FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

EXTENDED ABSTRACT OF DOCTORAL THESIS

Matej Moravčík

Bridging the Gap: Towards Unified Approach to Perfect and Imperfect Information Games

Department of Applied Mathematics

Supervisor of the doctoral thesis:	Prof. Mgr. Milan Hladík, Ph.D.
Study programme:	Computer Science
Study branch:	Discrete Models and Algorithms

Prague 2023

Title: Bridging the Gap: Towards Unified Approach to Perfect and Imperfect Information Games

Author: Matej Moravčík

Department: Department of Applied Mathematics

Supervisor: Prof. Mgr. Milan Hladík, Ph.D., Department of Applied Mathematics

Abstract: From the onset of AI research, games have played an important part, serving as a benchmark for progress in artificial intelligence. Recent approaches using search in combination with learning from self-play have shown strong performance and the ability to generalize across a wide range of perfect information games. In contrast, the leading algorithms for imperfect information traditionally used a small, abstract version of a game and solved this abstraction in one go. This thesis introduces a chain of improvements for imperfect information algorithms that culminates in two significant milestones that helped bridge the gap between perfect and imperfect information games. The first milestone is DeepStack — the first agent that successfully used a combination of sound search and a learned value function in imperfect information games. This led to the first AI to achieve victory over human professional players in no-limit poker. The second milestone is Player of Games — a universal algorithm that can master both perfect and imperfect information games starting from scratch.

Keywords: game theory, search, imperfect information, games, DeepStack, Player of Games

Contents

1. Introduction

Games have played a key role in AI history, engaging top minds and serving as important benchmarks. They can model a wide range of real-world situations, have well-defined objectives, and the performance of agents can be directly compared with that of humans. Also, games are fun to play and even more fun to do research on.

In this thesis, we will consider only two-player zero-sum games.

1.1 History of AI In Games

From the 1950s to the present, there have been significant developments in algorithms resulting in multiple milestones. Until recently, this development was largely separate for perfect and imperfect information games. In this section, we will first look at the historical development of perfect information games, then at imperfect information games, and finally at algorithms that unify approaches from both areas. Given the extent of the research in this area, the mentioned milestones are by no means exhaustive; we will only look at a few of the most prominent and illustrative results.

1.2 Perfect Information Games

1.2.1 Turing chess

One of the earliest examples was Turochamp , a chess program developed by Alan Turing and David Champernowne in 1948 [Copeland, 2004]. Though it was not executed on a real computer due to its complexity for contemporary machines, it was run manually step by step by Turing himself, showing it could handle a full game against a human. While this was a first attempt, it already had important concepts that would repeatedly appear in later algorithms - namely, search and heuristic evaluation function.

1.2.2 Samuel's Checkers

The next important milestone was Samuel's checkers program [Samuel, 1959]. It improved both the search and the value function. For the search, it used minimax search with alpha-beta pruning — an algorithm that is still used today in top chess engines [The Stockfish Development Team, 2021]. Even more important and interesting are the improvements in the heuristic value function. Instead of hard coding, the function was trained using self-play and machine learning. This was one of the first big successes of machine learning. The final version of the program achieved a strong amateur level in checkers - much better than the author himself.

1.2.3 TD-Gammon

The concept of using machine learning with self-play to learn value was then taken one step further by TD-Gammon, a Backgammon player program developed by Gerald Tesauro [Tesauro, 1995]. It used neural networks to approximate the value function and a TD-update rule borrowed from reinforcement learning, combined again with search. It was one of the first programs approaching the level of top human players in a large game.

1.2.4 Deep Blue

In 1997, almost 50 years after Samuel's checkers program, Deep Blue became the first computer program to defeat a reigning world champion after winning a match against Garry Kasparov [Campbell et al., 2002]. This came after a close defeat in the previous year. The program used alpha-beta search in combination with a sophisticated value function. While the value function was largely hand-crafted, it was tuned using a large database of human games. The program was also accelerated using special chess chips.

1.2.5 AlphaGo

Even after mastering chess, the game of Go remained a long-standing challenge for computer players. Two sources of difficulty hindered classical search approaches. The first was the large branching factor of Go - this made delving deeper into the search tree exponentially harder. The other significant problem was the absence of a known strong value function. AlphaGo solved both these problems with the help of machine learning using deep networks [Silver et al., 2016].

The search was based on Monte Carlo Tree Search (MCTS) [Kocsis and Szepesvári, 2006], previously used successfully for Go, but the space of searched actions was greatly reduced thanks to the use of a policy network that suggested the most promising actions to investigate. The evaluation of positions was composed of a combination of two different approaches. Firstly, there was a value function implemented by a convolutional neural network that took a representation of the board and returned a corresponding value. The other evaluation approach used a fast policy network to quickly unroll the game - simulating the actions of both players until the eventual end of the game. This end value was then returned as a final estimate. A linear combination of both these approaches was then used to provide a single value estimate for MCTS.

To train the agent, a large dataset of human Go games was first used for supervised training of the policy and value functions. These functions were then improved using self-play training.

AlphaGo was able to defeat Lee Sedol, one of the world's best Go players.

1.2.6 AlphaGo Zero

AlphaGo Zero was a successor of AlphaGo [Silver et al., 2017b]. It demonstrated that even such a complex game as Go could be trained in a zero-knowledge fashion - that means without human data, using only self-play, the rules of the game, and minimal prior knowledge. Not only was it able to surpass the performance of the original AlphaGo, but it did so using a simpler and more general algorithm.

In contrast to AlphaGo, value estimation came only from the value network, and both policy and value estimation were trained directly from self-play.

1.2.7 AlphaZero

Most high-performance computer programs were designed to play just a single game. For example, Deep Blue would not be able to play Go or checkers. However, the general architecture of AlphaGo Zero allowed one to simply take the same algorithm without any big modifications and train it to play two more games. The resulting agent — AlphaZero — achieved state-of-the-art performance on chess, Go and shogi, all of this using the same network architecture and almost identical hyper-parameters [Silver et al., 2017a].

1.2.8 Summary

Most successful algorithms for perfect information games share a few common traits. The first is the use of search methods, either minimax or MCTS, which allows for real-time reasoning about complex situations as they occur during gameplay. The second is the utilization of a heuristic value function at the leaves of the search tree. More general approaches that use less expert knowledge also share the use of self-play — a technique from reinforcement learning where an AI agent repeatedly plays games against itself, using the outcomes of these games as learning input. As a result, both the value and the policy can be learned without human input.

