. We also showed that FIXP $\subseteq$ BU $\subseteq$ TFETR and that LinearBU = PPA, where LinearBU is the subclass of BU whose input can be expressed by a linear arithmetic circuit. The latter is analogous to the result that LinearFIXP = PPAD by Etessami and Yannakakis in [66] which established a relation between the class FIXP of exact solutions via Brouwer's fixed point theorem and the class PPAD of solutions guaranteed to exist by the parity argument on directed graphs.

The main open problem of this work is to find matching upper and lower complexity bounds. We believe that the true complexity of Consensus Halving is BU-completeness. Such a result would make Consensus Halving the first natural problem that characterizes BU, and establish BU as a distinct complexity class between FIXP and TFETR.

8.4 Approximation Algorithms

In the last set of results, we extended the Lipton-Markakis-Mehta (LMM) algorithm for computing ϵ-Nash equilibria, and employed it in order to derive approximation schemes for a wide subclass of ETR. For a given constrained ϵ-ETR instance whose variables' domain is the convex hull of l vectors, we presented an algorithm which runs in time $l^{O(k)}$, for k indicated in Theorem 29, that either computes a solution or respond that a solution to the exact problem does not exist. This algorithm is a QPTAS or PTAS for many well-known problems. However, our algorithm, being an extension of the LMM algorithm, for some problems does not have better running time than the state of the art algorithms that are

tailored to these problems. The most important open problem is to make the quantity k depend logarithmically on crucial parameters, such as the number of variables n and the degree of the polynomials d, instead of polynomially. This would generalize many algorithms, such as the PTAS for computing an ϵ-Nash equilibrium in anonymous games [51] and the best algorithm for computing an ϵ-Nash equilibrium in general multi-player normal form games [19].

Bibliography

[1] Zachary Abel, Erik D Demaine, Martin L Demaine, Sarah Eisenstat, Jayson Lynch, and Tao B Schardl. Who needs crossings? Hardness of plane graph rigidity. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 51. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[2] Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is ETR-complete. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 65–73. ACM, 2018.

[3] James Aisenberg, Maria Luisa Bonet, and Sam Buss. 2-d Tucker is PPA complete. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:163, 2015.

[4] Eleni C. Akrida, Argyrios Deligkas, Themistoklis Melissourgos, and Paul G. Spirakis. Connected subgraph defense games. In *Algorithmic Game Theory - 12th International Symposium, SAGT 2019, Athens, Greece, September 30 - October 3, 2019, Proceedings*, pages 216–236, 2019.

[5] N. Alon, R. M. Karp, D. Peleg, and D. B. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

[6] Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987.

[7] Noga Alon and Douglas B West. The Borsuk-Ulam theorem and bisection of necklaces. *Proceedings of the American Mathematical Society*, 98(4):623–628, 1986.

[8] Ingo Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199:339 – 355, 1994. Special Issue Honoring Ingram Olkin.

[9] Eitan Altman, Dhiman Barman, Abderrahim Benslimane, and Rachid El Azouzi. Slotted aloha with priorities and random power. In *International Conference on Research in Networking*, pages 610–622. Springer, 2005.

[10] Eitan Altman, Rachid El Azouzi, and Tania Jiménez. Slotted aloha as a game with partial information. *Computer networks*, 45(6):701–713, 2004.

[11] Georgios Amanatidis, George Christodoulou, John Fearnley, Evangelos Markakis, Christos-Alexandros Psomas, and Eftychia Vakaliou. An improved envy-free cake cutting protocol for four agents. In *International Symposium on Algorithmic Game Theory*, pages 87–99. Springer, 2018.

[12] Bo An, James Pita, Eric Shieh, Milind Tambe, Chris Kiekintveld, and Janusz Marecki. Guards and protect: Next generation applications of security games. *ACM SIGecom Exchanges*, 10(1):31–34, 2011.

[13] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[14] J. Aspnes, K. L. Chang, and A. Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. *J. Comput. Syst. Sci.*, 72(6):1077–1093, 2006.

[15] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, Inc., 2004.

[16] Per Austrin, Mark Braverman, and Eden Chlamtac. Inapproximability of NP-complete variants of Nash equilibrium. *Theory of Computing*, 9:117–142, 2013.

[17] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–427. IEEE, 2016.

[18] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 454–464. ACM, 2016.

[19] Yakov Babichenko, Siddharth Barman, and Ron Peretz. Empirical distribution of equilibrium play and its testing application. *Mathematics of Operations Research*, 42(1):15–29, 2016.

[20] Siddharth Barman. Approximating Nash equilibria and dense bipartite subgraphs via an approximate version of Caratheodory's theorem. In *Proc. of STOC*, pages 361–369, 2015.

