

Master Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Efficiency of Counterfactual Regret Minimizations Variants in Diverse Domains

Bc. Jan Rudolf

Supervisor: Mgr. Viliam Lisý, MSc., Ph.D.
Field of study: Artificial Intelligence
January 2021

Acknowledgements

I want to thank my supervisor for his support and cooperation. Without it, finishing my studies would not be possible.

Declaration

I declare that I did this thesis alone and cite the used literature.

In Prague, 5 January 2021

Abstract

Since the beginning, algorithms that could competitively play against people's popular games have been on the research's focus in the artificial intelligence field. Two-player poker is one of them as a representative of imperfect-information games. The research on algorithms solving two-player imperfect-information games had given birth to a group of algorithms based on counterfactual regret minimization (CFR). Their authors had been proving the importance of their CFR variant mainly by empirical speed of convergence experiments to approximate Nash equilibria. The games, which they use for experiments, were variants of poker. This thesis is examining if the results of previous uphold to diverse domains. We implemented the most popular published CFR variants Vanilla CFR, CFR+, LCFR, and DCFR. We measured their convergence speed on the different parametrization of Goofspiel, Liar's Dice, Oshi-Zumo, and Darkchess. We found out that the authors of these CFR variants did not skew their results using mostly poker domains. However, some of their statements were too optimistic. We showed older CFR+ with a different weighting of average strategy can better results than newer DCFR. DCFR(1.5, 0.5, 2.0) is not universally the best CFR variant.

Keywords: game theory, regret minimization

Supervisor: Mgr. Viliam Lisý, MSc., Ph.D.

Abstrakt

Algoritmy, které jsou kompetitivně schopné hrát proti lidem populární hry, byly v pozornosti výzkumu v oboru umělé inteligence od počátku. Jako reprezentant her s neúplnou informací byl i dvouhráčový poker. Výzkum algoritmů řešící hry s neúplnou informací dal vzniknout skupině algoritmů minimalizující fiktivní lítost (CFR). V posledních letech vznikla řada CFR variant. Jejich autoři dokazují důležitost jejich varianty převážně na empirických experimentech rychlosti konvergence k aproximovanému Nashovu equilibriu. Hra, na které porovnávali svoje algoritmy, byl převážně právě poker. Tato práce se zjišťuje, zda svoje výsledky obhájí na více hrách. Implementujeme nejpopulárnější publikované CFR varianty CFR, CFR+, LCFR a DCFR. Měříme jejich rychlost konvergence na různých parametrech her Goofspiel, Liar's Dice, Oshi-Zumo a Darkchess. Zjistili jsme, že autoři původně provedených experimentů pouze na pokeru příliš nezakreslili svoje závěry, nicméně některé tvrzení byli příliš optimistické. Ukazujeme, starší CFR+ má v některých hrách rychlejší konvergenci než novější DCFR, nebo že DCFR(1.5, 0.5, 0) není vždy nejlepší varianta, jak tvrdí autor.

Klíčová slova: teorie her, minimalizace lítosti

Překlad názvu: Efektivita variant algoritmu minimalizace fiktivní lítosti v různých doménách

Contents

1 Introduction	1	4 Survey of Tabular CFR Variants	15
1.1 Thesis Goal	2	4.1 CFR (Vanilla CFR)	16
1.2 Thesis Outline	2	4.2 CFR+	17
2 Game Theory Fundamentals	3	4.2.1 Regret Matching+	17
2.1 Modeling Games	3	4.2.2 Alternating Updates	18
Normal-Form Games	3	4.2.3 Weighting the Average Strategy	18
Extensive-Form Games	4	4.3 Linear CFR	18
Computationally Convenient Properties	6	4.3.1 Weighting the Average Strategy	19
2.2 Computing Strategies	6	4.3.2 Weighting the Cumulative Counterfactual Regret	19
2.3 Evaluating Strategies	10	4.4 Discounted CFR	19
3 Computing Strategies via Online Learning	11	5 Experiments	21
3.1 Online Learning and Prediction	11	5.1 Games	21
3.2 Computing Strategies for NFG	12	5.1.1 Goofspiel	21
3.3 Computing Strategies for EFG	14	5.1.2 Liar's Dice	22
		5.1.3 Oshi-Zumo	22
		5.1.4 Darkchess	23

5.2 Correctness of LCFR and DCFR	23
5.3 Empirical Speed of Convergence	24
5.3.1 Experiment Setup	25
5.3.2 Empirical Speed of Convergence on Goofspiel	25
5.3.3 Empirical Speed of Convergence on Liar’s Dice	40
5.3.4 Empirical Speed of Convergence on Oshi-zumo	45
5.3.5 Empirical Speed of Convergence on Darkchess	46
5.4 Discussion	47
6 Conclusion	49
Bibliography	51

Figures

2.1 The figure shows the linear program for computing a possibly mixed Nash equilibrium strategy s for player 1. U is the Value of the game (player 1 expected utility in NE). . .	9
5.1 Empirical speed of convergence by Noam Brown.	24
5.2 Empirical speed of convergence on perfect-information Goofspiel with 5 cards with scalar utilities and ascending prize cards A, 2, 3, 4, 5.	24

Tables

2.1 Rock-Paper-Scissors modeled as a two-player zero-sum normal-form game. Rows of the matrix represents actions of player 1, columns represents actions of player 2.	4
2.2 RPS with a pure strategy.	4
2.3 The table shows the Rock-Paper-Scissor game in normal form, where player 2 has a fixed beviour strategy $\sigma_2 = (0, 1, 0)$, and player 1 has an unknown behavior strategy $\sigma_1 = (x_1, x_2, x_3)$. Player 1 wants to get as much utility as possible, so his best response to player 2 is, $\sigma_1^* = BR_1(\sigma_2) = (0, 0, 1)$, to player Scissors. The strategy profile (σ_1^*, σ_2) has the outcome 1 for the player 1 and -1 for the player 2.	7
2.4 The table shows Rock-Paper-Scissors with a strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*) = ((\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}))$. The strategy profile σ^* is a Nash equilibrium of the game. The outcome of the game is for both players 0.	8
5.1 Basic game statistics of perfect-information Goofspiel with 3 cards. The Depth column corresponds to the maximal depth of EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.	26

5.2 Basic game statistics of imperfect-information Goofspiel with 3 cards. The Depth column corresponds to the maximal depth of EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.	26
5.3 The table shows heatmaps for perfect-information Goofspiel 3 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.	26
5.4 The table shows heatmaps for imperfect-information Goofspiel 3 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.	27
5.5 Speed of convergence on perfect-information Goofspiel 3. The first row contains the game's variant with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. Similarly, the second row contains the game's variant with scalar utilities. Figure (c) has randomized Chance, and figure (d) fixed Chance. An x-axis shows the number of iteration. A y-axis is logarithmic and shows average exploitability.	28
5.6 Speed of convergence on imperfect-information Goofspiel 3. The first row contains the game's variant with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. Similarly, the second row contains the game's variant with scalar utilities. Figure (c) has randomized Chance, and figure (d) fixed Chance. An x-axis shows the number of iteration. A y-axis is logarithmic and shows average exploitability.	29
5.7 Speed of convergence on imperfect-information Goofspiel 3. The first row contains the game's variant with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. Similarly, the second row contains the game's variant with scalar utilities. Figure (c) has randomized Chance, and figure (d) fixed Chance. An x-axis shows the number of iteration. A y-axis is logarithmic and shows average exploitability.	29
5.8 The strategy statistics after running CFR for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and fixed Chance. Depth means EFG depth of the game tree. The row Pure shows number of information sets with a pure strategy based of the EFG depth. The row Mixed shows the same for a mixed strategy. The last row show a ratio between mixed strategies compared to all strategies in that depth.	30

5.9 The strategy statistics after running CFR+ with quadratic averaging for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and fixed Chance.	30
5.10 The strategy statistics after running CFR for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and randomized Chance.	30
5.11 The strategy statistics after running CFR+ with quadratic averaging for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and randomized Chance.	30
5.12 Basic game statistics of perfect-information Goofspiel with 4 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.	31
5.13 Basic game statistics of imperfect-information Goofspiel with 4 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.	31
5.14 The table shows heatmaps for perfect-information Goofspiel 4 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.	32
5.15 The table shows heatmaps for imperfect-information Goofspiel 4 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.	32
5.16 Speed of convergence on perfect-information Goofspiel 4 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.	33
5.17 Speed of convergence on perfect-information Goofspiel 4 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.	34

5.18 Speed of convergence on imperfect-information Goofspiel 4 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.	35
5.19 Speed of convergence on imperfect-information Goofspiel 4 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.	35
5.20 Basic game statistics of perfect-information Goofspiel with 5 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.	36
5.21 Basic game statistics of imperfect-information Goofspiel with 5 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.	36
5.22 The table shows heatmaps for perfect-information Goofspiel 5 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.	36
5.23 The table shows heatmaps for imperfect-information Goofspiel 5 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.	37
5.24 Speed of convergence on perfect-information Goofspiel 4 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.	37
5.25 Speed of convergence on perfect-information Goofspiel 5 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.	38

<p>5.26 Speed of convergence on imperfect-information Goofspiel 5 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale. 38</p>	<p>5.31 Basic game statistics of Liar's Dice with 2 dice per player and various number of faces. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions. 40</p>
<p>5.27 Speed of convergence on imperfect-information Goofspiel 5 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale. 39</p>	<p>5.32 Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and two faces on each die. DCFR variants and CFR+ with quadratic averaging have similar convergence curves. Figure (a) shows the size of the information set in dependence of EFG depth. 40</p>
<p>5.28 The strategy statistics after running LCFR with quadratic averaging for 8192 iterations on imperfect-information Goofspiel 5 with binary utilities and fixed Chance. 39</p>	<p>5.33 Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and three faces on each die. LCFR and CFR+ have identical convergence curves. Also, DCFR variants and CFR+ with quadratic averaging have similar convergence curves. Figure (a) shows the size of the information set in dependence of EFG depth. 41</p>
<p>5.29 The strategy statistics after running DCFR(1.5, 0.5, 2) for 8192 iterations on imperfect-information Goofspiel 5 with binary utilities and randomized Chance..... 39</p>	<p>5.34 Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and four faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth. 41</p>
<p>5.30 Basic game statistics of Liar's Dice with 1 dice per player and various number of faces. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions. 40</p>	<p>5.35 Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and five faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth. 42</p>

<p>5.36 Figure (b) shows the speed of convergence on Liar’s Dice with one dice per player and six faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth. 42</p> <p>5.37 Figure (b) shows the speed of convergence on Liar’s Dice with two dices per player and two faces on each die. LCFR and CFR+ have identical convergence curves under the red curve. Also, DCFR variants and CFR+ with quadratic averaging have similar convergence curves under the purple curve. Figure (a) shows the size of the information set in dependence of EFG depth. 43</p> <p>5.38 Figure (b) shows the speed of convergence on Liar’s Dice with one dice per player and three faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth. 43</p> <p>5.39 Figure (b) shows the speed of convergence on Liar’s Dice with two dices per player and four faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth. 44</p> <p>5.40 Speed of convergence on Oshi-Zumo. 5 starting point, 10 coins, 1 min. bid. 45</p> <p>5.41 Speed of convergence on the Darkchess’s minimal 4x3 board. Figure A shows the convergence graph for 2 moves per player. Figure B shows the convergence graph for 3 moves per player. 46</p>	<p>5.42 Speed of convergence on the Darkchess’s minimal 4x3 board. Figure A shows the convergence graph for 4 moves per player. Figure B shows the convergence graph for 5 moves per player. 47</p>
---	---

I. Personal and study details

Student's name: **Rudolf Jan** Personal ID number: **420776**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Computer Science**
Study program: **Open Informatics**
Specialisation: **Artificial Intelligence**

II. Master's thesis details

Master's thesis title in English:

Efficiency of Counterfactual Regret Minimizations Variants in Diverse Domains

Master's thesis title in Czech:

Efektivita variant algoritmu minimalizace fiktivní lítosti v různých doménách

Guidelines:

Counterfactual Regret Minimization (CFR) is the scheme of algorithms that was key in the recent results in achieving super-human performance in large variants of poker. Large flexibility of the scheme led to its many instances with practically the same theoretical guarantees, but very diverse empirical performance. Since poker was the large standing challenge problem, they were rarely evaluated on other domains and therefore, it is not clear that their relative performance observed in poker holds in general. Therefore, the student will:

- 1) Survey the existing variants of tabular CFR (e.g., CFR+, Linear CFR, Discounted CFR, Lazy CFR);
- 2) choose at least three and implement them in the existing software framework;
- 3) rigorously choose at least three (ideally more) diverse domains for evaluation;
- 4) empirically compare the speed of convergence of CFR variants on the selected domains.

Bibliography / sources:

Zinkevich M, Johanson M, Bowling M, Piccione C. Regret minimization in games with incomplete information. In Advances in neural information processing systems 2008 (pp. 1729-1736).
Bowling M, Burch N, Johanson M, Tammelin O. Heads-up limit hold'em poker is solved. Science. 2015 Jan 9;347(6218):145-9.
Brown N, Sandholm T. Solving imperfect-information games via discounted regret minimization. In Proceedings of the AAAI Conference on Artificial Intelligence 2019 Jul 17 (Vol. 33, pp. 1829-1836).
Farina G, Kroer C, Brown N, Sandholm T. Stable-Predictive Optimistic Counterfactual Regret Minimization. arXiv preprint arXiv:1902.04982. 2019 Feb 13.
Brown N, Sandholm T. Superhuman AI for multiplayer poker. Science. 2019 Aug 30;365(6456):885-90.