1.3 Imperfect Information

There is a small but crucial difference between perfect information games such as chess and Go, and imperfect information games such as poker or rock-paper-scissors. In chess, all necessary information is known by both players. In contrast, poker players don't know what cards their opponents hold. This allows chess players to simply choose the single best action to play an optimal maximin strategy. It's easy to see that this cannot be done in games like rock-paper-scissors or poker. Instead, the player has to act strategically and mix actions to carefully conceal information. Moreover, in contrast to perfect information games, where one can just examine possible future actions, in imperfect information games, the optimal policy also depends on the past actions of the players and their opponent. Because of this, classical approaches solve the whole game at once using some optimization technique.

Let's now consider some examples of these games, alongside the classical techniques used.

1.3.1 Matrix Games

Matrix games, also known as normal form games, represent the simplest form of imperfect information games. They depict a situation where all players make decisions simultaneously. In the case of two-player, zero-sum games, such a game can be described by a single payoff matrix. The possible actions for Player 1 are to choose a row from the matrix, while Player 2 must choose a column. The value of the corresponding matrix element is then equal to the utility of Player 1 at the end of the game. Since we are considering zero-sum games, this value also corresponds to the negative utility of Player 2. A simple example of such a game is rock-paper-scissors. Note that actions can be stochastic — each player can choose a probability distribution over their actions.

In 1928, Von Neumann developed the minimax theorem, which has become a foundational principle of game theory, demonstrating what the optimal solution of matrix games looks like [Neumann, 1928]. In 1951, Dantzig showed the equivalence of zero-sum games and linear programs [Dantzig, 1951]. This allows for efficient solutions of large normal form games, using any available LP solver.

1.3.2 Sequential Decision Making

Extensive form Games

In many real-world situations, players do not act simultaneously but instead take sequences of actions. This is the case for the majority of board games, including Checkers, Chess, Go, and Poker. The extensive form game formalism represents all possible action sequences using a game tree. The leaves of the tree correspond to terminal states where the game's terminal utility is defined. Nodes in the tree represent decision points for the players and the edges represent players' actions. If the game involves imperfect information, as in Poker, a player must apply the same strategy in all states that he cannot distinguish between. These states form an information set. If there's a stochastic element in the game, such as dealing cards in Poker, an additional player called a ”chance player” is introduced to model this stochasticity. This player acts according to a known fixed probability distribution.

Counterfactual Regret Minimization

Recently, most of the successful solving techniques for large extensive form games have been using some version of Counterfactual Regret Minimization (CFR) [Zinkevich et al., 2007]. It is an iterative algorithm that operates directly on information sets, thus requiring several orders of magnitude less memory. In each iteration, both players update their strategies, and the average of these strategies provably converges to a Nash equilibrium. The algorithm can be stopped at any time.

1.3.3 Computer Poker

Poker is the canonical game of imperfect information where players cannot see their opponent's cards.Strong play involves bluffing and insights into potential opponent strategies, qualities that have traditionally not been considered computer-like. In their groundbreaking work ”Theory of Games and Economic Behavior,” von Neumann and Morgenstern dedicated an entire section (over 30 pages) to poker [Morgenstern and Von Neumann, 1953].

Over the years, there has been a substantial body of research in imperfect information games, with poker game variants being the only domain used for evaluating the algorithms.

1.3.4 Annual Computer Poker Competition

The Annual Computer Poker Competition [Bard et al., 2013] was started in 2006 as an effort to develop a system to evaluate poker agents that were being developed by the University of Alberta and Carnegie Mellon University. It has been held annually since 2006 until 2018, open to all competitors, in conjunction with top-tier artificial intelligence conferences: AAAI and IJCAI. Multiple university teams and individuals participated each year, submitting dozens of poker agents.

1.3.5 Game Abstraction

Classical solution approaches for imperfect information games require reasoning about the entire game tree at once and producing a complete strategy prior to play. Since a lot of poker variants, like Heads-Up No Limit Texas Hold'em, are too large to be solved directly, the common technique is to solve a smaller, abstracted game that is similar to the original game. To play the original game, one must first translate actions from the original game to the abstracted game, then choose an action based on the abstracted game's policy, and finally translate this action back to the original game. This entire process is called game abstraction.

The majority of the top Annual Computer Poker Competition entries used game abstraction along with counterfactual regret minimization.

1.3.6 Success of Classical Techniques

The combination of the counterfactual regret minimization and abstraction resulted in important milestones for imperfect information games.

Polaris

In 2007 and 2008, the Computer Poker Research Group at the University of Alberta organized the Man-vs-Machine Poker Championships, using the game of Heads-Up Limit Texas Hold'em [Bowling et al., 2009]. In 2007, a poker agent named Polaris competed against human professional players but lost narrowly. In 2008, the improved versions of Polaris narrowly won. This was the first time that a poker-playing AI defeated human professionals.

Cepheus

In January of 2015, the poker agent Cepheus reached another milestone by essentially solving the entire game of Heads-Up Limit Texas Hold'em [Bowling et al., 2015]. While other large games, such as checkers or Connect Four, had been solved previously, this was the first time that any large imperfect information game played professionally by humans was solved.

1.3.7 Limitations of Classical Techniques

While abstraction techniques were very successful in Limit Heads-Up Texas Hold'em poker, their success in No-Limit Texas Hold'em poker, a more complex but also more popular version of poker, was modest. In 2015, the abstraction-based computer program Claudico lost to a team of professional poker players in a No-Limit Texas Hold'em poker match by a margin of 91 mbb/g, which is considered a 'huge margin of victory' [Moravčík et al., 2017]. Furthermore, the local best-response technique showed that abstraction-based programs from the Annual Computer Poker Competition have massive flaws, and moreover, these flaws are relatively easy to find. All evaluated abstraction-based programs lost by at least 3,000 mbb/g against the local best response, which is four times more than if they had simply folded each game [Lisý and Bowling, 2017b]. To illustrate the naivety of the abstraction-based approach for use in large extensive games, one can imagine its application to chess. It would require constructing an abstracted version of chess small enough to solve directly and then mapping states and actions between this abstraction and the original game [Schmid, 2021].

1.3.8 Search Based Techniques

In perfect information, the combination of decision-time search with a heuristic value function leads to strong performance. Some popular perfect information search methods, like Monte Carlo tree search can be also used in imperfect information settings [Whitehouse, 2014]. Unfortunately they fail to produce optimal policies even in very small games. Until recently, it had been even thought that sound search is impossible in imperfect information games[Frank et al., 1998, Lisý et al., 2015]. Fortunately, a significant milestone in computational game theory has been reached recently — a sound search in imperfect information games.