[21] Siddharth Barman, Katrina Ligett, and Georgios Piliouras. Approximating Nash equilibria in tree polymatrix games. In *Proc. of SAGT*, pages 285–296. Springer, 2015.

[22] Michael A Bender, Martin Farach-Colton, Simai He, Bradley C Kuszmaul, and Charles E Leiserson. Adversarial contention resolution for simple channels. In *Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 325–332. ACM, 2005.

[23] Daniel Bienstock. Some provably hard crossing number problems. *Discrete & Computational Geometry*, 6(3):443–459, 1991.

[24] Vittorio Bilò and Marios Mavronicolas. The complexity of decision problems about Nash equilibria in win-lose games. In *Proc. of SAGT*, pages 37–48, 2012.

[25] Vittorio Bilò and Marios Mavronicolas. Complexity of rational and irrational Nash equilibria. *Theory of Computing Systems*, 54(3):491–527, 2014.

[26] Vittorio Bilò and Marios Mavronicolas. A catalog of EXISTS-R-complete decision problems about nash equilibria in multi-player games. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, pages 17:1–17:13, 2016.

[27] Vittorio Bilò and Marios Mavronicolas. Existential-R-complete decision problems about symmetric Nash equilibria in symmetric multi-player games. In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, pages 13:1–13:14, 2017.

[28] Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society (N.S.)*, 21(1):1–46, 07 1989.

[29] Immanuel M Bomze and Etienne De Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2):163–185, 2002.

[30] Steven J Brams and D Marc Kilgour. Competitive fair division. *Journal of Political Economy*, 109(2):418–443, 2001.

[31] Steven J Brams and Alan D Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

[32] Steven J Brams and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[33] Mark Braverman, Young Kun-Ko, and Omri Weinstein. Approximating the best Nash equilibrium in $n^{o(\log\ n)}$-time breaks the exponential time hypothesis. In *Proc. of SODA*, pages 970–982, 2015.

[34] L. E. J. Brouwer. Über abbildung von mannigfaltigkeiten. *Mathematische Annalen*, 71(1):97–115, Mar 1911.

[35] John Canny. Some algebraic and geometric computations in PSPACE. In *Proc. of STOC*, pages 460–467, New York, NY, USA, 1988. ACM.

[36] John Capetanakis. Generalized tdma: The multi-accessing tree protocol. *IEEE Transactions on Communications*, 27(10):1476–1484, 1979.

[37] John Capetanakis. Tree algorithms for packet broadcast channels. *IEEE transactions on information theory*, 25(5):505–515, 1979.

[38] Jean Cardinal. Computational geometry column 62. *ACM SIGACT News*, 46(4):69–78, 2015.

[39] Jean Cardinal and Udo Hoffmann. Recognition and complexity of point visibility graphs. *Discrete & Computational Geometry*, 57(1):164–178, 2017.

[40] Jenhui Chen, Shiann-Tsong Sheu, and Chin-An Yang. A new multichannel access protocol for ieee 802.11 ad hoc wireless lans. In *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, volume 3, pages 2291–2296. IEEE, 2003.

[41] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)*, 56(3):14, 2009.

[42] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2 edition, 2003.

[43] George Christodoulou, Martin Gairing, Sotiris E. Nikoletseas, Christoforos Raptopoulos, and Paul G. Spirakis. Strategic contention resolution with limited feedback. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, pages 30:1–30:16, 2016.

[44] George Christodoulou, Martin Gairing, Sotiris E. Nikoletseas, Christoforos Raptopoulos, and Paul G. Spirakis. A 3-player protocol preventing persistence in strategic contention with limited feedback. In *Algorithmic Game Theory - 10th International Symposium, SAGT 2017, L'Aquila, Italy, September 12-14, 2017, Proceedings*, pages 240–251, 2017.

[45] George Christodoulou, Katrina Ligett, and Evangelia Pyrga. Contention resolution under selfishness. *Algorithmica*, 70(4):675–693, 2014.

[46] George Christodoulou, Themistoklis Melissourgos, and Paul G. Spirakis. Short paper: Strategic contention resolution in multiple channels with limited feedback. In *Algorithmic Game Theory - 11th International Symposium, SAGT 2018, Beijing, China, September 11-14, 2018, Proceedings*, pages 245–250, 2018.

[47] George Christodoulou, Themistoklis Melissourgos, and Paul G. Spirakis. Strategic contention resolution in multiple channels. In *Approximation and Online Algorithms - 16th International Workshop, WAOA 2018, Helsinki, Finland, August 23-24, 2018, Revised Selected Papers*, pages 165–180, 2018.