Name and workplace of master's thesis supervisor:

Mgr. Viliam Lisý, MSc., Ph.D., Artificial Intelligence Center, FEE

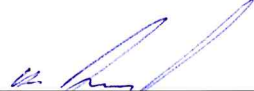
Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **11.02.2020** Deadline for master's thesis submission: _____

Assignment valid until: **30.09.2021**


Mgr. Viliam Lisý, MSc., Ph.D.
Supervisor's signature


Head of department's signature


prof. Mgr. Petr Páta, Ph.D.
Dean's signature



Chapter 1

Introduction

Games and competitions are essential parts of human lives and serve various purposes. The most obvious one that everyone is associating with them is entertainment. The other is extending and expanding mental and physical capabilities, pushing one's limits to be better than the day before. A natural consequence is comparing each other within a friendly match or on a world championship level. We can view games and competitions as an optimization method, selecting the most competitive individuals or groups, forming hierarchies in society. Besides the obvious, we can see games and business companies competitions, social interactions, or even warfare.

The mathematical formalism for games and strategic interaction of decision-makers is game theory. Game theory has found use in various scientific fields, including computer science and artificial intelligence. Games are interesting for artificial intelligence because they serve as a testbed for progress in algorithms. The latest successes of computer programs playing games are AlphaZero ([17]) playing chess, shogi, or Go better than any human, DeepStack ([12]) and Libratus ([3]) that can play poker also on superhuman level, Five ([14]) with Dota 2, or AlphaStar ([21]) that is playing StarCraft 2. These algorithms' common theme is a connection between the core algorithm and neural networks as function approximators. This thesis's subject is one of these core algorithms, Counterfactual Regret Minimization (CFR), developed upon poker games.

The research of CFR and its variants has been strongly influenced by apply-

ing these algorithms to more and more realistic poker instances. Researchers have been comparing these variants by the ability to prove lower worst-case bounds, or quicker speed of convergence to the optimal strategyt mostly on poker games variations. It is not clear how these CFR based algorithms are going to perform on a more diverse set of games.

■ 1.1 Thesis Goal

The thesis aims to provide a more comprehensive empirical study of CFR based algorithms on a diverse set of games. Concretely:

- **A** Survey the existing variants of tabular CFR.
- **B** Choose at least three and implement them in the existing software framework.
- **C** Rigorously choose at least three diverse domains for evaluation.
- **D** Empirically compare the speed of convergence of CFR variants on selected domains.

■ 1.2 Thesis Outline

Chapter 2 introduces the fundamental game-theoretic concepts we use throughout the thesis. Chapter 3 extends these concepts and connects them with online learning. Chapter 4 describes the Counterfactual Regret Minimization algorithm and its most important variations. Chapter 5 presents games and experiments performed on them. Chapter 6 concludes the thesis.



Chapter 2

Game Theory Fundamentals

Game theory is mathematically modeling strategic interaction between decision-makers we call players. We restrict ourselves to the non-cooperative game theory, where each player is self-interested, independent, and rational, that is, maximizing his expected utility. The following restriction is the set of games. We will deal with games for two players and other properties described in section 2.1 Modeling Games. Section 2.2 Computing Strategies explains what we mean by strategy, optimal strategy, and how we can compute or approximate optimal strategy. In the last section, we discuss how to evaluate strategy and estimate how far is computed strategy from the optimal.



2.1 Modeling Games

Games model simplified life situations[Edit: in every scientific field, models siplifies real world situations]. Every game has a set of decision points, which belong to one of the players. Players have defined actions on these decision points, and they are choosing their actions according to their strategy. Players' interaction and their utility functions form a strategy. The result of their interaction is an outcome at the end of the game. Two standard formalisms for modeling games in game theory are normal-form games (NFGs) and extensive-form games (EFGs).

Normal-Form Games

A normal-form game¹ is the most basic model, where every player performs one action simultaneously with other players. Each player has his own utility function, which assigns him a payoff based on all players' chosen actions.

Definition 2.1. (Normal Form Game) A finite, n-person normal-form game is a tuple (N, A, u) , where:

- N is a finite set of n players indexed by i ,
- $A = A_1 \times \dots \times A_n$ where A_i is a finite set of actions available to player i ,
- $u = (u_1, \dots, u_n)$ where $u_i : A \mapsto \mathbb{R}$ is a real-valued utility (payoff) function for player i .

1 \ 2	Rock	Paper	Scissors
Rock	(0, 0)	(-1, 1)	(1, -1)
Paper	(1, -1)	(0, 0)	(-1, 1)
Scissors	(-1, 1)	(1, -1)	(0, 0)

Table 2.1: Rock-Paper-Scissors modeled as a two-player zero-sum normal-form game. Rows of the matrix represents actions of player 1, columns represents actions of player 2.

1 \ 2		Rock	Paper	Scissors
	σ	0	1	0
Rock	1	(0, 0)	(-1, 1)	(1, -1)
Paper	0	(1, -1)	(0, 0)	(-1, 1)
Scissors	0	(-1, 1)	(1, -1)	(0, 0)

Table 2.2: RPS with a pure strategy.

Extensive-Form Games

An extensive-form game is a model or formalism by which we describe sequential (dynamical) games. Every game usually contains multiple states,

¹Normal-form games are synonymous with strategic games, one-shot games, or matrix games.

and players make more than one decision during the play. Imagine you are starting a Chess game as a white player. The first state (decision point) of the game is your and opponent figures in starting positions. You need to decide on a move. Every legal move leads to a new board placement (a new state) where your opponent is acting. The game in EFG formalism forms a tree called a game tree. Every internal node is a decision point of one of the players, and leaf nodes, also called terminal nodes, assign every player utility, then the game ends.

Chess is an example of a perfect-information game, where the opponent has the same amount of information about the game's current state. You see all of the opponent's pieces. The opponent sees all your pieces.

Definition 2.2. (Perfect-information EFG) A finite perfect-information game in extensive form is a tuple $G = (N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- N is a finite set of $n \in \mathbb{N}_0^+$ players,
- A is a finite set of actions,
- H is a set of nonterminal decision nodes (history),
- $\chi : H \mapsto 2^A$ is the action function, which assigns to each choice node a set of possible actions,
- $\rho : H \mapsto N$ is the player function, which assigns to each nonterminal node a player $i \in N$ who chooses an action at that node,
- $\sigma : H \times A \mapsto H \cup Z$ is the successor function, which maps a choice node and an action to a new choice node or terminal node,
- $u = (u_1, \dots, u_n)$, where $u_i : Z \mapsto \mathbb{R}$ is a real-valued utility function for player i on the terminal nodes Z .

In imperfect-information EFG, at least one player can have some information about the current game state hidden. In the Chess variant called Kriegspiel, players do not see the opponent's pieces, to continue with the Chess analogy. Players have to reason about all the possible opponent's figures configurations on the board. It is going to be usually more than one state as it is in a perfect-information variant. We factorize the game states for every player into information sets. An information set contains every state of the game a player cannot distinguish based on his current history.

Definition 2.3. (Imperfect-information EFG) An imperfect-information game in extensive form is a tuple $(N, A, H, Z, \chi, \rho, \sigma, u, I)$, where:

- $(N, A, H, Z, \chi, \rho, \sigma, u)$ is a perfect-information extensive form game,
- $I = (I_1, \dots, I_m)$, where $I_i = (I_{i,1}, \dots, I_{i,k_i})$ is an equivalence relation on $\{h \in H : \rho(h) = i\}$ with the property that $\chi(h) = \chi(h')$ and $\rho(h) = \rho(h')$ whenever there exists a j for which $h \in I_{i,j}$ and $h' \in I_{i,j}$.

■ Computationally Convenient Properties

Algorithms have been mainly developed for perfect-information, zero-sum games.

Definition 2.4. (Perfect recall [18]) Player i has perfect recall in an imperfect-information game G if for any two nodes h, h' that are in the same information set for player i , for any path $h_0, a_0, h_1, a_1, h_2, \dots, h_m, a_m, h$ from the root of the game to h (where the h_j are decision nodes and the a_j are actions) and for any path $h_0, a'_0, h'_1, a'_1, h'_2, \dots, h'_m, a'_m, h'$ from the root to h' it must be the case that:

- $m = m'$,
- for all $0 \leq j \leq m$, if $\rho(h_j) = i$, then h_j and h'_j are in the same equivalence class for i ,
- for all $0 \leq j \leq m$, if $\rho(h_j) = i$, then a_j and a'_j .

G is a game of perfect recall if every player has perfect recall in it.

■ 2.2 Computing Strategies

In the previous section, we learned that a game G consists of histories H_G representing the game's states, and information sets $I_G = I_1 \cup I_2 \cup I_c$ factorize histories based on the players private information. In every history $h \in H_G$, one of the players $i \in N$ is acting with one of his actions $a \in A_h$ determined by the game's rules. Because an information set $I \in I_i$ contains multiple histories that the player i cannot distinguish, the player i acts the same within all information set histories. Therefore we can talk about the player's i actions A_I over the information set I .

A player $i \in N$ having a strategy σ_i intuitively means the player knows how to act in every information set $I \in I_i$. He has a description of what action $a \in A_I$ to play when reaching the information set for every $I \in I_i$. Formally, a mapping $\sigma_i : I \mapsto A_I$ is called a pure (deterministic) strategy² of player i . The strategy profile $\sigma = (\sigma_1, \sigma_2, \sigma_c)$ is an N -tuple of players' strategies. By σ_{-i} , we mean a strategy profile of all players' strategies except player's i .

Pure strategies aren't describing the whole strategy space. It is often convenient for a player to choose his action in a randomized way. For example, player 1 could choose the action Rock with 0.4, Paper with 0.5, and Scissors with 0.1 probability in the Rock-Paper-Scissors game. In other words, this randomized strategy is assigning 40% for a pure strategy playing Paper, 50% for playing Scissors, and 10% for playing Scissors. A mixed strategy σ_i is a probability distribution over all player's i pure strategies. People naturally don't reason in terms of mixed strategies, meaning randomizing their pure strategies for the whole game. The usual approach is to randomize over actions in the current situation (information set). Player's i behavior strategy, $\sigma_i : I_i \mapsto \Delta(A_{I_i})$, maps from the information set $I \in I_i$ to a probability simplex $\Delta(A_I)$ over actions A_I (also denoted $\beta_i(I)$). Theorem 2.5 states that mixed and behavior are strategies equivalent for perfect-recall games.

Theorem 2.5. (Equivalence of mixed and behavior strategy [18]) *In a game of perfect recall, any mixed strategy of a given agent can be replaced by an equivalent behavioral strategy, and any behavioral strategy can be replaced by an equivalent mixed strategy. Here two strategies are equivalent in the sense that they induce the same probabilities on outcomes, for any fixed strategy profile (mixed or behavioral) of the remaining agents.*

1 \ 2		Rock	Paper	Scissors
	σ	0	1	0
Rock	x_1	(0, 0)	(-1, 1)	(1, -1)
Paper	x_2	(1, -1)	(0, 0)	(-1, 1)
Scissors	x_3	(-1, 1)	(1, -1)	(0, 0)

Table 2.3: The table shows the Rock-Paper-Scissor game in normal form, where player 2 has a fixed behavior strategy $\sigma_2 = (0, 1, 0)$, and player 1 has an unknown behavior strategy $\sigma_1 = (x_1, x_2, x_3)$. Player 1 wants to get as much utility as possible, so his best response to player 2 is, $\sigma_1^* = BR_1(\sigma_2) = (0, 0, 1)$, to player Scissors. The strategy profile (σ_1^*, σ_2) has the outcome 1 for the player 1 and -1 for the player 2.

²Strategy is similar to policy (control) in the reinforcement learning (control theory) literature.

$$\max_{s,U} U \quad (2.1)$$

$$s.t. \sum_{a_1 \in A_1} s(a_1)u_1(a_1, a_2) \geq U \quad \forall a_2 \in A_2 \quad (2.2)$$

$$\sum_{a_1 \in A_1} s(a_1) = 1 \quad (2.3)$$

$$s(a_1) \geq 0 \quad \forall a_1 \in A_1 \quad (2.4)$$

Figure 2.1: The figure shows the linear program for computing a possibly mixed Nash equilibrium strategy s for player 1. U is the Value of the game (player 1 expected utility in NE).

utility in every information set. This way, we can compute a Maxmin strategy (value), definition 2.9, in terms of the player's i utility and Minmax strategy (value), definition 2.10, in terms of $-i$ utility. The Maxmin value of player 1 has the name Value of the game. The Minimax (Maximin) algorithm computes these (pure) strategies, and figure 2.1 shows a linear program for computation of behavior (mixed) strategy for perfect-information games.

Definition 2.9. (Maxmin strategy and value [18]) The maxmin strategy for player $i \in N$ is $\operatorname{argmax}_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$ and the maxmin value for player i is $\max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$.

Definition 2.10. (Minmax strategy and value [18]) In a two-player game, the minmax strategy for player $i \in N$ is $\operatorname{argmin}_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$ and the minmax value for player $-i$ is $\min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$.

The Minimax theorem, theorem 2.11, connects Nash equilibrium and Maxmin (Minmax) strategies for finite, two-player, zero-sum games.

Theorem 2.11. (Minimax theorem [18]) In any finite, two-player, zero-sum game, in any Nash equilibrium each player receives a payoff that is equal to both his maxmin and his minmax value.

We can conclude from theorem 2.8 and 2.11 that there exists a Nash equilibrium strategy profile for any finite, two-player, zero-sum game. This strategy profile is equal to Maxmin and Minmax strategy profiles. Maxmin (minmax) values are equal to the Value of the game for every Nash equilibrium.

Another useful solution concept in this thesis is ϵ -Nash equilibrium. ϵ -Nash equilibrium is an approximation of Nash equilibrium, i.e., for $\epsilon = 0$.