DeepStack

DeepStack was the first algorithm to introduce the combination of sound decision-time search and heuristic value function for imperfect information games [Moravčík et al., 2017]. This was made possible by a technique called continual-resolving. It is analogous to the search in perfect information games, but with a few important modifications that make it theoretically sound. The lookahead tree contains not only states in which the player is acting, but also all the states that are sharing the same publicly known information. This allows for coordination of the policy between states that either player cannot distinguish. In poker, this means that the search always takes into consideration all possible private cards a player and his opponent could hold. To reason within this complex search tree, the value function also has to be more intricate. In contrast to perfect information games, it outputs a vector of values for each player. The root of the lookahead tree is also significantly modified—it forms a gadget game. This gadget ensures that if the policy is optimal in the lookahead tree, it is also optimal in the whole game. The search algorithm must be able to compute a precise stochastic policy, therefore DeepStack uses a version of CFR instead of simple minimax. DeepStack's value function was implemented by a neural network and trained using a large number of examples generated from random poker situations. Continual resolving allowed DeepStack to ditch the abstraction and reason about situations independently as they arise during play, which led to a significant improvement over prior methods. In December of 2016, DeepStack became the first program to beat professional human players in no-limit Texas hold'em poker. In contrast to previous abstraction-based agents, DeepStack is unexploitable by the local best response.

Libratus

Subsequent to DeepStack, the computer program Libratus defeated a team of four professional heads-up poker specialists in a HUNL competition held in January 2017 [Brown and Sandholm, 2018]. Libratus could be described as a hybrid approach. Near the end of the game, it used a 'nested endgame solving' technique similar to the continuous re-solving used by DeepStack. Since it did not utilize a value function, it couldn't execute the search in the early stages of the game and instead used classical abstraction techniques. The use of abstraction resulted in weaknesses in the strategy. To address this, the abstract strategy was augmented with the help of human analysis during match breaks.

Both DeepStack and Libratus demonstrated that real-time decision-making is crucial to achieving high-level performance.

Player of Games

Even after the introduction of decision time search, the worlds of perfect and imperfect games remained separate. Imperfect information agents were usually designed to handle just a single specific game. The Player of Games (PoG) bridges this gap [Schmid et al., 2021]. It was the first algorithm to achieve strong empirical performance in large perfect information games — chess and Go, as well as in imperfect information games — poker and Scotland Yard. This marked an important step toward creating general algorithms for arbitrary environments. The algorithm combines ideas from DeepStack and AlphaZero in a theoretically sound fashion. Continual-resolving introduced by DeepStack is used to ensure that the policy is consistent during online play. Growing-tree counterfactual regret minimization (GT-CFR) builds a lookahead tree non-uniformly, expanding the tree toward the most relevant states similarly to the MCTS used by AlphaZero. Sound self-play is used to train the policy and value network. Both networks are specified in a ”zero-like” fashion with minimal domain-specific knowledge

Limitations of Sound Search

The main limitation of the sound search used by DeepStack and Player of Games is the need to enumerate all possible information states contained in a public state, which can be prohibitively expensive for some games.

This could be an interesting area for future research; one possible solution is to use sampling of the information states.

1.4 Author’s Contribution

The remaining sections of this Ph.D. thesis delve into my contributions to the field of algorithmic game theory. These contributions can be broadly categorized into two main areas: theoretical advancements and novel algorithms.

1.4.1 Theoretical Advancements

Revisiting CFR+ and Alternating Updates

Many successful imperfect information game agents, such as Polaris, Libratus, DeepStack, and Player of Games, leverage a variant of the Counterfactual Regret Minimization (CFR) algorithm. A recent and widely adopted version of CFR is CFR+ due to its superior empirical performance across various problem domains, making it one of the key factors contributing to the success of Cepheus [Burch, 2017]. Although CFR+ was initially introduced with a theoretical upper bound on solution error, subsequent research revealed an error in one of the proof steps [Farina et al., 2019]. We provide updated proofs to recover the original bound [Burch et al., 2019].

Bounding the Support Size in Extensive Form Games with Imperfect Information

Optimal play in imperfect information games often necessitates the use of stochastic policies. This stands in contrast to perfect information games, where a simple deterministic optimal strategy always exists. Naturally, one may wonder about the impact on the number of optimal actions as the level of uncertainty increases.

We have established a linear relationship between the level of uncertainty and the support size, which refers to the number of actions with non-zero probability [Schmid et al., 2014].

Sound Algorithms in Imperfect Information Games

The concept of Nash equilibrium is traditionally defined for a fixed offline set of strategies. However, extending this concept to online settings, where the entire strategy is not computed in advance, is not a straightforward task. Naively attempting to do so may result in the disappearance of certain guarantees for two-player zero-sum games. To tackle this issue, we introduced the concept of a consistency hierarchy, which enables the analysis of algorithms that perform online search, such as those employed in DeepStack or Player of Games [Šustr et al., 2020].

1.4.2 Novel Algorithms

Variance Reduction in Monte Carlo Counterfactual Regret Minimization for Extensive Form Games Using Baselines

Monte Carlo Counterfactual Regret Minimization (MCCFR) [Lanctot et al., 2009] is a family of game-solving algorithms designed for imperfect information games. In contrast to the vanilla CFR implementation, MCCFR doesn't require traversing the entire game tree in each iteration. Instead, it samples a limited number of trajectories, similar to algorithms used in reinforcement learning. While MCCFR still offers good probabilistic convergence guarantees, the introduction of sampling introduces variance in value estimates, which can considerably slow down the convergence speed [Burch, 2017].

In the realm of reinforcement learning, this variance issue has traditionally been addressed using baselines in policy-based methods. In VR-MCCFR (Variance-Reduced MCCFR) [Schmid et al., 2019], we employed similar ideas to obtain unbiased value estimates and reduce variance. In the ideal scenario of perfect estimates, the variance can be reduced to zero. In experimental evaluations, VR-MCCFR achieved an order of magnitude speedup and decreased empirical variance by three orders of magnitude.

Refining Subgames in Large Imperfect Information Games

Traditionally, state-of-the-art game algorithms have employed an abstraction approach, where they solve a smaller, abstracted version of the game and then map the strategy from this reduced game back to the original game at decision time. However, to enable online improvement of strategies, particularly in situations close to the game end where the problem is more tractable, we introduced the concept of safe refinement of sub-games [Moravčík et al., 2016]. The ideas from this work have since been utilized by Libratus, DeepStack, and Player of Games.