[48] V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621 – 641, 2008.

[49] Vincent Conitzer. The exact computational complexity of evolutionarily stable strategies. In *Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings*, pages 96–108, 2013.

[50] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009.

[51] Constantinos Daskalakis and Christos H. Papadimitriou. On oblivious ptas's for nash equilibrium. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 75–84, 2009.

[52] Etienne de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *CEJOR*, 16(2):111–125, 2008.

[53] Etienne De Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.

[54] Etienne De Klerk, Monique Laurent, and Pablo A Parrilo. A PTAS for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, 361(2-3):210–225, 2006.

[55] Etienne De Klerk, Monique Laurent, and Pablo A Parrilo. A PTAS for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, 361(2-3):210–225, 2006.

[56] Etienne de Klerk, Monique Laurent, and Zhao Sun. An alternative proof of a PTAS for fixed-degree polynomial optimization over the simplex. *Mathematical Programming*, 151(2):433–457, 2015.

[57] Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Approximating the Existential Theory of the Reals. In George Christodoulou and Tobias Harks, editors, *Web and Internet Economics*, pages 126–139, Cham, 2018. Springer International Publishing.

[58] Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Computing exact solutions of Consensus Halving and the Borsuk-Ulam theorem. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece.*, pages 138:1–138:14, 2019.

[59] Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Computing exact solutions of Consensus Halving and the Borsuk-Ulam theorem. *CoRR*, abs/1903.03101, 2019.

[60] Argyrios Deligkas, John Fearnley, and Rahul Savani. Computing constrained approximate equilibria in polymatrix games. In *Proc. of SAGT*, pages 93–105, 2017.

[61] Argyrios Deligkas, John Fearnley, and Rahul Savani. Inapproximability results for constrained approximate Nash equilibria. *Information and Computation*, 2018.

[62] Argyrios Deligkas, John Fearnley, and Paul Spirakis. Lipschitz continuity and approximate equilibria. In *Proc. of SAGT*, pages 15–26. Springer, 2016.

[63] Xiaotie Deng, Jack R Edmonds, Zhe Feng, Zhengyang Liu, Qi Qi, and Zeying Xu. Understanding PPA-completeness. In *Proceedings of the 31st Conference on Computational Complexity*, page 23. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[64] Xiaotie Deng, Zhe Feng, and Rucha Kulkarni. Octahedral Tucker is PPA-complete. In *Electronic Colloquium on Computational Complexity Report TR17-118*, 2017.

[65] Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *The American mathematical monthly*, 106(10):930–942, 1999.

[66] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.

[67] Kousha Etessami and Andreas Lochbihler. The computational complexity of evolutionarily stable strategies. *Int. J. Game Theory*, 37(1):93–113, 2008.

[68] Leonid Faybusovich. Global optimization of homogeneous polynomials on the simplex and on the sphere. In *Frontiers in global optimization*, pages 109–121. Springer, 2004.

[69] Amos Fiat, Yishay Mansour, and Uri Nadav. Efficient contention resolution protocols for selfish agents. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 179–188, 2007.

[70] Aris Filos-Ratsikas, Søren Kristoffer Stiil Frederiksen, Paul W. Goldberg, and Jie Zhang. Hardness results for consensus-halving. In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, pages 24:1–24:16, 2018.

[71] Aris Filos-Ratsikas and Paul W. Goldberg. Consensus halving is PPA-complete. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 51–64, 2018.

[72] Aris Filos-Ratsikas and Paul W. Goldberg. The complexity of splitting necklaces and bisecting ham sandwiches. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 638–649, New York, NY, USA, 2019. ACM.

[73] M. K. Franklin, Z. Galil, and M. Yung. Eavesdropping games: a graph-theoretic approach to privacy in distributed systems. *J. ACM*, 47(2):225–243, 2000.

[74] Katalin Friedl, Gábor Ivanyos, Miklos Santha, and Yves F Verhoeven. Locally 2-dimensional Sperner problems complete for the polynomial parity argument classes. In *Italian Conference on Algorithms and Complexity*, pages 380–391. Springer, 2006.

[75] Jugal Garg, Ruta Mehta, Vijay V Vazirani, and Sadra Yazdanbod. ETR-completeness for decision versions of multi-player (symmetric) Nash equilibria. *ACM Transactions on Economics and Computation (TEAC)*, 6(1):1, 2018.

[76] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80 – 93, 1989.

[77] Leslie Ann Goldberg. Notes on contention resolution. In *http://www.cs.ox.ac.uk/people/leslieann.goldberg/contention.html*, 2002.