Definition 2.12. (ϵ -Nash equilibrium) Fix $\epsilon > 0$. A strategy profile $s = (s_1, \dots, s_n)$ is a ϵ -Nash equilibrium if, for all agents i and for all strategies

$$s'_i \neq s_i, u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}) - \epsilon.$$

■ 2.3 Evaluating Strategies

Strategy profile distance from Nash eq. is measured by exploitability.

Definition 2.13. (Exploitability [1])

$$e(\sigma_i) = u_i(\sigma_i^*, BR(\sigma_i^*)) - u_i(\sigma_i, BR(\sigma_i))$$

Definition 2.14. (Average Exploitability [1][8])

$$e(\sigma) = \frac{1}{|N|} \sum_{i \in N} e(\sigma_i)$$

Chapter 3

Computing Strategies via Online Learning

This chapter describes online learning in games. First, we describe online learning problem in general, then we focus on computing strategies for NFG and EFG. This chapter is based on [16], [5], [11] [7].

3.1 Online Learning and Prediction

Prediction (forecast) is a process of estimating some phenomena' future value based on a model built from past observations (experience), or more profound knowledge about the modeling system, and current observation (information we can extract in the present moment). Sooner or later, we can verify our predictions and measure how good or bad predictions we are making as a predictor (forecaster). Different fields frame this problem from their angle of viewpoint and setting. For example, in reinforcement learning based on psychology, the predictor is called an agent, who builds a model from rewards he gets from an interaction with the environment to maximize his expected reward. In control theory, the predictor is called a controller, usually controlling some physical process. The controller wants to minimize the cost because of suboptimal control that could lead to bad outcomes like a loss of spacecraft. In machine learning, there are approaches like Probably Approximately Correct (PAC) framework or online learning. PAC framework learning (like supervised learning) assumes data (observations) are independent and identically distributed from a fixed probability distribution. The model is trained from the training data set in a batched fashion to generalize to the actual underlying probability distribution. The generalization measures

empirical risk minimization. On the other hand, online learning does not have any assumption about the source of data. Online in this context means the predictor can make predictions right away after the first observation received and does not have to be trained by any data set beforehand. The workflow of online learning is as follows, during T iterations of the online predictor operations, the predictor's model m_t receives at iteration t an observation $x_t \in X$ and makes a prediction $\hat{y}_t = m_t.predict(x_t) \in Y$, then the predictor receives the true value $y_t \in Y$ which incurs loss $l(y_t, \hat{y}_t)$. The predictor uses the loss to update the model $m_{t+1} = m_t.update(l(y_t, \hat{y}_t))$. Suppose the model contains a countable set of hypotheses that contains the true hypothesis. In that case, we use the loss notion of mistakes, and we bound the predictor by the number of mistakes the predictor will make until he finds the right (mistake-free) hypothesis. Otherwise, we use the concept of regret and will use it from now on.

Definition 3.1. (External regret [15]) Fix reward vector r^1, r^2, \dots, r^T . The external regret of the action sequence a^1, \dots, a^T is

$$R^T = \max_{a \in A} \sum_{t=1}^T r^t(a) - \sum_{t=1}^T r^t(a^t)$$

3.2 Computing Strategies for NFG

Regret Matching (RM)[7] is an iterative, no-regret, anytime algorithm for approximately solving, computing ϵ -Nash equilibrium, normal-form games using self-play. Using a self-play means that both players are using an instance of the same algorithm. Imagine player 1 and player 2 are playing the same NFG G repeatedly for T iterations against each other.

Player i has to choose a strategy σ_i^t at iterations t , probabilities over his actions, ideally in a way that guarantees his strategy is improving. If player i plays according to RM, he maintains cumulative regret $R_i \in \mathbb{R}^{|A_i|}$ and average strategy $\bar{\sigma}_i \in \mathbb{R}^{|A_i|}$. He initializes them with the zero vector before the first iteration. Player's i current strategy σ_i^t is proportional to the positive cumulative regret at iteration t :

$$\sigma_i^t(a) = \begin{cases} R_i^t(a)^+ / \sum_{b \in A_i} R_i^t(b)^+ & \text{if } \sum_{b \in A_i} R_i^t(b)^+ > 0 \\ \frac{1}{|A_i|} & \text{otherwise} \end{cases} \quad (3.1)$$

for each action $a \in A_i$, where $x^+ = \max(x, 0)$ for any $x \in \mathbb{R}$. After the current strategy is computed, it is added to the average strategy:

$$\bar{\sigma}_i^t = \frac{1}{t} \sum_{j=1}^t \sigma_i^j = \frac{t-1}{t} \bar{\sigma}_i^{t-1} + \frac{1}{t} \sigma_i^t \quad (3.2)$$

The instantaneous regret $\Delta R_i(a) \in \mathbb{R}$ for not playing $a \in A_i$:

$$\Delta R_i(a) = \sigma_i^t(a) \sigma_{-i}^t u_i(a) - \sum_{b \in A} \sigma^t(b) u^t(b) \quad (3.3)$$

The instantaneous regret $\Delta R_i(a)$ added to the cumulative regret used for the current strategy computation in the next iteration $t + 1$.

$$R_i^{t+1}(a) = R_i^t(a) + \Delta R_i(a) \quad (3.4)$$

RM produces $\sigma_i^1, \dots, \sigma_i^T$ strategies. The average strategy of player i is $\bar{\sigma}_i^T = \frac{1}{T} \sum_{t=1}^T \sigma_i^t$.

Theorem 4 forms an important connection between average cumulated regrets, average strategies, and ϵ -Nash equilibrium that is central to all algorithms in this thesis.

Theorem 3.2. ([1]) *Let G be a two-player, zero-sum game. If the average regrets after T iterations are $\frac{R_i^T}{T} \leq \epsilon_i$, $\epsilon_i > 0$, for each player $i \in N$, then the strategy profile $\bar{\sigma}^T$, made from average strategies, is a $\frac{\epsilon_1 + \epsilon_2}{2}$ -Nash equilibrium in G .*

3.3 Computing Strategies for EFG

A finite, two-player, perfect-recall game in normal-form has a strategically equivalent game in extensive form, where each player has one information set. RM keeps two vectors, cumulative regrets R_i and average strategy $\bar{\sigma}_i$, per player i , which corresponds to keeping them for one information set in the EFG. Therefore, R_i^T is the overall regret for the entire strategy incurred after T iterations. CFR is extending this approach for any EFG, which usually has more than one information set per player. Zinkevich et al. at [23] showed that minimizing cumulative regrets $R_i(I)$ of each information set $I \in I_i$ minimizes the overall external regret R_i of the player's i strategy.

CFR also operates iteratively and maintains a cumulative regret vector and average strategy for each information set. It is using RM to update an information set's current strategy by the cumulative regret. The current strategy of an information set refers to a strategy at iteration t . After CFR computes the current strategy by RM, it adds the current strategy to the average strategy. The critical difference is how CFR computes regrets at each iteration. CFR computes regrets by a type of expected utility for imperfect information EFG called counterfactual (utility) value. Counterfactual values are computed by considering the current strategies of both players and terminal utilities. Then, the immediate counterfactual regret vector is computed by counterfactual values induced by current strategies. The immediate counterfactual regret is added to the cumulative regret, also called cumulative counterfactual regret, and closes the cycle.



Chapter 4

Survey of Tabular CFR Variants

CFR [23], also called Vanilla CFR, is an iterative, anytime algorithm guaranteed to converge to ϵ -Nash equilibrium using self-play in two-players, zero-sum, perfect-recall, imperfect-information games. The algorithm became very popular amongst researchers and scientists in the game solving community, especially those trying to solve poker. Vanilla CFR gained popularity because of better computer memory requirements than previously used linear programming techniques, better empirical performance than its asymptotic worst-case bounds, easy modification, and the absence of hyperparameters. The research in the following tabular CFR variants built upon Vanilla CFR advanced towards decreasing convergence computation time, memory requirements, and tightening the asymptotic worst-case bounds.

We present survey of tabular CFR variants which are using regret matching types of strategy updates in chronological order. First, we describe Vanilla CFR, then empirically better performing CFR+[20][19]. Zhou et al. introduced Lazy CFR [22] in 2018. Brown and Sandholm introduced Linear CFR and Discounted CFR in 2019 [2]. The second variant in 2019 was Instant CFR [9]. During the writing of the thesis, Farina and Sandhold presented a new state of the art variant called Predictive CFR [6]. A similar survey of CFR variants exists [10], but the survey is not up to date and contains critical errors at least in the definition of counterfactual regret.

4.1 CFR (Vanilla CFR)

Vanilla CFR, an alternative name for the original CFR algorithm [23] described in chapter 3.3, consists of 4 conceptual steps. The following description is based on [4].

For iteration t from $1, 2, \dots, T$:

1. Compute current strategy from cumulative counterfactual regrets. For each information set I , each action $a \in A(I)$, player $i = p(I)$:

$$\sigma_i^t(I, a) = \begin{cases} R^t(I, a)^+ / \sum_{b \in A(I)} R^t(I, b)^+ & \text{if } \sum_{b \in A(I)} R^t(I, b)^+ > 0 \\ \frac{1}{|A(I)|} & \text{otherwise} \end{cases}$$

2. Update the average strategy to include the new current strategy. For each information set I , each action $a \in A(I)$, player $i = p(I)$:

$$\bar{\sigma}_i^t(I, a) = \frac{1}{t} \sum_{t'=1}^t \pi_i^{\sigma^{t'}}(t') \sigma_i^{t'}(I, a) = \frac{t-1}{t} \bar{\sigma}_i^{t-1} + \frac{1}{t} \sigma_i^t$$

3. Compute counterfactual values. For each information set I , each action $a \in A(I)$, player $i = p(I)$:

$$v_i^{\sigma^t}(I, a) = \sum_{h \in I \cdot a} v_i^{\sigma^t}(h) = \sum_{h \in I \cdot a} \sum_{z \in Z, h \sqsubset z} \pi_{-i}^{\sigma^t}(h) \pi^{\sigma^t}(z|h) u_i(z)$$

4. Update cumulative counterfactual regret with immediate counterfactual regret, that is computed from counterfactual values and current strategy. For each information set I , each action $a \in A(I)$, player $i = p(I)$:

$$R^{t+1}(I, a) = R^t(I, a) + v_i^{\sigma^t}(I, a) - \sum_{b \in A(I)} \sigma^t(I, b) v_i^{\sigma^t}(I, b)$$

where initial values of cumulative counterfactual regrets are, $R^1(I, a) = 0$, zero.

4.2 CFR+

CFR+ [20] is an algorithm introduced by Oskar Tammelin in 2014. CFR+ is based on Vanilla CFR. CFR+ use three tricks that did not prove the algorithm is superior to Vanilla CFR, but they made the algorithm perform empirically better on poker than Vanilla CFR:

1. using Regret Matching+
2. alternating updates between players
3. linearly weighting the average strategy

The following paper [19] grounded CFR+ theoretically and proved the same convergence guarantees compared to Vanilla CFR. The following three subsections describe these three tricks in CFR+.

4.2.1 Regret Matching+

Regret Matching+ (RM+) uses the observation that Regret Matching never evaluates negative values of cumulative regrets. Equation 3.1 in RM computes the current strategy σ_i^t at iteration t from cumulative regrets with negative values clipped to 0. Therefore, RM+ is working exactly the same as RM with the following update of the current strategy instead of the equation 3.1:

$$\sigma_i^t(a) = \begin{cases} R_i^t(a) / \sum_{b \in A_i} R_i^t(b) & \text{if } \sum_{b \in A_i} R_i^t(b) > 0 \\ \frac{1}{|A_i|} & \text{otherwise} \end{cases} \quad (4.1)$$

and the following update of cumulative regrets, the equation 4.2, that do not store negative values instead of the equation 3.4.

$$R_i^{t+1}(a) = (R_i^t(a) + \Delta R_i(a))^+ = \max(R_i^t(a) + \Delta R_i(a), 0) \quad (4.2)$$

4.2.2 Alternating Updates

At iteration t , Vanilla CFR updates current strategies for all players, all information sets I , then Vanilla CFR is using these strategies, σ_1^t and σ_2^t , to compute counterfactual values and regrets.

However, CFR+ at iteration t first updates counterfactual regrets and the current strategy for player 1, then updates counterfactual regrets for player 2 using player's 1 new current strategy and player's 2 counterfactual regrets from previous iteration. Then it updates also player's 2 current strategy. It alternates updates between players within an iteration.

4.2.3 Weighting the Average Strategy

Vanilla CFR and RM compute the average strategy as a uniform mean over T iterations. CFR+ weights current strategies linearly. The average strategy $\bar{\sigma}_i^t$ is weighted by t at iteration t out of T .

$$\bar{\sigma}_i^t(I, a) = \frac{2}{t(t+1)} \sum_{k=1}^t k \pi_i^{\sigma^k}(I) \sigma_i^k(I, a) \quad (4.3)$$

4.3 Linear CFR

Linear CFR [2] was introduced by Noam Brown and Tuomas Sandholm in 2019. They introduced Linear CFR (LCFR) and Linear CFR+ (LCFR+). Both of these variants are using alternating updates and linear averaging of current strategies from CFR+. LCFR is using RM, and LCFR+ is using RM+ to compute the current strategy. The difference between CFR(+) and LCFR(+) is that LCFR(+) is also linearly averaging cumulative counterfactual regrets. The authors state that LCFR+ is performing worse than LCFR, so next we will consider just LCFR.

■ 4.3.1 Weighting the Average Strategy

Contribution to the average strategy is weighted by t at iteration t ,

$$\bar{\sigma}_i^t(I, a) = \frac{2}{t(t+1)} \sum_{k=1}^t k \pi_i^{\sigma^k}(I) \sigma_i^k(I, a) \quad (4.4)$$

alternatively, the authors propose to multiply the accumulated strategy by $\frac{t}{t+1}$ at iteration t .