AIVAT: A New Variance Reduction Technique for Agent Evaluation in Imperfect Information Games

Evaluation of agents in imperfect information games is inherently noisy, especially when compared to perfect information games. Traditionally, evaluating agents in computer poker competitions required millions of matches to obtain statistically significant results. While this approach worked for simple, abstraction-based agents that don't require complex computation at run-time, it becomes computationally expensive when agents employ decision-time search. Furthermore, comparing agent performance to human players exacerbates the problem. To address these challenges, we developed AIVAT, a provably unbiased method that significantly reduces variance during evaluation [Burch et al., 2018].

Deepstack: Expert-level artificial intelligence in heads-up no-limit poker

DeepStack was the first algorithm to use a theoretically sound combination of limited search depth and machine learning for imperfect information games [Moravčík et al., 2017]. This approach reduced the gap between approaches for perfect and imperfect information. Significantly, it was also the first AI system to outperform professional poker players in No-Limit Texas Hold'em, representing a major accomplishment in the field of AI.

Player of Games

The Player of Games represents the culmination of our efforts to unify the domains of perfect and imperfect information games [Schmid et al., 2021]. It synthesizes techniques employed in both DeepStack and AlphaZero, demonstrating strong empirical performance in both types of games. This achievement is a significant step towards developing truly general algorithms for arbitrary environments. The approach gradually builds a search tree, similar to Monte Carlo Tree Search, and learns from self-play with minimal prior information, similar to AlphaZero. At the same time, it incorporates sound game theoretic reasoning, akin to DeepStack. We have proved that the Player of Games is theoretically sound, and have evaluated its performance in two perfect information games — chess and Go - and two imperfect information games — Heads-Up No-Limit Texas Hold'em Poker and Scotland Yard.

2. Revisiting CFR+ and Alternating Updates

CFR$^+$ was introduced [Tammelin, 2014] as an algorithm for approximately solving imperfect information games, and was subsequently used to essentially solve the game of heads-up limit Texas Hold'em poker [Bowling et al., 2015]. Another paper associated with the poker result gives a correctness proof for CFR$^+$, showing that approximation error approaches zero [Tammelin et al., 2015].

CFR$^+$ is a variant of the CFR algorithm [Zinkevich et al., 2007], with much better empirical performance than CFR. One of the CFR$^+$ changes is switching from simultaneous updates to alternately updating a single player at a time. A crucial step in proving the correctness of both CFR and CFR$^+$ is linking regret, a hindsight measurement of performance, to exploitability, a measurement of the solution quality.

Later work pointed out a problem with the CFR$^+$ proof [Farina et al., 2019], noting that the CFR$^+$ proof makes reference to a folk theorem making the necessary link between regret and exploitability, but fails to satisfy the theorem's requirements due to the use of alternating updates in CFR$^+$. Farina[Farina et al., 2019] give an example of a sequence of updates which lead to zero regret for both players, but high exploitability.

We state a version of the folk theorem that links alternating update regret and exploitability, with an additional term in the exploitability bound relating to strategy improvement. By proving that CFR and CFR$^+$ generate improved strategies, we can give a new correctness proof for CFR$^+$, recovering the original bound on approximation error.

With a corrected proof, we once again have a theoretical guarantee of correctness to fall back on, and can safely use CFR$^+$ with alternating updates, in search of its strong empirical performance without worrying that it might be worse than CFR.

The alternating update analogue of the folk theorem also provides some theoretical motivation for the empirically observed benefit of using alternating updates. Exploitability is now bounded by the regret minus the average improvement in expected values. While we proved that the improvement is guaranteed to be non-negative for CFR and CFR$^+$, we would generally expect non-zero improvement on average, with a corresponding reduction in the bound on exploitability.

3. Bounding the Support Size in Extensive Form Games with Imperfect Information

Arguably the most important solution concept in non-cooperative games is the notion of Nash equilibrium, where no player improves by deviating from this strategy profile. Support is defined as the set of actions played with non-zero probability and there are many crucial implications related to it.

Once the support is known, it is easy to compute the equilibrium in polynomial time even for general-sum games. Performance of some algorithms, namely the double-oracle algorithm for extensive form games, is tightly bound to the size of the support [Bošanský et al., 2013]. Other work shows that minimizing the support in abstracted games can lead to better strategies in the original game [Ganzfried et al., 2012]. Finally, it is advantageous to prefer strategies having a small support. Such strategies are both easier to store and play.

Extensive form games model a wide class of games with a varying levels of uncertainty. In the case of perfect information, there is an optimal strategy using only one action in any information set. In contrast, in some extensive games with imperfect information, the player can be forced to use all the possible actions to play optimally.

In this chapter, we focus on the relation between the level of uncertainty and the support size. We present an upper bound for the support size based on the uncertainty level.

Some games, such as Bayesian extensive games with observable actions or card games (such as no-limit Texas hold'em poker) have most of the information about the current state observable by all players, and therefore a low level of uncertainty. In these games, our bound guarantees the existence of Nash equilibrium having the support size considerably smaller than the number of all possible actions.

Instead of explicitly defining a level of uncertainty, we use the concept of the *public tree*. This concept provides a nice interpretation of uncertainty and *public actions*. Using the public tree, we present a new technique called the *equilibrium preserving transformation*, which transforms some equilibrium strategy profile into another. We provide an upper bound on the number of public actions used in the transformed Nash equilibrium.

Our approach also applies to games with non-observable actions, where it simply limits the number of public actions.

Applying our result to specific games, we present a new bound for the support size in these games.

For example, in no-limit Texas hold'em poker, there can be any finite number of actions available in some information sets. Our result implies the existence of an optimal strategy for which the number of actions used in every information set depends only on the number of players and the number of card combinations players can be dealt.

In Bayesian extensive games with observable actions, the bound equals to the number of different player types the chance can reveal.

Moreover, our proof is constructive. Given an extensive form game and an optimal strategy, the equilibrium preserving transformation finds another optimal strategy satisfying our bound in polynomial time.

4. Sound Algorithms in Imperfect Information Games

From the very dawn of computer game research, search was a fundamental component of many algorithms. Turing's chess algorithm from 1950 was able to think two moves ahead [Copeland, 2004], and Shannon's work on chess from 1950 includes an extensive section on how an evaluation function can be used within search [Shannon, 1950]. Samuel's checkers algorithm from 1959 already combines search and learning of a value function, approximated through a self-play method and bootstrapping [Samuel, 1959]. The combination of search and learning has been a crucial component in the remarkable milestones where computers outperformed their human counterparts in challenging games: DeepBlue in Chess [Campbell et al., 2002], AlphaGo in Go [Silver et al., 2016], DeepStack and Libratus in Poker [Moravčík et al., 2017, Brown and Sandholm, 2018].