[78] Michelangelo Grigni. A Sperner lemma complete for PPA. *Information Processing Letters*, 77(5-6):255–259, 2001.

[79] Claus-Jochen Haake, Matthias G Raith, and Francis Edward Su. Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4):723–749, 2002.

[80] Kristoffer Arnsfelt Hansen. The real computational complexity of minmax value and equilibrium refinements in multi-player games. *Theory of Computing Systems*, 63(7):1554–1571, Oct 2019.

[81] Kristoffer Arnsfelt Hansen, Michal Koucký, Niels Lauritzen, Peter Bro Miltersen, and Elias P. Tsigaridas. Exact algorithms for solving stochastic games: extended abstract. In *Proc. of STOC*, pages 205–214, 2011.

[82] Johan Håstad, Tom Leighton, and Brian Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM Journal on Computing*, 25(4):740–774, 1996.

[83] Ji Hayes. An adaptive technique for local distribution. *IEEE Transactions on Communications*, 26(8):1178–1186, 1978.

[84] E. Hazan and R. Krauthgamer. How hard is it to approximate the best Nash equilibrium? *SIAM J. Comput.*, 40(1):79–91, 2011.

[85] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[86] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[87] Manish Jain, Vincent Conitzer, and Milind Tambe. Security scheduling for real-world networks. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 215–222. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[88] Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, pages 85–103, 1972.

[89] M. J. Kearns and L. E. Ortiz. Algorithms for interdependent security games. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS]*, pages 561–568, 2003.

[90] Michael J. Kearns and Siddharth Suri. Networks preserving evolutionary equilibria and the power of randomization. In *Proceedings 7th ACM Conference on Electronic*

Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006, pages 200–207, 2006.

[91] Frank P Kelly and Iain M MacPhee. The number of packets transmitted by collision detect random access schemes. *The Annals of Probability*, pages 1557–1568, 1987.

[92] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[93] Joshua Letchford and Vincent Conitzer. Solving security games on graphs via marginal probabilities. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[94] R.C. Lewontin. Evolution and the theory of games. *Journal of Theoretical Biology*, 1(3):382 – 403, 1961.

[95] E Lieberman, C Hauert, and M A Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, January 2005.

[96] Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proc. of EC*, pages 36–41. ACM, 2003.

[97] Richard TB Ma, Vishal Misra, and Dan Rubenstein. Modeling and analysis of generalized slotted-aloha mac protocols in cooperative, competitive and adversarial environments. In *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*, pages 62–62. IEEE, 2006.

[98] Allen B MacKenzie and Stephen B Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1583–1590. IEEE, 2003.

[99] Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *CoRR*, abs/1406.2636, 2014.

[100] M. Mavronicolas, L. Michael, V. G. Papadopoulou, A. Philippou, and P. G. Spirakis. The price of defense. In *Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS*, pages 717–728, 2006.

[101] M. Mavronicolas, V. Papadopoulou, A. Philippou, and P. G. Spirakis. A network game with attackers and a defender. *Algorithmica*, 51(3):315–341, 2008.

[102] M. Mavronicolas, V. G. Papadopoulou, G. Persiano, A. Philippou, and P. G. Spirakis. The price of defense and fractional matchings. In *Distributed Computing and Networking, 8th International Conference, ICDCN*, pages 115–126, 2006.

[103] M. Mavronicolas, V. G. Papadopoulou, A. Philippou, and P. G. Spirakis. A graph-theoretic network security game. In *Internet and Network Economics, First International Workshop, WINE*, pages 969–978, 2005.

[104] J. Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

[105] Nimrod Megiddo and Christos H. Papadimitriou. A note on total functions, existence theorems, and computational complexity. *Theoretical Computer Science*, 81:317–324, 1989.

[106] Ruta Mehta. Constant rank bimatrix games are PPAD-hard. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 545–554, 2014.

[107] Themistoklis Melissourgos, Sotiris Nikoletseas, Christoforos Raptopoulos, and Paul Spirakis. Mutants and residents with different connection graphs in the Moran process. In Michael A. Bender, Martín Farach-Colton, and Miguel A. Mosteiro, editors, *LATIN 2018: Theoretical Informatics*, pages 790–804, Cham, 2018. Springer International Publishing.

[108] Themistoklis Melissourgos and Paul G. Spirakis. Existence of evolutionarily stable strategies remains hard to decide for a wide range of payoff values. In *Algorithms and Complexity - 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings*, pages 418–429, 2017.

[109] George B. Mertzios and Paul G. Spirakis. Strong bounds for evolution in networks. *J. Comput. Syst. Sci.*, 97:60–82, 2018.