■ 4.3.2 Weighting the Cumulative Counterfactual Regret

LCFR is linearly weighting the contribution to cumulative counterfactual regret:

$$R_i^t(a) = R_i^{t-1}(a) + t \Delta R_i(a) \quad (4.5)$$

also here, at iteration t , the authors propose instead to multiply accumulated cumulative counterfactual regret by $\frac{t}{t+1}$.

■ 4.4 Discounted CFR

Discounted CFR (DCFR) [2][1] also introduced Noam Brown and Tuomas Sandholm in 2019. DCFR generalizes the LCFR's idea of linearly weighting average strategies and cumulative counterfactual regrets to a parametrized polynomial.

DCFR is using three hyperparameters α , β and γ , stylized as $\text{DCFR}(\alpha, \beta, \gamma)$. Positive accumulated cumulative counterfactual regret is multiplied by

$\frac{t^\alpha}{t^{\alpha+1}}$, negative is multiplied by $\frac{t^\beta}{t^{\beta+1}}$ and average strategy¹ by $\frac{t^\gamma}{t^{\gamma+1}}$ at iteration t . DCFR($\alpha = 1, \beta = 1, \gamma = 1$) is equal to LCFR. DCFR($\alpha = 1, \beta = 1, \gamma = 1$) is equal to LCFR. The authors recommend DCFR($\alpha = \frac{3}{2}, \beta = 0, \gamma = 2$), because they claim this parametrization of DCFR performs better than CFR+. Therefore, DCFR was considered state-of-the-art algorithm after CFR+.

¹Although the authors state that one should multiply the average strategy by $(\frac{t}{t+1})^\gamma$, then empirical comparison with LCFR and CFR+ does not yield correct results.

Chapter 5

Experiments

This chapter presents experiments and results of these experiments performed on investigated CFR variants. Section 5.1 describes imperfect-information games, on which we perform two empirical experiments. The first experiment, section 5.2, confirms the correctness of implemented Linear CFR and Discounted CFR. The second experiment, section 5.3, measures speed of convergence to the ϵ -Nash equilibrium on introduced games.

5.1 Games

5.1.1 Goofspiel

Goofspiel is a card game using classical French playing cards. At the beginning of the game, every player receives a different suit of cards A (with the value 1), 2, 3, ..., N , where N is a variable usually up to 13 (which is equal to the K, king, card). We put another suit of cards, so-called prize cards, with the same range between the players with the top card visible for both players. Every round of the game, players observe the top prize card, simultaneously choose one card from their remaining cards. The player with a greater value of the selected card wins the round and receives the prize card's points. The prize card and each player's played card are removed, the new round starts until they have cards left. The winner of the game is the player with the highest accumulated points.

We talk about complete-information Goofspiel when players find out which card every player selected each round. Incomplete-information when they hand them to an umpire who announces only the winner and players don't know what card their opponent used. The prize cards are in ascending, descending, or random order before the first round begins. With binary utilities, the winner gets utility 1, and the loser receives utility -1 . Draw happens when both players accumulate the same amount of points, and both get 0 utility. With scalar utilities, the winner gets the positive difference between points both won, and the loser gets negative.

■ 5.1.2 Liar's Dice

Liar's Dice is an old dice game with various other names like Dudo or Bluff. Player one has available D_1 dices, and player two has D_2 dices, with faces 1, 2, ..., K . Each round, players toss their dices in a way no other player can see their outcomes. Then iterations of betting follow. One player estimates the outcome of all thrown dices in the game as a minimal number of dices with a particular face. The face K works as a wild card and matches any other. The opponent can either raise the face's value with the same minimal number of dices, increase the minimal number of dices, or call the previous player a liar. When a player calls his opponent a liar, players reveal their dices.

The winner gets utility of 1 and the loser -1 . In the original game, the loser would lose one die, and the game would proceed into another round until there is just one player with some remaining number of dices. We will consider only a single round of Liar's Dice.

■ 5.1.3 Oshi-Zumo

Oshi-Zumo is a simultaneous game with a playing field of $2K + 1$ consecutive squares and a stone at K -th position. At the beginning of the game, each player has N coins for making a bet every round. Players choose bets with a minimal amount of coins M . The winner of the round is the player who bets more coins the opponent, the stone moves closer to the opponent, and players cannot put their betted coins back. If both players' bets are equal,

the stone doesn't move. The game ends if one player pushes the stone out of the opponent's side or if some player gets out of coins. Draw happens when the stone ends up at the K -th position at the end of the game. Both players get 0 utility during the draw. The winner is a player who pushes the stone to the opponent's half of the playing field, gets utility 1, and the loser utility -1 .

■ 5.1.4 Darkchess

Darkchess is imperfect-information chess, where a player can see only the opponent's pieces that can capture the by one move.

■ 5.2 Correctness of LCFR and DCFR

The original article [2] introducing Linear CFR and Discounted CFR doesn't contain source code nor pseudo-code for implementing these variants, only includes changes to the standard CFR and CFR+ implementation. This experiment replicates a result from the article and verify the approximative correctness of our implementation intended by authors.

The experiment description Measure average exploitability of average strategy after 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 iterations of CFR, CFR+ with quadratic averaging (Brown), LCFR, DCFR(1.5, 0, 2), DCFR(1.5, 0.5, 2), and DCFR(1.5, $-\infty$, 2) on the perfect-information Goofspiel with 5 cards with scalar utilities and ascending prize cards A, 2, 3, 4, 5.

Figure 5.2 shows our results of this experiment. See Figure 5 in [2] for a comparison with the original article.

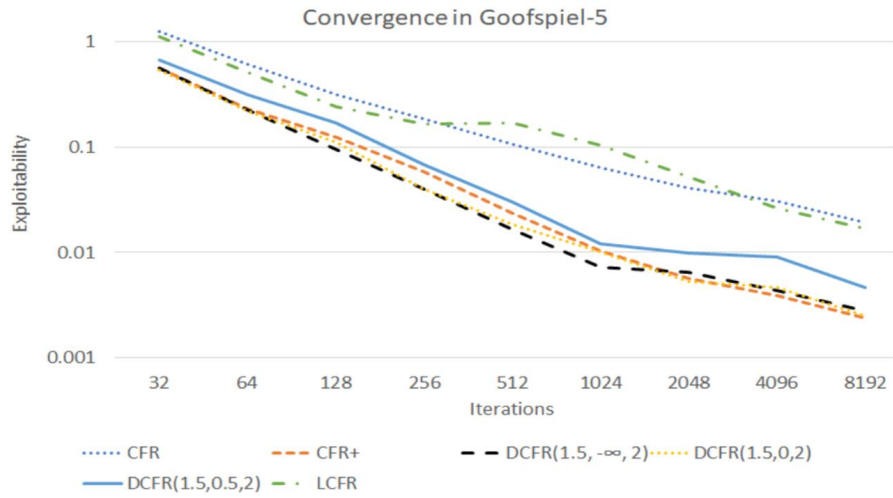


Figure 5.1: Empirical speed of convergence by Noam Brown.

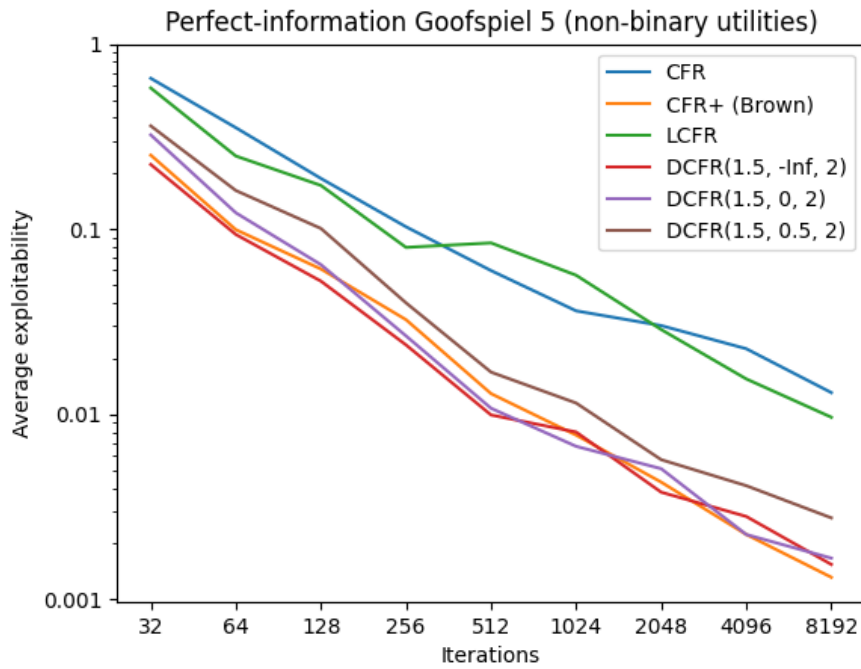


Figure 5.2: Empirical speed of convergence on perfect-information Goofspiel with 5 cards with scalar utilities and ascending prize cards A, 2, 3, 4, 5.

5.3 Empirical Speed of Convergence

Experiment 2 is measuring the empirical speed of convergence to ϵ -Nash equilibrium on introduced games.

■ 5.3.1 Experiment Setup

The empirical speed of convergence experiment compares how many iterations are necessary to approximate Nash equilibria, computing epsilon-Nash equilibria, by CFR based algorithms on imperfect-information or simultaneous games. The average exploitability measures the approximation for the average strategy profile after iteration T . Previous research had considered that 1000 iterations are enough precise approximation. We will measure and plot average exploitability after 32, 64, 128, 256, 512, 1024, 2048, 4096, and 8192 iterations to observe long-term progress. For comparison, we will use CFR, CFR+ (with linear and quadratic averaging of average strategy), LCFR, and DCFR with used parameters, which are $\text{DCFR}(1.5, 0, 2)$, $\text{DCFR}(1.5, 0.5, 2)$, and $\text{DCFR}(1.5, -\infty, 2)$. All algorithms will use standardly used alternating updates. Because all algorithms are deterministic, every convergence curve will correspond to one run of the algorithm.

■ 5.3.2 Empirical Speed of Convergence on Goofspiel

Goofspiel is originally a simultaneous game, so even perfect-information Goofspiel has information sets with a size bigger than one. We compare perfect-information and imperfect-information Goofspiel with 3, 4, and 5 cards. Besides the number of cards, we also combine variants with binary and scalar utilities. Finally, we add combinations with a fixed Chance and randomized Chance. The fixed Chance is the same as in the previous experiment. The next three subsections are describing empirical results according to the number of cards.

■ Goofspiel 3

Goofspiel with three cards (Goofspiel 3) is a relatively small game with less than 1000 information sets and EFG nodes. See Table 5.2 and 5.1, Goofspiel 3 with the randomized Chance player is around six times bigger game in terms of information sets and the number of EFG nodes. The maximal depth of the game tree is 7 (5) for the game with the randomized (fixed) Chance.

Fixed Chance	Depth	No. IS	No. nodes	No. actions
True	5	92	103	138
False	7	546	606	792

Table 5.1: Basic game statistics of perfect-information Goofspiel with 3 cards. The Depth column corresponds to the maximal depth of EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

Fixed Chance	Depth	No. IS	No. nodes	No. actions
True	5	72	103	138
False	7	426	606	792

Table 5.2: Basic game statistics of imperfect-information Goofspiel with 3 cards. The Depth column corresponds to the maximal depth of EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

Imperfect-information Goofspiel 3 has slightly fewer information sets than perfect-information Goofspiel 3. The imperfect-information variant has information sets more spread across the game tree’s width, and their information sets contain more EFG nodes than the perfect-information variant. This show heatmaps in Table 5.3 and 5.4. Almost all information sets in the perfect-information game have size one. On the other hand, the imperfect-information variant has the upper part of the game tree covered with information sets of size more than one.

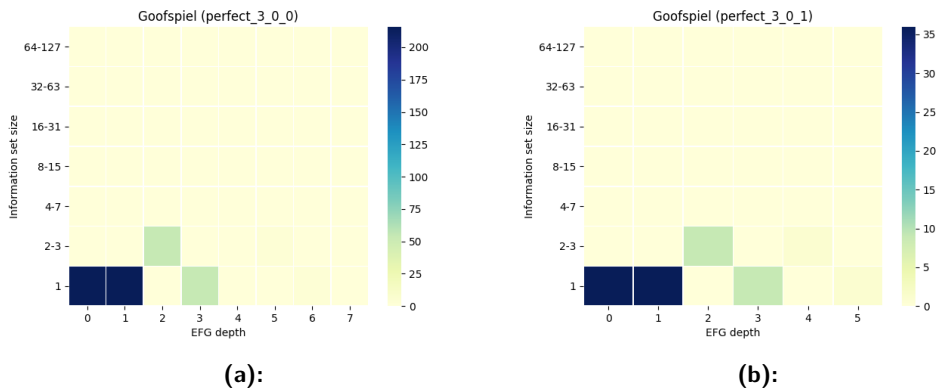


Table 5.3: The table shows heatmaps for perfect-information Goofspiel 3 with randomized Chance (a) and fixed Chance (b). The heatmap’s x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.