Online methods for approximating Nash equilibria in sequential imperfect information games appeared only in the last few years [Lisý et al., 2015, Brown and Sandholm, 2017, Moravčík et al., 2017, Brown and Sandholm, 2018, 2019, Brown et al., 2020]. We thus investigate what it takes for an online algorithm to be sound in imperfect information settings. While it has been known that search with imperfect information is more challenging than with perfect information [Frank and Basin, 1998, Lisý et al., 2015], the problem is more complex than previously thought. Online algorithms "live" in a fundamentally different setting, and they need to be evaluated appropriately.

Previously, a common approach to evaluate online algorithms was to compute a corresponding offline strategy by "querying" the online algorithm at each state ("tabularization" of the strategy) [Lisý et al., 2015, Šustr et al., 2019]. One would then report the exploitability of the resulting offline strategy. We show that this is not generally possible and that naive tabularization can also lead to incorrect conclusions about the online algorithm's worst-case performance. As a consequence we show that some algorithms previously considered to be sound are not.

We first give a simple example of how an online algorithm can lose to an adversary in a repeated game setting. Previously, such an algorithm would be considered optimal based on a naive tabularization. We build on top of this example to introduce a framework for properly evaluating an online algorithm's performance. Within this framework, we introduce the definition of a sound and ϵ-sound algorithm. Like the exploitability of a strategy in the offline setting, the soundness of an algorithm is a measure of its performance against a worst-case adversary. Importantly, this notion collapses to the previous notion of exploitability when the algorithm follows a fixed strategy profile.

We then introduce a consistency framework, a hierarchy that formally states in what sense an online algorithm plays "consistently" with an ϵ-equilibrium. The hierarchy allows stating multiple bounds on the algorithm's soundness, based on the ϵ-equilibrium and consistency type. The stronger the consistency is in our hierarchy, the stronger are the bounds. This further illustrates the discrepancy of search in perfect and imperfect information settings, as these bounds sometimes differ for perfect and imperfect information games.

The definitions of soundness and the consistency hierarchy finally provide appropriate tools to analyze online algorithms in imperfect information games. We thus inspect some of the previous online algorithms in a new light, bringing new insights into their worst-case performance guarantees. Namely, we focus on the Online Outcome Sampling (OOS) [Lisý et al., 2015] algorithm. Consider the following statement from the OOS publication: "We show that OOS is consistent, i.e., it is guaranteed to converge to an equilibrium strategy as search time increases. To the best of our knowledge, this is not the case for any existing online game playing algorithm. . . ' The problem is that OOS provides only the weakest of the introduced consistencies — local consistency. As the local consistency gives no guarantee for imperfect information games (in contrast to perfect information games), OOS (and potentially other locally consistent algorithms) can be highly exploited by an adversary. The experimental section then confirms this issue for OOS in two small imperfect information games.

5. Variance Reduction in Monte Carlo Counterfactual Regret Minimization (VR-MCCFR) for Extensive Form Games using Baselines

Policy gradient algorithms have shown remarkable success in single-agent reinforcement learning (RL) [Mnih et al., 2016, Schulman et al., 2017]. While there has been evidence of empirical success in multiagent problems [Foerster et al., 2017, Bansal et al., 2018], the assumptions made by RL methods generally do not hold in multiagent partially-observable environments. Hence, they are not guaranteed to find an optimal policy, even with tabular representations in two-player zero-sum (competitive) games [Littman, 1994]. As a result, policy iteration algorithms based on computational game theory and regret minimization have been the preferred formalism in this setting. Counterfactual regret minimization [Zinkevich et al., 2007] has been a core component of this progress in Poker AI, leading to solving Heads-Up Limit Texas Hold'em [Bowling et al., 2015] and defeating professional poker players in No-Limit [Moravčík et al., 2017, Brown and Sandholm, 2018].

The two fields of RL and computational game theory have largely grown independently. However, there has been recent work that relates approaches within these two communities. Fictitious self-play uses RL to compute approximate best responses and supervised learning to combine responses [Heinrich et al., 2015]. This idea is extended to a unified training framework that can produce more general policies by regularizing over generated response oracles [Lanctot et al., 2017]. RL-style regressors were first used to compress regrets in game theorietic algorithms [Waugh et al., 2015]. DeepStack introduced deep neural networks as generalized value-function approximators for online planning in imperfect information games [Moravčík et al., 2017]. These value functions operate on a belief-space over all possible states consistent with the players' observations.

This chapter similarly unites concepts from both fields, proposing an unbiased variance reduction technique for Monte Carlo counterfactual regret minimization using an analog of state-action baselines from actor-critic RL methods. While policy gradient methods typically involve Monte Carlo estimates, the analog in imperfect information settings is Monte Carlo Counterfactual Regret Minimization (MCCFR) [Lanctot et al., 2009]. Policy gradient estimates based on a single sample of an episode suffer significantly from variance. A common technique to decrease the variance is a state or state-action dependent baseline value that is subtracted from the observed return. These methods can drastically improve the convergence speed. However, no such methods are known for MCCFR.

MCCFR is a sample based algorithm in imperfect information settings, which approximates counterfactual regret minimization (CFR) by estimating regret quantities necessary for updating the policy. While MCCFR can offer faster short-term convergence than original CFR in large games, it suffers from high variance which leads to slower long-term convergence.

CFR+ provides significantly faster empirical performance and made solving Heads-Up Limit Texas Hold'em possible [Bowling et al., 2015]. Unfortunately, CFR+ has so far did not outperform CFR in Monte Carlo settings [Burch, 2017].

In this work, we reformulate the value estimates using a control variate and a state-action baseline. The new formulation includes any approximation of the counterfactual values, which allows for a range of different ways to insert domain-specific knowledge (if available) but also to design values that are learned online.

Our experiments show two orders of magnitude improvement over MCCFR. For the common testbed imperfect information game – Leduc Poker – Variance Reduction MCCFR (VR-MCCFR) with a state-action baseline needs 250 times fewer iterations than MCCFR to reach the same solution quality. In contrast to RL algorithms in perfect information settings, where state-action baselines bring little to no improvement over state baselines [Tucker et al., 2018], state-action baselines lead to significant improvement over state baselines in multiagent partially-observable settings. We suspect this is due to variance from the environment and different dynamics of the policies during the computation.

6. Refining Subgames in Large Imperfect Information Games

Extensive form games are a powerful model capturing a wide class of real-world problems. The games can be either perfect information (Chess) or imperfect information (poker). Applications of imperfect information games range from security problems [Pita et al., 2009] to card games [Bowling et al., 2015]

The largest imperfect information game to be (essentially) solved today is the limit version of two-player Texas Hold'em poker [Bowling et al., 2015], with approximately 10^{17} nodes [Johanson, 2013]. Unfortunately, many games remain that are much too large to be solved with current techniques. For example, the more popular "No-Limit" variant of two-player Texas Hold'em poker has approximately 10^{165} nodes [Johanson, 2013].