[110] Jeonghoon Mo, Hoi-Sheung Wilson So, and Jean Walrand. Comparison of multichannel mac protocols. *IEEE Transactions on mobile computing*, 7(1):50–65, 2008.

[111] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

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

[113] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[114] Asis Nasipuri, Jun Zhuang, and Samir R Das. A multichannel csma mac protocol for multihop wireless networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, volume 3, pages 1402–1406. IEEE, 1999.

[115] Noam Nisan. A note on the computational hardness of evolutionary stable strategies. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(076), 2006.

[116] J.R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[117] Sergei Ovchinnikov. Max-min representation of piecewise linear functions. *BeitrÃďge zur Algebra und Geometrie*, 43(1):297–302, 2002.

[118] C.H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244 – 259, 1984.

[119] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[120] Christos H Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[121] Lawrence G Roberts. Aloha packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, 1975.

[122] Aviad Rubinstein. Inapproximability of Nash equilibrium. *SIAM Journal on Computing*, 47(3):917–959, 2018.

[123] Marcus Schaefer. Complexity of some geometric and topological problems. In *International Symposium on Graph Drawing*, pages 334–344. Springer, 2009.

[124] Marcus Schaefer. Realizability of graphs and linkages. In *Thirty Essays on Geometric Graph Theory*, pages 461–482. Springer, 2013.

[125] Marcus Schaefer and Daniel Stefankovic. Fixed points, Nash equilibria, and the existential theory of the reals. *Theory Comput. Syst.*, 60(2):172–193, 2017.

[126] Mark E. Schaffer. Evolutionarily stable strategies for a finite population and a variable contest size. *Journal of Theoretical Biology*, 132(4):469 – 478, 1988.

[127] Benjamin Weiss Sergiu Hart, Yosef Rinott. Evolutionarily stable strategies of random games, and the vertices of random polygons. *The Annals of Applied Probability*, 18(1):259–287, 2008.

[128] Paulo Shakarian, Patrick Roos, and Anthony N. Johnson. A review of evolutionary graph theory with applications to game theory. *Biosystems*, 107(2):66–80, 2012.

[129] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[130] Mei-Ju Shih, Guan-Yu Lin, and Hung-Yu Wei. A distributed multi-channel feedbackless mac protocol for d2d broadcast communications. *IEEE Wireless Communications Letters*, 4(1):102–105, 2015.

[131] Yaroslav Shitov. A universality theorem for nonnegative matrix factorizations. *arXiv preprint arXiv:1606.09068*, 2016.

[132] Yaroslav Shitov. The complexity of positive semidefinite matrix factorization. *SIAM Journal on Optimization*, 27(3):1898–1909, 2017.

[133] Forest W. Simmons and Francis Edward Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Mathematical Social Sciences*, 45(1):15–25, 2003.

[134] J. Maynard Smith. The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology*, 47(1):209 – 221, 1974.

[135] H Wilson So, Jean Walrand, and Jeonghoon Mo. Mcmac: A multi-channel mac proposal for ad-hoc wireless networks. In *Proc. of IEEE WCNC*, pages 334–339, 2007.

[136] Jungmin So and Nitin H Vaidya. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 222–233. ACM, 2004.

[137] E. H. Spafford. The internet worm: Crisis and aftermath. *Commun. ACM*, 32(6):678–687, 1989.

[138] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM*, 52(10):76–84, 2009.

[139] W. Stallings. *Cryptography and network security - principles and practice (3. ed.)*. Prentice Hall, 2003.

[140] Francis Edward Su. Borsuk-Ulam implies Brouwer: A direct construction. *The American Mathematical Monthly*, 104(9):855–859, 1997.

[141] Fouad Tobagi and Leonard Kleinrock. Packet switching in radio channels: part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on communications*, 23(12):1417–1433, 1975.

[142] Boris Solomonovich Tsybakov and Viktor Alexandrovich Mikhailov. Free synchronous packet access in a broadcast channel with feedback. *Problemy Peredachi Informatsii*, 14(4):32–59, 1978.

[143] Ondřej Vaněk, Zhengyu Yin, Manish Jain, Branislav Bošanskỳ, Milind Tambe, and Michal Pěchouček. Game-theoretic resource allocation for malicious packet detection in computer networks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 905–912. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[144] Haifeng Xu. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 497–514, New York, NY, USA, 2016. ACM.

[145] Jingbin Zhang, Gang Zhou, Chengdu Huang, Sang Hyuk Son, and John A Stankovic. Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks. In

Communications, 2007. ICC'07. IEEE International Conference on, pages 3554–3561. IEEE, 2007.