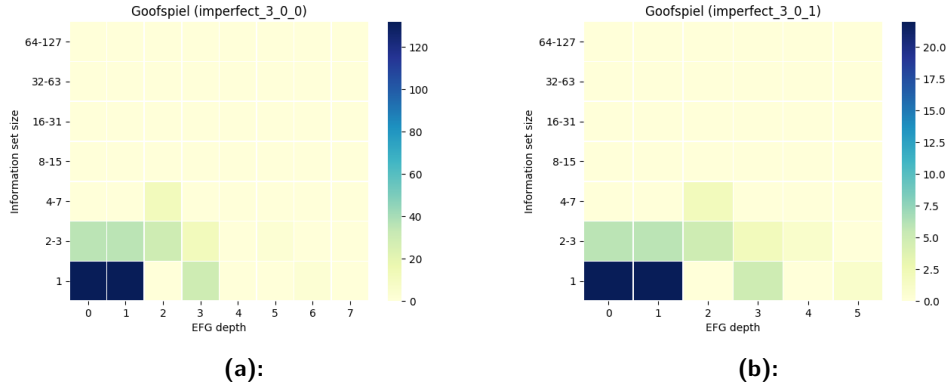


Table 5.4: The table shows heatmaps for imperfect-information Goofspiel 3 with randomized Chance (a) and fixed Chance (b). The heatmap’s x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.

Empirical speed of convergence for perfect-information Goofspiel 3 displays graphs in table 5.5. We can observe that the average exploitability for all tested CFR variants is steadily exponentially declining to zero. All CFR variants tend to cluster into three convergence speeds with about two orders of magnitude distance between each other. The slowest group has only CFR. Then the second group is CFR+ and LCFR. The quickest is a group with all parametrized DCFR and CFR+ with quadratic averaging. If you stop any variant in any iteration, for perfect-information Goofspiel 3, you have a certainty of the order of convergence speed to Nash equilibria.

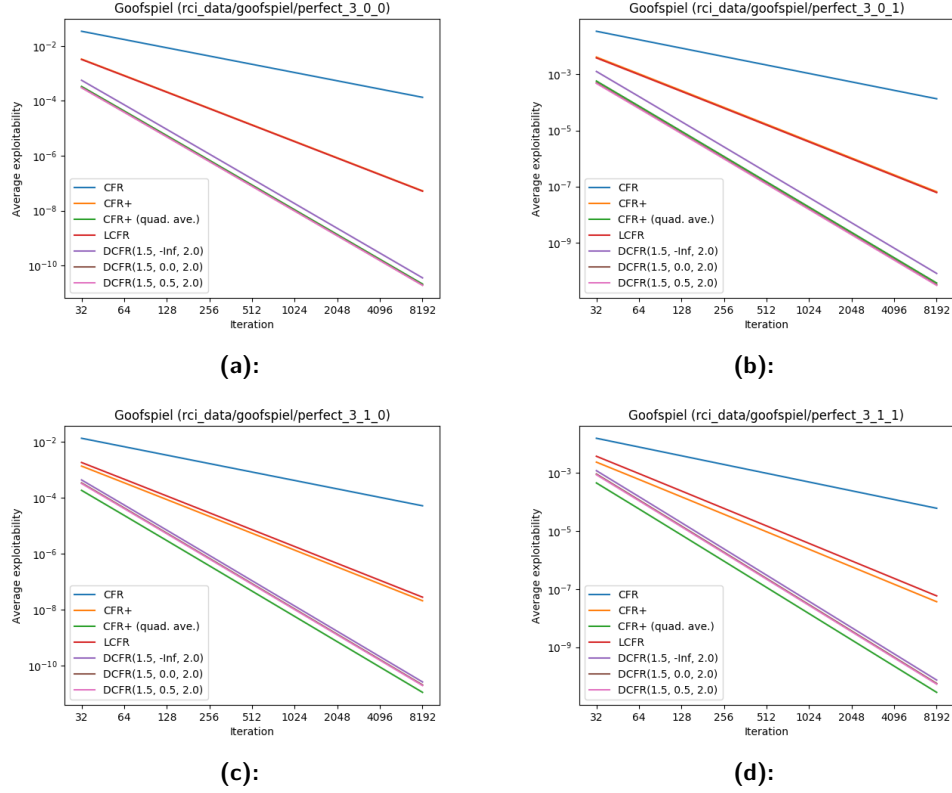
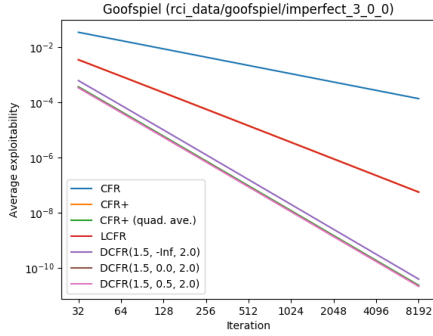


Table 5.5: Speed of convergence on perfect-information Goofspiel 3. The first row contains the game’s variant with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. Similarly, the second row contains the game’s variant with scalar utilities. Figure (c) has randomized Chance, and figure (d) fixed Chance. An x-axis shows the number of iteration. A y-axis is logarithmic and shows average exploitability.

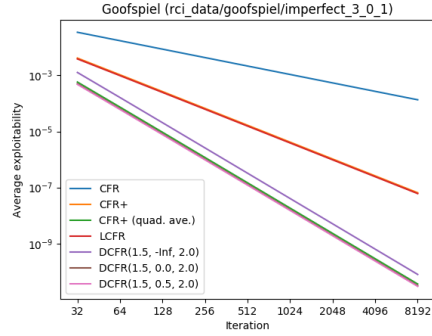
In table 5.5, the first row are figures (a) and (b) where the variants run on perfect-information Goofspiel 3 with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. CFR+ and LCFR converge identical. Also, DCFR(1.5, 0.0, 2.0), DCFR(1.5, 1.5, 2.0) and CFR+ with quad. averaging converge very closely. DCFR(1.5, -Inf, 2.0) is slightly behind the three. On figure (b), all variants converge about one order magnitude slower than on (a). Also, there is a slightly bigger distance between DCFR(1.5, -Inf, 2.0) and other DCFRs and CFR+ with quadratic averaging.

Figures (c) and (d) shows convergence graphs for perfect-information Goofspiel 3 with scalar utilities. Similarly, the figure (c) has a randomized Chance, and the figure (d) has a fixed Chance. We can see that CFR+ with quadratic averaging is slightly ahead of DCFR variants. In the second group, CFR+ is slightly ahead of LCFR. Fixed Chance increasing they lead.

The table 5.6 shows graphs with empirical speed of convergence for imperfect-information Goofspiel 3. All graphs show the same progress as for perfect-information variant.

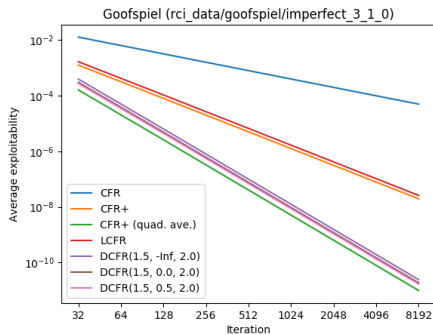


(a) : Figure A

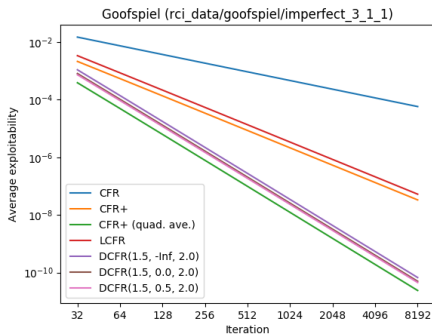


(b) : Figure B

Table 5.6: Speed of convergence on imperfect-information Goofspiel 3. The first row contains the game’s variant with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. Similarly, the second row contains the game’s variant with scalar utilities. Figure (c) has randomized Chance, and figure (d) fixed Chance. An x-axis shows the number of iteration. A y-axis is logarithmic and shows average exploitability.



(a) : Figure C



(b) : Figure D

Table 5.7: Speed of convergence on imperfect-information Goofspiel 3. The first row contains the game’s variant with binary utilities. Figure (a) has a randomized Chance. Figure (b) has a fixed Chance. Similarly, the second row contains the game’s variant with scalar utilities. Figure (c) has randomized Chance, and figure (d) fixed Chance. An x-axis shows the number of iteration. A y-axis is logarithmic and shows average exploitability.

The following tables show statistics of pure and mixed strategies for the slowest and quickest variant in the convergence. For imperfect-information

Goofspiel 3 with scalar utilities and fixed Chance are statistics in the table 5.8 and 5.9. For imperfect-information Goofspiel 3 with scalar utilities and randomized Chance are statistics in the table 5.10 and 5.11. All of them have pure strategy except in depth 4 and 5. In both case, the slower convergence has CFR in tables 5.8 and 5.10. Tables 5.11 and 5.9 are from CFR+ with quadratic averaging. The ration of pure strategies in depth 4 is increased and in depth 5 increased.

Depth	0	1	2	3	4	5
Pure	1	1	2	2	28	28
Mixed	0	0	5	5	0	0
Mixed/All	0.0	0.0	0.71	0.71	0.0	0.0

Table 5.8: The strategy statistics after running CFR for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and fixed Chance. Depth means EFG depth of the game tree. The row Pure shows number of information sets with a pure strategy based of the EFG depth. The row Mixed shows the same for a mixed strategy. The last row show a ratio between mixed strategies compared to all strategies in that depth.

Depth	0	1	2	3	4	5
Pure	1	1	4	1	28	28
Mixed	0	0	3	6	0	0
Mixed/All	0.0	0.0	0.43	0.86	0.0	0.0

Table 5.9: The strategy statistics after running CFR+ with quadratic averaging for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and fixed Chance.

Depth	0	1	2	3	4	5	6	7
Pure	0	3	3	0	12	12	168	168
Mixed	0	0	0	0	30	30	0	0
Mixed/All	0.0	0.0	0.0	0.0	0.71	0.71	0.0	0.0

Table 5.10: The strategy statistics after running CFR for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and randomized Chance.

Depth	0	1	2	3	4	5	6	7
Pure	0	3	3	0	16	10	168	168
Mixed	0	0	0	0	26	32	0	0
Mixed/All	0.0	0.0	0.0	0.0	0.62	0.76	0.0	0.0

Table 5.11: The strategy statistics after running CFR+ with quadratic averaging for 8192 iterations on imperfect-information Goofspiel 3 with binary utilities and randomized Chance.

■ Goofspiel 4

Goofspiel with 4 cards (Goofspiel 4) has a naturally similar structure of information sets depending on if we are dealing with the perfect or imperfect-information variant. Table 5.12 and 5.13 show that games with the fixed Chance have lower thousands of information sets and EFG nodes. In games with randomized Chance, these values increase to tens of thousands. Goofspiel 4 has more information sets from 11 to 40 times while having 16 times more EFG nodes with the fixed Chance and 40 times more EFG nodes with the randomized Chance.

Fixed Chance	Depth	No. IS	No. nodes	No. actions
True	7	1474	1653	2228
False	10	34952	38804	50768

Table 5.12: Basic game statistics of perfect-information Goofspiel with 4 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

Fixed Chance	Depth	No. IS	No. nodes	No. actions
True	7	738	1653	2228
False	10	17432	38804	50768

Table 5.13: Basic game statistics of imperfect-information Goofspiel with 4 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

The size of information sets for the perfect-information game, Table 5.14, is usually 2-3 EFG nodes in the upper half of the game tree, rarely more. The bottom part dominates information sets with one node. For the imperfect-information game, in Table 5.15, the size of information sets varies more. There are more information sets with 4-7 EFG nodes and a few with 8-15 nodes. Although randomized Chance causes an increase in the depth of a game tree, the number of information sets and EFG nodes, and volumes of information sets. The distribution of information sizes is the same across the depth of a game tree.

If you recall from Goofspiel 3, CFR variants tended to cluster into three groups of convergence speed. Also, every convergence curve was nicely exponentially declining. The same phenomenon is apparent only for Goofspiel 4 with binary utilities and fixed Chance. See Table 5.16, Figure (b). The three

groups are CFR and LCFR, being the slowest group, about two orders of magnitude quicker is CFR+. Finally, the fastest convergence group is DCFR variants and CFR+ with quadratic averaging. The order of convergence curves in the fastest group is similar to Goofspiel 3 but not the same. DCFR(1.5, 0.0, 2.0), DCFR(1.5, -Inf, 2.0), and CFR+ with quad. ave. have tight curves, so that the CFR+'s curve is not visible. Unlike in Goofspiel 3, this group's slowest variant with substantial distance DCFR(1.5, 0.5, 2.0).

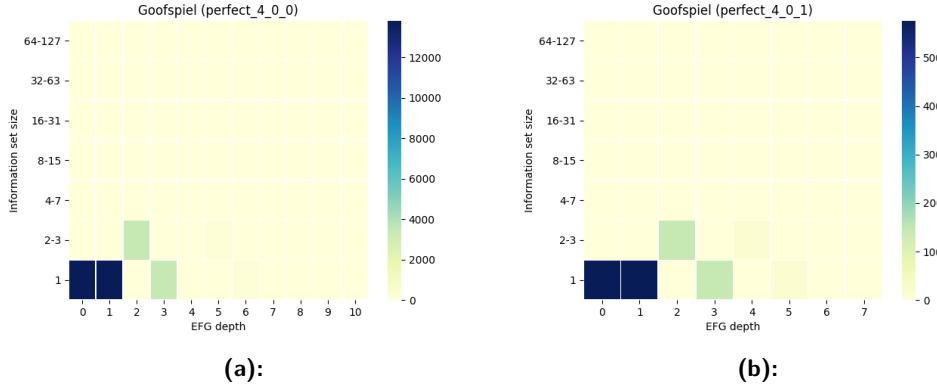


Table 5.14: The table shows heatmaps for perfect-information Goofspiel 4 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.