The leading approach to solving imperfect information games of this magnitude is to create a simplified abstraction of the game, compute an ϵ-equilibrium in the abstract game, and finally use the strategy from the abstracted game to play the original, unabstracted game [Billings et al., 2003a] [Sandholm, 2010] [Johanson et al., 2013] [Gibson, 2014]. The amount of simplification needed to produce the abstracted game is determined by the maximum size of the game tree that we are able to learn with the computing resources available. While abstraction pathologies mean that larger abstractions are not guaranteed to produce better strategies [Waugh et al., 2009], empirical results have shown that finer-grained abstractions are generally better [Johanson et al., 2013]

An appealing compromise is to pre-calculate the largest possible abstraction we can handle for the entire game and then improve this in real-time with refinements. The original strategy is used to play the early parts of the game (the trunk) and once the remaining portion of the game tree (the subgame) becomes tractable, we can refine the strategy for the subgame in real-time using even finer-grained abstraction. Figure 6.1 illustrates the approach.

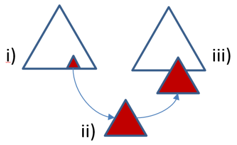

Figure 6.1: Subgame refinement framework. (i) the strategy for the game is pre-computed using coarse-grained abstraction (ii) during the play, once we reach a node defining a sufficiently small subgame, we refine the strategy for that subgame (iii) this together with the original strategy for the trunk creates a combined strategy. The point is to produce improved combined strategy

Note that not only can we enlarge the size of the abstraction in the subgame, we can also reduce the "off the tree problem". When an opponent takes an action that is not found in the abstraction, it needs to be mapped onto a (similar) one in the abstraction. This mapping can destroy relevant game information. To reduce this effect, we can construct the subgame so that it starts in the exact state of the game so far [Ganzfried and Sandholm, 2015].

Subgame refinement has been successfully used in perfect information games to improve the strategies [Müller and Gasser, 1996] [Müller, 2002]. Unfortunately, the nature of imperfect information games means that it is difficult to isolate subgames. Current attempts to apply subgame refinement to imperfect information games have lead to marginal gains or potentially result in a more exploitable final solution. The reason for this is that if we change our strategy in the subgame then this gives our opponent the opportunity to exploit our combined strategy by altering their behavior in the trunk of the game. See [Burch et al., 2014] or [Ganzfried and Sandholm, 2015] for details and several nice examples of this flaw.

The first approach, “endgame solving”, does not guarantee a decrease in exploitability, and can instead produce a strategy that is drastically more exploitable. [Ganzfried and Sandholm, 2015]. The second approach, re-solving, was originally designed for subgame strategy re-solving. In other words, it aims to reproduce the original strategy from a compact representation. The resulting strategy is guaranteed to be no more exploitable than the original one. Although this technique can be used to refine the subgame strategy, there is no explicit construction that forces the refined strategy to be any better than the original, even if much stronger strategies exist. [Burch et al., 2014]

In this chapter, we present a new technique, max-margin subgame refinement, that is tailor-made to reduce exploitability in imperfect information games. We introduce the notion of subgame margin, a simple value with appealing properties, which motivates subgame refinements that result in large positive margins.

We regard the problem of safe subgame refinement as a linear optimization problem. This perspective demonstrates the drawbacks and connections between the two previous approaches, and ultimately introduce linear optimization to maximize the subgame margin. Subsequently, we describe an imperfect information game construction that can be used to find such a strategy (rather than solving the resulting linear optimization problem). This allows us to solve larger subgames using recently introduced techniques, namely the CFR+ [Tammelin et al., 2015] and domain-specific speedup tricks [Johanson et al., 2012].

Finally, we experimentally evaluate all the approaches - endgame solving, re-solving and max-margin subgame refinement. For the first time, we evaluate these techniques on the safe-refinement task as part of a large-scale game by using one of the top participating agents in AAAI-14 Computer Poker Competition as the baseline strategy to be refined in subgames.

7. AIVAT: A New Variance Reduction Technique for Agent Evaluation in Imperfect Information Games

7.1 Introduction

Evaluating an agent's performance in stochastic settings can be hard. Non-zero variance in outcomes means the game must be played multiple times to compute a confidence interval that likely contains the true expected value. Regardless of whether the variance arises from player actions or from chance events, we might need to observe many samples before we get a narrow enough interval to draw desirable conclusions. In many situations, it is simply not feasible (e.g., when the evaluation involves human participation) to simply observe more samples, so we must turn to statistical techniques that use additional information to help narrow the confidence interval.

This agent evaluation problem is commonly encountered in games, where the goal is to estimate the expected performance difference between players. For example, consider poker games. Poker is not only a long-standing challenge problem for AI [von Neumann, 1928, Koller and Pfeffer, 1997, Billings et al., 2002] with annual competitions [Zinkevich and Littman, 2006, Bard et al., 2013], but also a very popular game played by an estimated 150 million players worldwide [Economist, 2007]. Heads-up no-limit Texas hold'em (HUNL) is a particular variant of the game that has received considerable attention in the AI community in recent years, including a "Brains vs. AI" event pitting Claudico [cmu, 2015], a top HUNL computer program, against professional poker players. That match involved 80,000 hands of poker, played over seven days, involving four poker players, playing dozens of hours each. Despite Claudico losing by over 9 big blinds per 100 hands (a margin that is considered huge by poker professionals) [Wood, 2015], the result is only on the edge of statistical significance, making it hard to draw a conclusion from this large investment of human time.

Previous techniques for variance reduction to achieve stronger statistical conclusions in this setting have used two broad classes of statistical techniques. Techniques like MIVAT [White and Bowling, 2009] use the method of control variates with heuristic value estimates to reduce the variance caused by chance events. The technique of importance sampling over imaginary observations [Bowling et al., 2008] takes a different approach, using knowledge of a player strategy to evaluate multiple states given a single observation. Imaginary observations can be used to reduce the variance caused by privately observed chance events, as well as the player's randomly chosen choice of whether to make any actions which would immediately end the game.

Techniques from the two classes can be combined, but are not specifically designed to work together for the greatest reduction in variance, and none of the techniques deal with the variance caused by non-terminal action selection. Because good play in imperfect information games generally requires randomised action selection, ignoring action variance is an important shortcoming. We introduce the action-informed value assessment tool (AIVAT), an unbiased low-variance estimator for imperfect information games which extends the use of control variates to player actions, and makes explicit use of imaginary observations to exploit knowledge of the game structure and player strategies.

AIVAT uses heuristic value functions, knowledge of game structure, and knowledge about player strategies to both add a control variate term for chance and player decisions, and to average over multiple possible outcomes given a single observation. We prove AIVAT is unbiased, and demonstrate that with (almost) perfect value functions we see (almost) complete elimination of variance. Even with imprecise value functions, we show variance reduction in a real-world game that significantly exceeds existing techniques. AIVAT's three times reduction in standard deviation allows us to achieve the same statistical significance with ten times less data. A factor of ten is substantial: for problems with limited data, like human play against bots, ten times as many games could be the distinction between practical and impractical.

8. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker

8.1 Introduction

Games have long served as benchmarks and marked milestones of progress in artificial intelligence (AI). In the last two decades, computer programs have reached a performance that exceeds expert human players in many games, e.g., backgammon [Tesauro, 1995], checkers [Schaeffer et al., 1996], chess [Campbell et al., 2002], Jeopardy! [Ferrucci, 2012], Atari video games [Mnih et al., 2015], and go [Silver et al., 2016]. These successes all involve games with information symmetry, where all players have identical information about the current state of the game. This property of perfect information is also at the heart of the algorithms that enabled these successes, e.g., local search during play [Samuel, 1959, Kocsis and Szepesvári, 2006].

The founder of modern game theory and computing pioneer, von Neumann, envisioned reasoning in games without perfect information. "Real life is not like that. Real life consists of bluffing, of little tactics of deception, of asking yourself what is the other man going to think I mean to do. And that is what games are about in my theory." [Bronowski, 1973] One game that fascinated von Neumann was poker, where players are dealt private cards and take turns making bets or bluffing on holding the strongest hand, calling opponents' bets, or folding and giving up on the hand and the bets already added to the pot. Poker is a game of imperfect information, where players' private cards give them asymmetric information about the state of game.

Heads-up no-limit Texas hold'em (HUNL) is a two-player version of poker in which two cards are initially dealt face-down to each player, and additional cards are dealt face-up in three subsequent rounds. No limit is placed on the size of the bets although there is an overall limit to the total amount wagered in each game. AI techniques have previously shown success in the simpler game of heads-up limit Texas hold'em, where all bets are of a fixed size resulting in just under 10^{14} decision points [Bowling et al., 2009, 2015]. By comparison, computers have exceeded expert human performance in go [Silver et al., 2016], a perfect information game with approximately 10^{170} decision points [Allis, 1994]. The imperfect information game HUNL is comparable in size to go, with the number of decision points exceeding 10^{160} [Johanson, 2013].

Imperfect information games require more complex reasoning than similarly sized perfect information games. The correct decision at a particular moment depends upon the probability distribution over private information that the opponent holds, which is revealed through their past actions. However, how our opponent's actions reveal that information depends upon their knowledge of our private information and how our actions reveal it. This kind of recursive reasoning is why one cannot easily reason about game situations in isolation, which is at the heart of heuristic search methods for perfect information games. Competitive AI approaches in imperfect information games typically reason about the entire game and produce a complete strategy prior to play [Zinkevich et al., 2007, Gilpin et al., 2007]. Counterfactual regret minimization (CFR) [Zinkevich et al., 2007, Burch et al., 2014, Bowling et al., 2015] is one such technique that uses self-play to do recursive reasoning through adapting its strategy against itself over successive iterations. If the game is too large to be solved directly, the common response is to solve a smaller, abstracted game. To play the original game, one translates situations and actions from the original game to the abstract game.

Although this approach makes it feasible for programs to reason in a game like HUNL, it does so by squeezing HUNL's 10^{160} situations down to the order of 10^{14} abstract situations. Likely as a result of this loss of information, such programs are behind expert human play. In 2015, the computer program Claudico lost to a team of professional poker players by a margin of 91 mbb/g, which is a "huge margin of victory" [Wood, 2015]. Furthermore, it has been recently shown that abstraction-based programs from the Annual Computer Poker Competition have massive flaws [Lisý and Bowling, 2017a]. Four such programs (including top programs from the 2016 competition) were evaluated using a local best-response technique that produces an approximate lower-bound on how much a strategy can lose. All four abstraction-based programs are beatable by over 3,000 mbb/g, which is four times as large as simply folding each game.

DeepStack takes a fundamentally different approach. It continues to use the recursive reasoning of CFR to handle information asymmetry. However, it does not compute and store a complete strategy prior to play and so has no need for explicit abstraction. Instead it considers each particular situation as it arises during play, but not in isolation. It avoids reasoning about the entire remainder of the game by substituting the computation beyond a certain depth with a fast approximate estimate. This estimate can be thought of as DeepStack's intuition: a gut feeling of the value of holding any possible private cards in any possible poker situation. Finally, DeepStack's intuition, much like human intuition, needs to be trained. We train it with deep learning using examples generated from random poker situations. We show that DeepStack is theoretically sound and produces strategies substantially more difficult to exploit than abstraction-based techniques.

DeepStack defeated professional poker players at HUNL with statistical significance, a game that is similarly sized to go, but with the added complexity of imperfect information. It achieves this goal with little domain knowledge and no training from expert human games. The implications go beyond being a milestone for artificial intelligence. DeepStack represents a paradigm shift in approximating solutions to large, sequential imperfect information games. Abstraction and offline computation of complete strategies has been the dominant approach for almost 20 years [Shi and Littman, 2001, Billings et al., 2003b, Sandholm, 2010]. DeepStack allows computation to be focused on specific situations that arise when making decisions and the use of automatically trained value functions. These are two of the core principles that have powered successes in perfect information games, albeit conceptually simpler to implement in those settings. As a result, the gap between the largest perfect and imperfect information games to have been mastered is mostly closed.

With many real world problems involving information asymmetry, DeepStack also has implications for seeing powerful AI applied more in settings that do not fit the perfect information assumption. The abstraction paradigm for handling imperfect information has shown promise in applications like defending strategic resources [Lisý et al., 2016] and robust decision making as needed for medical treatment recommendations [Chen and Bowling, 2012]. DeepStack's continual re-solving paradigm will hopefully open up many more possibilities.

9. Player of Games

In the 1950s, Arthur L. Samuel developed a Checkers-playing program that employed what is now called minimax search (with alpha-beta pruning) and “rote learning” to improve its evaluation function via self-play [Samuel, 1959]. This investigation inspired many others, and ultimately Samuel co-founded the field of artificial intelligence [Russell and Norvig, 2003] and popularized the term “machine learning”. A few years ago, the world witnessed a computer program defeat a long-standing professional at the game of Go [Silver et al., 2016]. AlphaGo also combined learning and search. Many similar achievements happened in between such as the race for super-human chess leading to DeepBlue [Hsu, 2006] and TD-Gammon teaching itself to play master-level performance in Backgammon through self-play [Tesauro, 1994], continuing the tradition of using games as canonical markers of mainstream progress across the field.