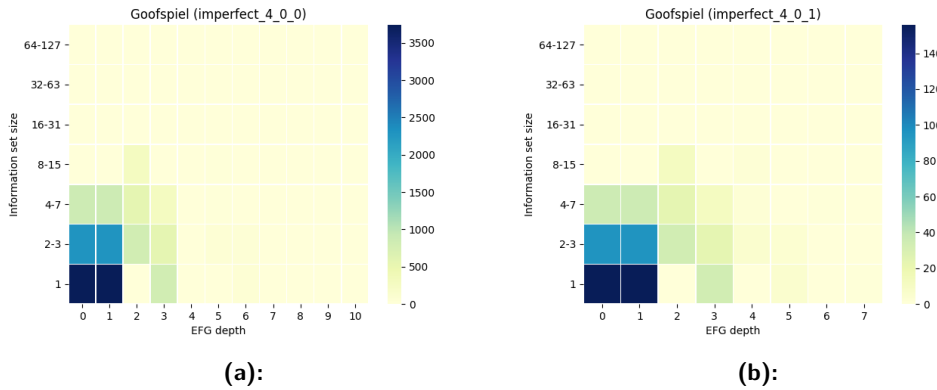


Table 5.15: The table shows heatmaps for imperfect-information Goofspiel 4 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.

The clustering into three convergence groups dissipates into two groups for other parametrization of the perfect-information game. See Table 5.16, Figure (a) with binary utilities, and Table 5.17, Figure(a) with scalar utilities.

They are perfect-information Goofspiel 4 with the randomized Chance.

Figure (a) in Table 5.16 displays that CFR is converging slower but stably than the rest of the CFR variants. The rest of the CFR variants converge at different rates depending on the number of iteration. Previous research usually considered around 1000 iterations as a sufficient number of iterations. For 1024 iterations, DCFR(1.5, -Inf, 2.0) has the fastest convergence. From 1024 to 4096 iterations, DCFR(1.5, -Inf, 2.0) is faster than DCFR(1.5, -Inf, 2.0). LCFR converges the slowest between CFR+ and DCFR variants.

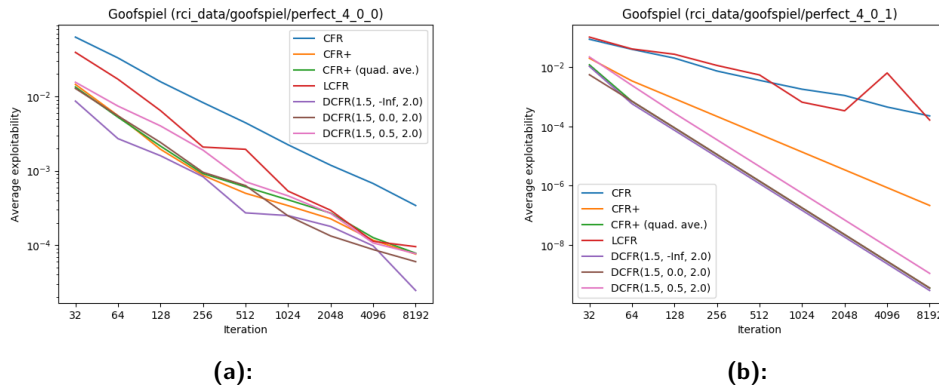


Table 5.16: Speed of convergence on perfect-information Goofspiel 4 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.

Interesting results have CFR variants on perfect-information Goofspiel 4 with scalar utilities and fixed Chance (Figure (b) in Table 5.17). Between iterations 32 and 512, we can observe the mentioned distinction of converging curves into two groups. The slower group contains LCFR and CFR, where LCFR is converging slightly slower than CFR. DCFR(1.5, -Inf, 2.0) converges the fastest for the first 512 iterations very closely to CFR in the quicker group. However, CFR+ takes the lead from 512 iterations onwards. DCFR(1.5, 0.0, 2.0) and DCFR(1.5, 0.5, 2.0) convergence slow down between CFR and CFR+.

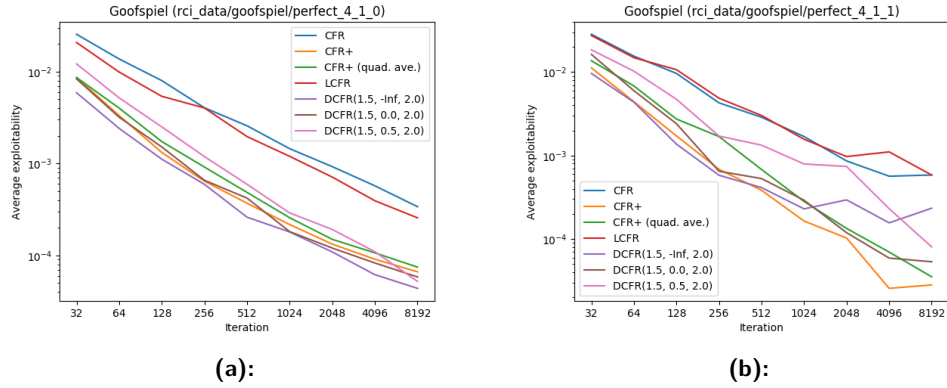
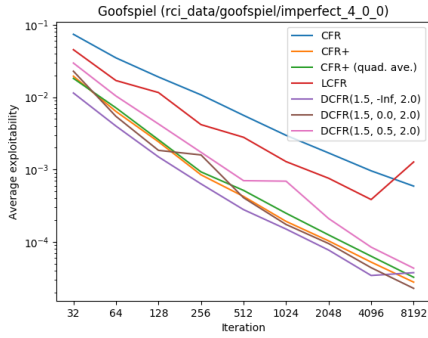
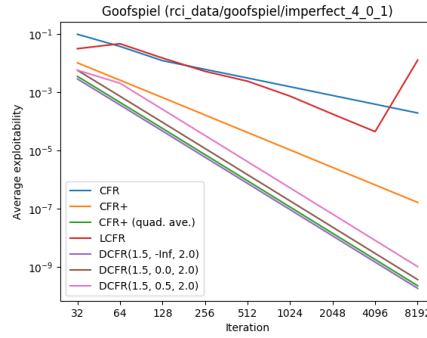


Table 5.17: Speed of convergence on perfect-information Goofspiel 4 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.

Table 5.18 shows the speed of convergence for imperfect-information Goofspiel 4 with binary utilities. Table 5.19 shows the same for the games with scalar utilities. For all parametrization of imperfect-information Goofspiel 4, we can also observe two distinctive groups of convergence curves. The slower group with CFR and LCFR. The quicker group with DCFR and CFR+ variants. In Figure (b), Table 5.18, all convergence curves stable exponentially decline except LCFR. All other game parametrizations have more oscillating convergence curves. In games with binary utilities (Table 5.18), DCFR(1.5, -Inf, 2.0) dominates all other variants. In games with scalar utilities (Table 5.19), DCFR(1.5, -Inf, 2.0) dominates the first 1024 iterations, then convergence speed slows down, and CFR+ with quadratic averaging and DCFR(1.5, 0, 2.0) have lower exploitability.

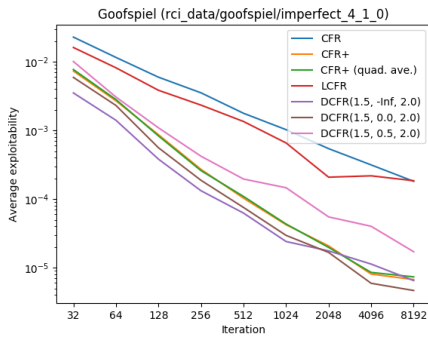


(a):

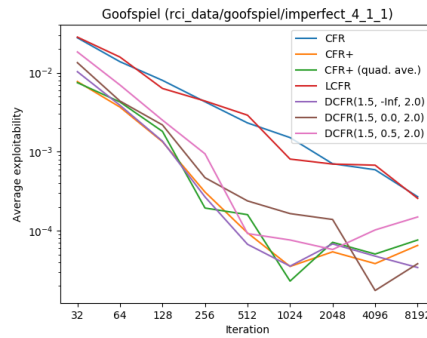


(b):

Table 5.18: Speed of convergence on imperfect-information Goofspiel 4 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.



(a):



(b):

Table 5.19: Speed of convergence on imperfect-information Goofspiel 4 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.

Goofspiel 5

This subsection describes experiments on Goofspiel with five cards. Table 5.20 contains basic statistics about perfect-information Goofspiel 5. The game with fixed Chance has about 40 thousand information sets (about 25 times more than GS 4). The parametrization with randomized Chance has almost 4.5 million information sets, about 125 times more than Goofspiel 4. Imperfect-information Goofspiel 5 (Table 5.21) with fixed Chance has

about 10 thousand information sets (13 times more than the same GS 4). Imperfect-information Goofspiel 5 with randomized Chance has 1.2 million information sets (67 times bigger than the same game with four cards).

Fixed Chance	Depth	No. IS	No. nodes	No. actions
True	9	36852	41331	55730
False	13	4369010	4850530	6346150

Table 5.20: Basic game statistics of perfect-information Goofspiel with 5 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

Fixed Chance	Depth	No. IS	No. nodes	No. actions
True	9	9948	41331	55730
False	13	1175330	4850530	6346150

Table 5.21: Basic game statistics of imperfect-information Goofspiel with 5 cards. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

Distributions of information sets sizes are similar as we have seen in smaller games but with more density. Information sets with the biggest sizes and density resides in the top half of the game tree. The sizes and number of information sets in the bottom part of the tree drastically decrease. Imperfect-information (perfection-information) Goofspiel 5, see Table 5.22 (Table 5.23), has information sets with up to 32-64 (4-7) EFG nodes.

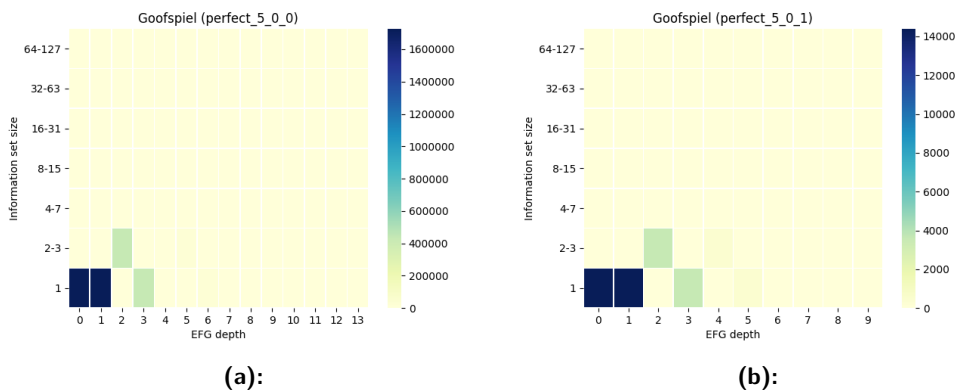


Table 5.22: The table shows heatmaps for perfect-information Goofspiel 5 with randomized Chance (a) and fixed Chance (b). The heatmap's x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.

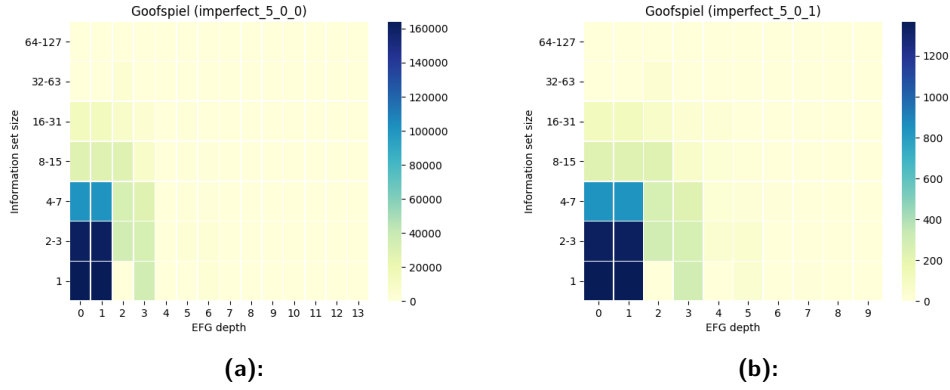


Table 5.23: The table shows heatmaps for imperfect-information Goofspiel 5 with randomized Chance (a) and fixed Chance (b). The heatmap’s x-axis is the EFG depth of the game tree and the y-axis number of EFG nodes in the information sets.

Empirical speed of convergence of perfect-information (Table 5.24 and Table 5.25) Goofspiel 5 and imperfect-information (Table 5.26 and Table 5.27) have clear two clusters of curves. The slowest belong to CFR and LCFR. The quickest group contains both CFR+ variants and DCFR variants. DCFR(1.5, -Inf, 2.0) the most rapid convergence with closely with CFR+. The most slowest convergence with this group has DCFR(1.5, 0.5, 0.0).

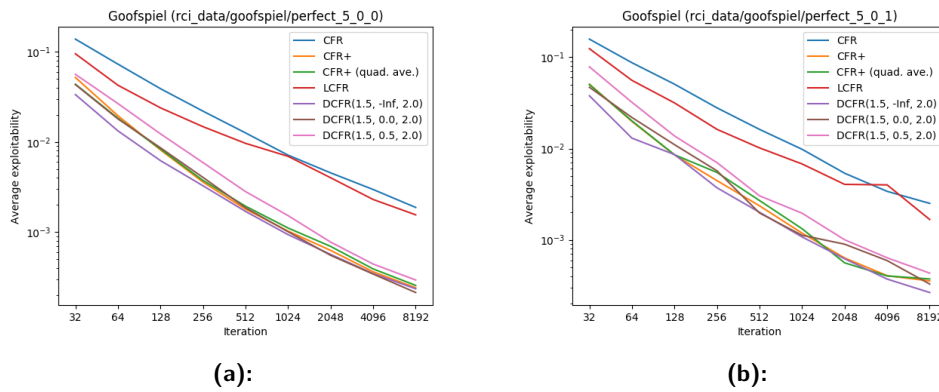
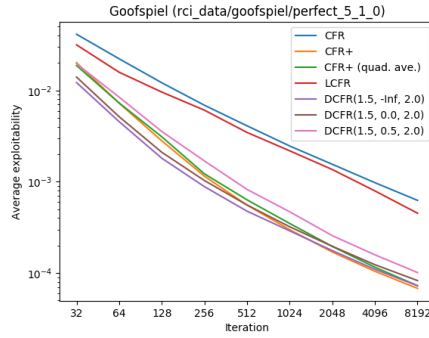
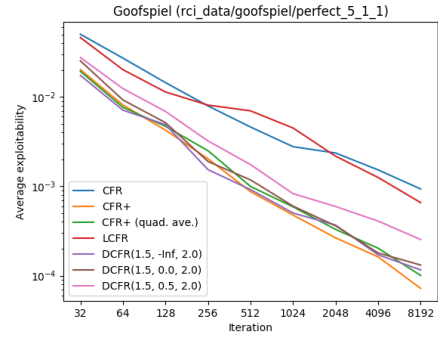


Table 5.24: Speed of convergence on perfect-information Goofspiel 4 with binary utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.



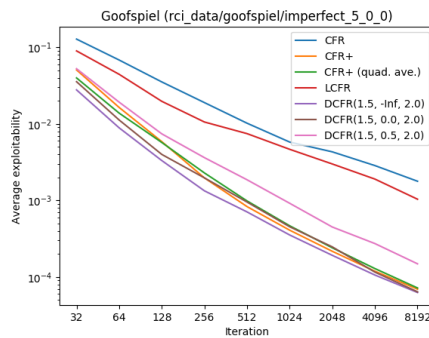
(a):



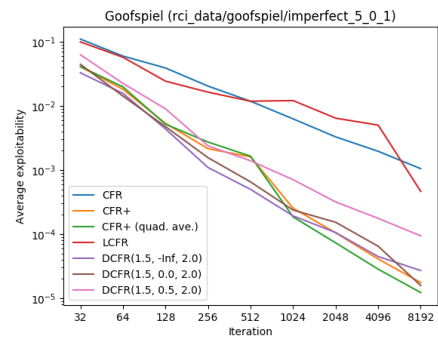
(b):

Table 5.25: Speed of convergence on perfect-information Goofspiel 5 with scalar utilites. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.

Empirical speed of convergence of perfect-information (Table 5.24 and Table 5.25) Goofspiel 5 and imperfect-information (Table 5.26 and Table 5.27) have clear two clusters of curves. The slowest belong to CFR and LCFR. The quickest group contains both CFR+ variants and DCFR variants. DCFR(1.5, -Inf, 2.0) the most rapid convergence with closely with CFR+. The most slowest convergence with this group has DCFR(1.5, 0.5, 0.0).



(a):



(b):

Table 5.26: Speed of convergence on imperfect-information Goofspiel 5 with binary utilites. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.

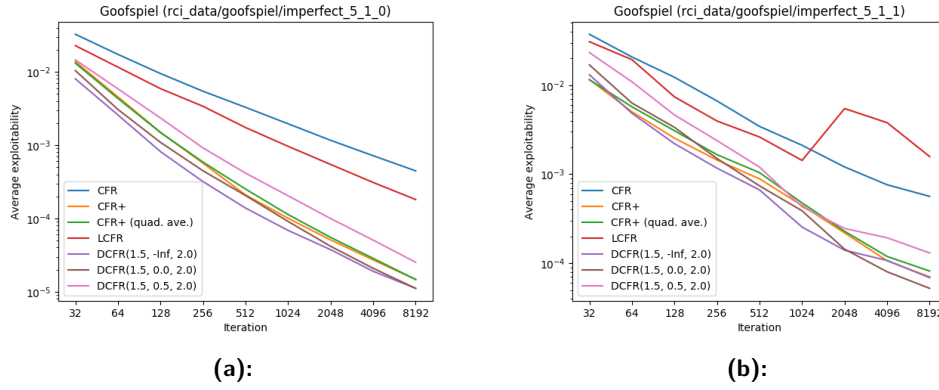


Table 5.27: Speed of convergence on imperfect-information Goofspiel 5 with scalar utilities. Figure (a) shows convergence of CFR variants on the game with the randomized Chance. Figure (b) shows the game with the fixed Chance. An x-axis is the number of iteration. A y-axis shows average exploitability in a logarithmic scale.

Table 5.29 (5.28) shows statistics for the strategy of LCFR (DCFR(1.5, 0.5, 2.0)), which converges the slowest (quickest) in imperfect-information Goofspiel 5 with fixed Chance and scalar utilities. DCFR(1.5, 0.5, 2.0) has less information sets with mixed strategies at depth 4 and 6, but it has more mixed strategies at depth 5.

Depth	0	1	2	3	4	5	6	7	8	9
Pure	0	0	1	0	13	13	257	168	3912	3912
Mixed	1	1	12	13	117	117	661	750	0	0
Mixed/All	1.0	1.0	0.92	1.0	0.9	0.9	0.72	0.82	0.0	0.0

Table 5.28: The strategy statistics after running LCFR with quadratic averaging for 8192 iterations on imperfect-information Goofspiel 5 with binary utilities and fixed Chance.

Depth	0	1	2	3	4	5	6	7	8	9
Pure	0	0	1	0	16	8	270	176	3912	3912
Mixed	1	1	12	13	114	122	648	742	0	0
Mixed/All	1.0	1.0	0.92	1.0	0.88	0.94	0.71	0.81	0.0	0.0

Table 5.29: The strategy statistics after running DCFR(1.5, 0.5, 2) for 8192 iterations on imperfect-information Goofspiel 5 with binary utilities and randomized Chance.

5.3.3 Empirical Speed of Convergence on Liar's Dice

Cards	Depth	No. IS	No. nodes	No. actions
2	5	32	64	120
3	7	192	576	1134
4	9	1024	4096	8160
5	11	5120	25600	51150
6	13	24576	147456	294840

Table 5.30: Basic game statistics of Liar's Dice with 1 dice per player and various number of faces. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

Cards	Depth	No. IS	No. nodes	No. actions
2	9	1024	4096	8160
3	13	36864	331776	663390
4	17	1048576	16777216	33553920

Table 5.31: Basic game statistics of Liar's Dice with 2 dice per player and various number of faces. The Depth column corresponds to the maximal depth of the EFG game tree, No. IS to the number of information sets, No. nodes to the number of EFG nodes, and No. actions to the number of actions.

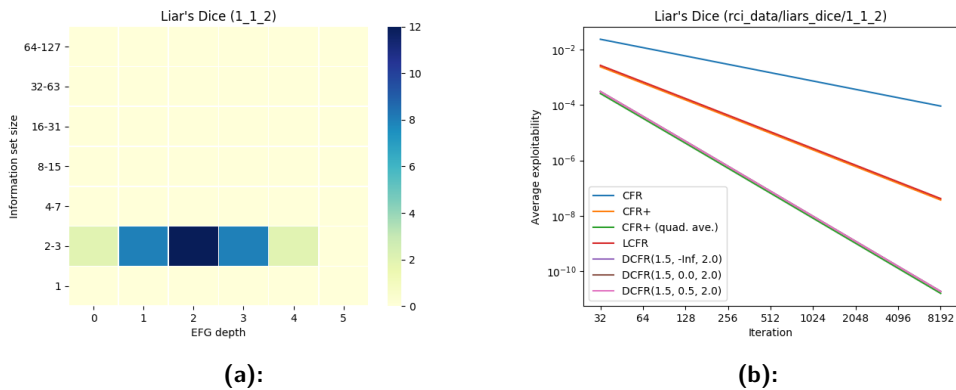
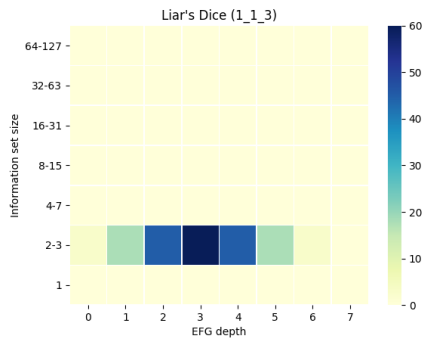
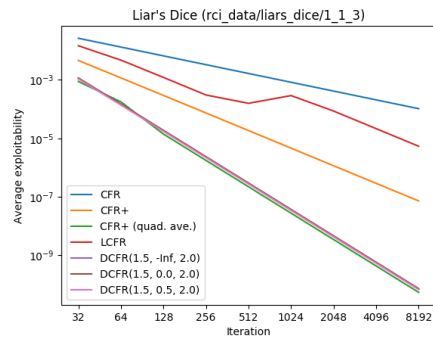


Table 5.32: Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and two faces on each die. DCFR variants and CFR+ with quadratic averaging have similar convergence curves. Figure (a) shows the size of the information set in dependence of EFG depth.

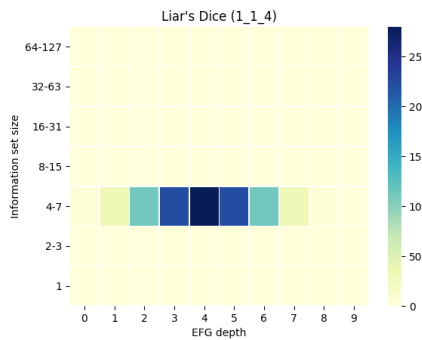


(a):

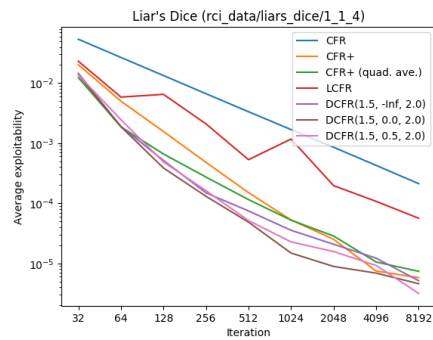


(b):

Table 5.33: Figure (b) shows the speed of convergence on Liar’s Dice with one dice per player and three faces on each die. LCFR and CFR+ have identical convergence curves. Also, DCFR variants and CFR+ with quadratic averaging have similar convergence curves. Figure (a) shows the size of the information set in dependence of EFG depth.



(a):



(b):

Table 5.34: Figure (b) shows the speed of convergence on Liar’s Dice with one dice per player and four faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth.

5. Experiments

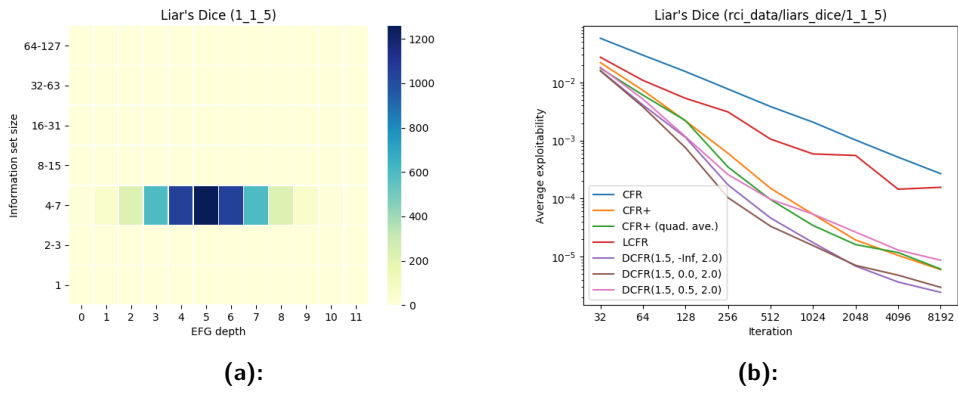


Table 5.35: Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and five faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth.

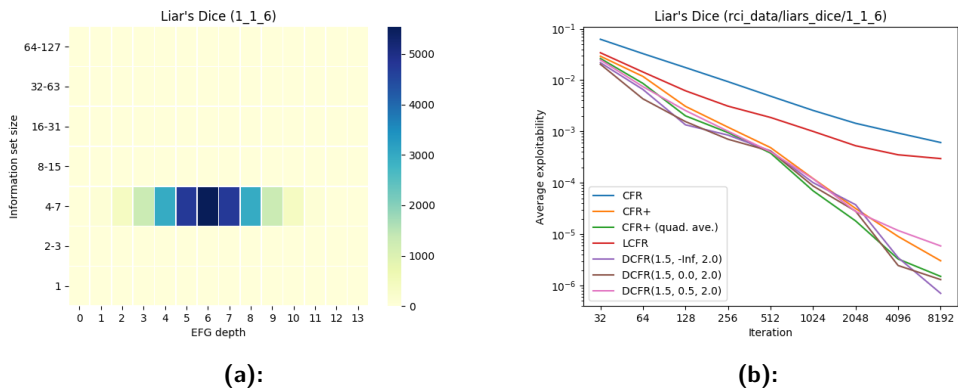


Table 5.36: Figure (b) shows the speed of convergence on Liar's Dice with one dice per player and six faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth.

5.3. Empirical Speed of Convergence

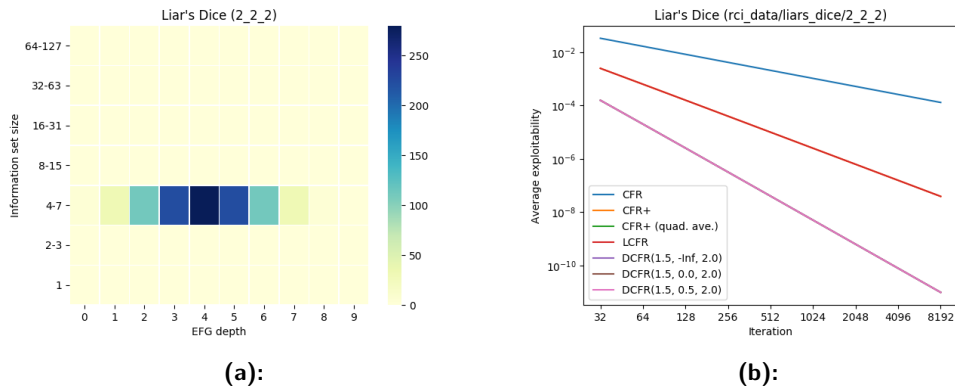


Table 5.37: Figure (b) shows the speed of convergence on Liar’s Dice with two dices per player and two faces on each die. LCFR and CFR+ have identical convergence curves under the red curve. Also, DCFR variants and CFR+ with quadratic averaging have similar convergence curves under the purple curve. Figure (a) shows the size of the information set in dependence of EFG depth.

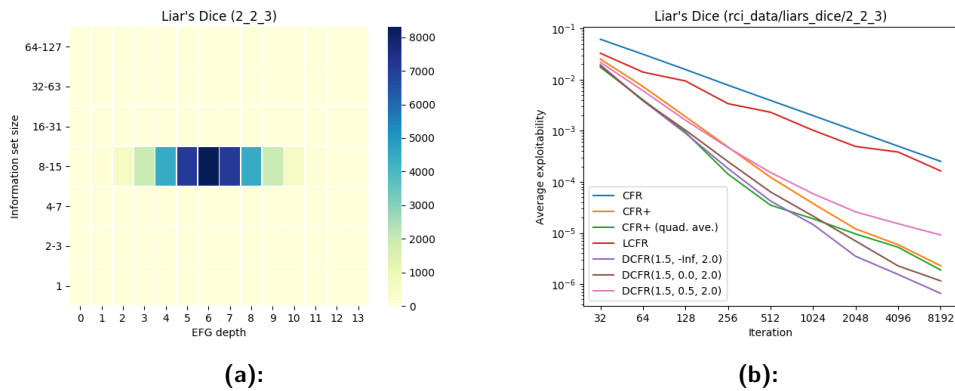
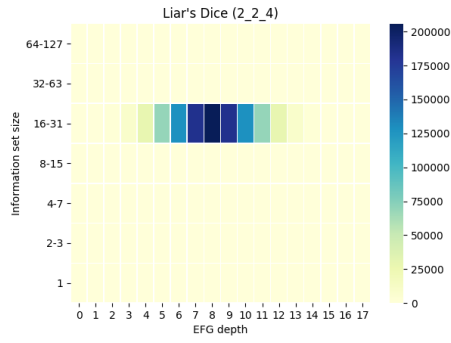
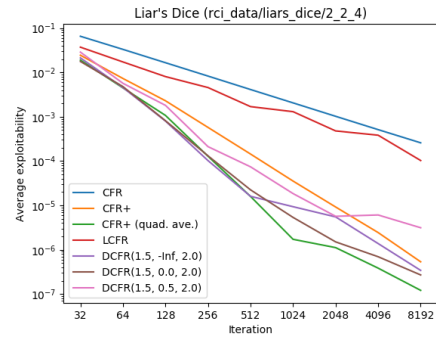


Table 5.38: Figure (b) shows the speed of convergence on Liar’s Dice with one dice per player and three faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth.

5. Experiments



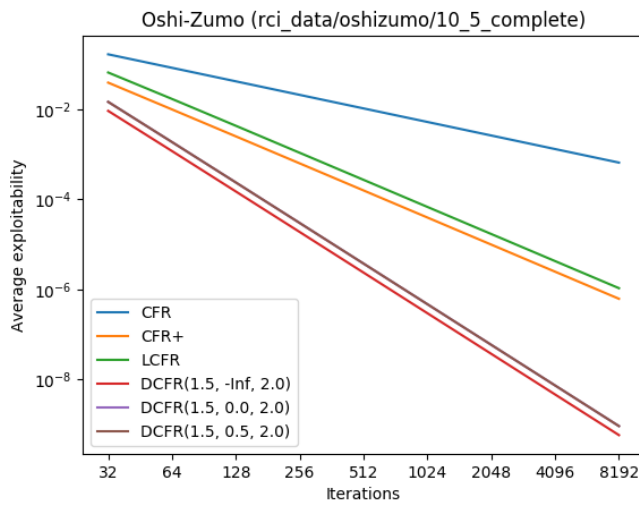
(a):



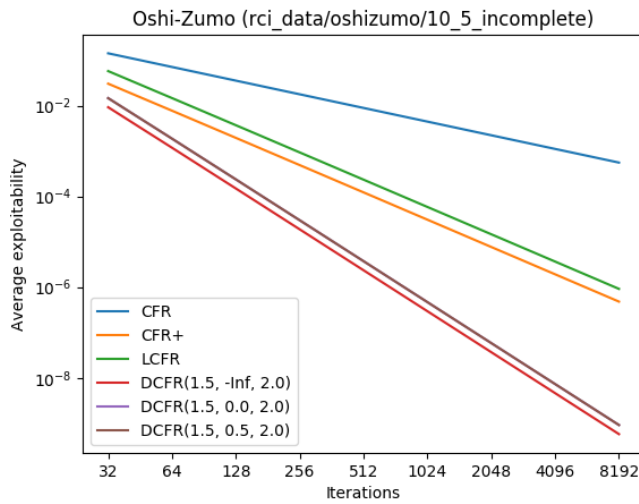
(b):

Table 5.39: Figure (b) shows the speed of convergence on Liar's Dice with two dices per player and four faces on each die. Figure (a) shows the size of the information set in dependence of EFG depth.

5.3.4 Empirical Speed of Convergence on Oshi-zumo



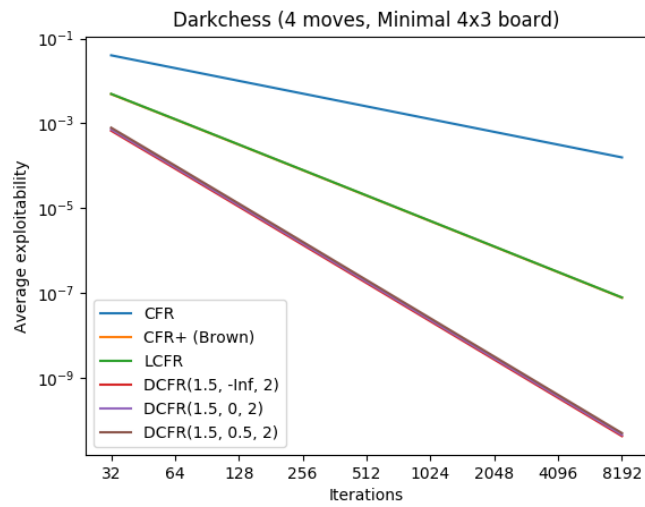
(a) : Figure A



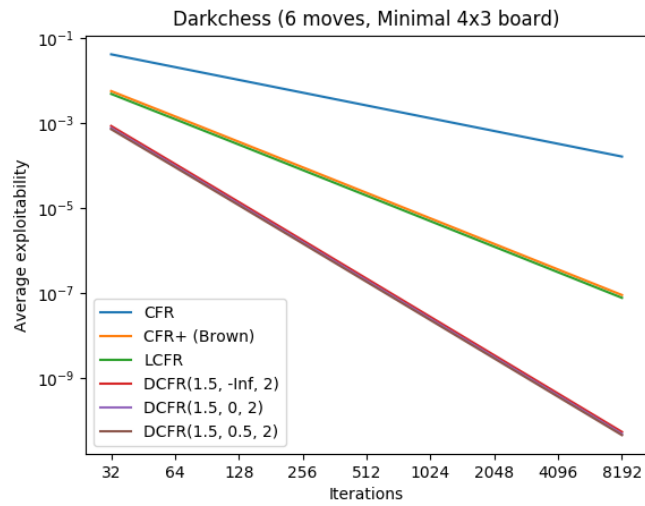
(b) : Figure B

Table 5.40: Speed of convergence on Oshi-Zumo. 5 starting point, 10 coins, 1 min. bid

5.3.5 Empirical Speed of Convergence on Darkchess

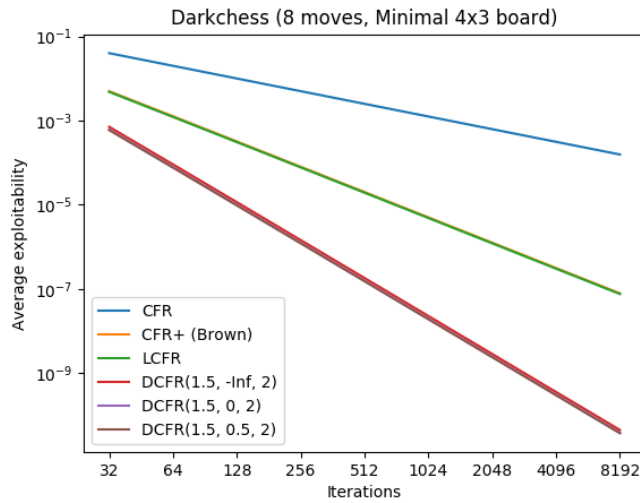


(a) : Figure A



(b) : Figure B

Table 5.41: Speed of convergence on the Darkchess’s minimal 4x3 board. Figure A shows the convergence graph for 2 moves per player. Figure B shows the convergence graph for 3 moves per player.



(a) : Figure A

Table 5.42: Speed of convergence on the Darkchess’s minimal 4x3 board. Figure A shows the convergence graph for 4 moves per player. Figure B shows the convergence graph for 5 moves per player.

5.4 Discussion

In games with less mixed information sets’ strategies, all variants have a clear exponential decline in exploitability. Variants cluster into three groups: 1. the slowest Vanilla CFR, 2. CFR+ and LCFR, 3. DCFR. For games with more mixed information sets’ strategies, differences tend to diminish between groups 2 and 3, also 1 and 2. For groups 2 and 3, CFR+ usually catches up DCFR. Some variants of DCFR are sometimes worse than CFR+. The recommended variant by Noam Brown is not always the best, depends on the domain. Finally, CFR and LCFR tend to have similar results in bigger games. LCFR particularly seems to be unstable after thousands of iterations. Although DCFR performs on average better, the difference between DCFR and CFR+ is either small or none for big domains. For a bad choice of hyperparameters, DCFR could be worse than CFR+. Also, notice with more than the linear average, after some number of iterations, the convergence curves are unstable.



Chapter 6

Conclusion

Section 1 introduced the fundamentals of game theory. Then Section 2 introduced online learning and connection to game theory. Section 3 surveyed CFR, CFR+, LCFR, and DCFR. Section 4 carries the empirical convergence speed on Goofspiel, Liar's Dice, Oshi-Zumo, and Darkchess. Section 4 is also discussing results.

Future work could include testing the new state of the art CFR variant, PCFR, between all other variants. Also, future work could consist of an empirical speed of convergence with PCFR and DCFR weighing.



Bibliography

- [Bro20] Noam Brown. “Equilibrium Finding for Large Adversarial Imperfect-Information Games”. PhD thesis. Carnegie Mellon University, 2020. URL: <http://reports-archive.adm.cs.cmu.edu/anon/2020/CMU-CS-20-132.pdf>.
- [BS19a] Noam Brown and Tuomas Sandholm. “Solving Imperfect-Information Games via Discounted Regret Minimization”. In: *Thirty-Third AAAI Conference on Artificial Intelligence* 33.1 (2019), pp. 1829–1836. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.33011829. eprint: <https://www.aaai.org/ojs/index.php/AAAI/article/view/4007/3885>. URL: <https://www.aaai.org/ojs/index.php/AAAI/article/view/4007>.
- [BS19b] Noam Brown and Tuomas Sandholm. “Superhuman AI for multiplayer poker”. In: *Science* 365.6456 (2019), pp. 885–890. ISSN: 0036-8075. DOI: 10.1126/science.aay2400. eprint: <https://science.sciencemag.org/content/365/6456/885.full.pdf>. URL: <https://science.sciencemag.org/content/365/6456/885>.
- [Bur17] Neil Burch. “Time and Space: Why Imperfect Information Games are Hard”. PhD thesis. University of Alberta, Computing Science, 2-32 Athabasca Hall, Edmonton, Alberta T6G 2E8: University of Alberta, Dec. 2017.
- [CL06] Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. 1st ed. Cambridge University Press, Mar. 2006. ISBN: 0521841089.

- [Sil+18] David Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419 (2018), pp. 1140–1144. ISSN: 0036-8075. DOI: 10.1126/science.aar6404. eprint: <https://science.sciencemag.org/content/362/6419/1140.full.pdf>. URL: <https://science.sciencemag.org/content/362/6419/1140>.
- [SL08] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. 1st ed. Cambridge University Press, Dec. 2008. ISBN: 0521899435.
- [Tam+15] Oskari Tammelin et al. “Solving Heads-Up Limit Texas Hold’em”. In: *IJCAI*. 2015, pp. 645–652. URL: <http://ijcai.org/Abstract/15/097>.
- [Tam14] Oskari Tammelin. *Solving Large Imperfect Information Games Using CFR+*. 2014. arXiv: 1407.5042 [cs.GT].
- [Vin+19] Oriol Vinyals et al. *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>. 2019.
- [Zho+18] Yichi Zhou et al. *Lazy-CFR: fast and near optimal regret minimization for extensive games with imperfect information*. 2018. URL: <https://arxiv.org/abs/1810.04433>.
- [Zin+08] Martin Zinkevich et al. “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt et al. Curran Associates, Inc., 2008, pp. 1729–1736. URL: <http://papers.nips.cc/paper/3306-regret-minimization-in-games-with-incomplete-information.pdf>.