Throughout the stream of successes, there is an important common element: the focus on a single game. Indeed, DeepBlue could not play Go, and Samuel’s program could not play chess. Likewise, AlphaGo could not play chess; however its successor AlphaZero [Silver et al., 2018] could, and did. AlphaZero demonstrated that a single algorithm could master three different perfect information games using a simplification of AlphaGo’s approach, and with minimal human knowledge. Despite this success, AlphaZero could not play poker, and the extension to imperfect information games was unclear.

Meanwhile, approaches taken to achieve super-human poker AI were significantly different. Strong poker play has relied on *game-theoretic reasoning* to ensure that private information is concealed effectively. Initially, super-human poker agents were based primarily on computing approximate Nash equilibria offline [Johanson, 2016]. Search was then added and proved to be a crucial ingredient to achieve super-human success in no-limit variants [Moravčík et al., 2017, Brown and Sandholm, 2018, 2019]. Training for other large games have also been inspired by game-theoretic reasoning and search, such as Hanabi [Bard et al., 2020, Lerer et al., 2020], The Resistance [Serrino et al., 2019], Bridge [Lockhart et al., 2020], AlphaStar [Vinyals et al., 2019], and (no-press) Diplomacy [Anthony et al., 2020, Gray et al., 2020, Bakhtin et al., 2021]. Here again, however, despite remarkable success: each advance was still on a single game, with some clear uses of domain-specific knowledge and structure to reach strong performance.

In this chapter, we introduce Player of Games (PoG), a new algorithm that generalizes the class of games in which strong performance can be achieved using self-play learning, search, and game-theoretic reasoning. PoG uses growing-tree counterfactual regret minimization (GT-CFR): an anytime local search that builds subgames non-uniformly, expanding the tree toward the most relevant future states while iteratively refining values and policies. In addition, PoG employs sound self-play: a learning procedure that trains value-and-policy networks using both game outcomes *and* recursive sub-searches applied to situations that arose in previous searches.

Player of Games is the first algorithm to achieve strong performance in challenge domains with both perfect *and* imperfect information — an important step towards truly general algorithms that can learn in arbitrary environments. Applications of traditional search suffer well-known problems in imperfect information games [Russell and Norvig, 2003]. Evaluation has remained focused on single domains (e.g. poker) despite recent progress toward sound search in imperfect information games [Moravčík et al., 2017, Brown and Sandholm, 2017, Šustr et al., 2020]. Player of Games fills this gap, using a single algorithm with minimal domain-specific knowledge. Its search is sound [Šustr et al., 2020] across these fundamentally different game types: it is guaranteed to find an approximate

Nash equilibrium by re-solving subgames to remain consistent during online play, and yielding low exploitability in practice in small games where exploitability is computable. PoG demonstrates strong performance across four different games: two perfect information (chess and Go) and two imperfect information (poker and Scotland Yard). Finally, unlike poker, Scotland Yard has significantly longer search horizons and game lengths, requiring long-term planning.

10. Conclusion

Combination of the decision-time search with a heuristic value function allowed AI agents to outperform the best human players in games such as Backgammon, Chess, Go, and Arimaa [Tesauro, 1995, Campbell et al., 2002, Silver et al., 2016, Wu, 2015]. More recently, universal agents that learned through self-play and can master multiple games using ”zero” prior knowledge have emerged [Silver et al., 2017a, Schrittwieser et al., 2020].

On the other hand, traditional techniques used in imperfect information games worked very differently. They created a small, abstract version of a game and solved this abstraction in one go. This process was game-specific and had to be manually redone for each new game. While this approach was successful when used in smaller games, it resulted in severe weaknesses in play when applied to larger games, as shown by local best response [Lisý and Bowling, 2017a].

Techniques discussed in this thesis help to bridge the gap between perfect and imperfect information.

DeepStack introduced generalization of the search with the learned value function to imperfect-information settings. This has led to the first AI victory over human professional players in no-limit poker completing a long-standing AI challenge.

Similarly, the generalization of self-play combined with a growing search tree introduced by the Player of Games resulted in a universal algorithm that can master both perfect and imperfect information games starting from scratch.

10.1 Potential Applications

The concepts introduced in this thesis hold potential for new and exciting applications, as many real-world problems lack perfect information.

Sound search from DeepStack is used by GTO Wizard [GTO Wizzard Development Team, 2023], software leveraged by top professional poker players to analyze and improve their play.

A lot of previous work on AI for large imperfect information games was focused on poker. One specificity of poker is that all actions of a player can be observed by their opponents. The game of Scotland Yard tackled by Player of Games is more general and it resembles patrolling games used for real world problems like airport security [Pita et al., 2009] and wildlife protection [Fang et al., 2015].

10.2 Future Work

The main limitation of the sound search used by DeepStack and Player of Games is the need to enumerate all possible information states contained in a public state. This prohibits straightforward use of these algorithms in games with large belief spaces such as full Stratego. This could be an interesting area of future research; potential solutions could involve Monte Carlo subsampling of information sets or learned implicit representations of belief states.

A significant recent milestone in perfect information games was MuZero [Schrittwieser et al., 2020]. It is not only able to learn to play a game without prior knowledge, it is also capable of learning the rules of the game itself just from interaction with the environment. Extending this capability to imperfect information games would produce even more general agents capable of mastering environments with unknown dynamics.

Bibliography

V. L. Allis. *Searching for solutions in games and artificial intelligence*. PhD thesis, University of Limburg, 1994.

Thomas W. Anthony, Tom Eccles, Andrea Tacchetti, János Kramár, Ian M. Gemp, Thomas C. Hudson, Nicolas Porcel, Marc Lanctot, Julien Pérolat, Richard Everett, Satinder Singh, Thore Graepel, and Yoram Bachrach. Learning to play no-press Diplomacy with best response policy iteration. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Anton Bakhtin, David Wu, Adam Lerer, and Noam Brown. No-press Diplomacy from scratch. In *Proceedings of the Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *Proceedings of the Sixth International Conference on Learning Representations*, 2018.

Nolan Bard, John Hawkin, Jonathan Rubin, and Martin Zinkevich. The annual computer poker competition. *AI Magazine*, 34(2):112–112, 2013.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The Hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280, 2020.

Darse Billings, Aaaron Davidson, Jonathen Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, 134(1–2):201–240, 2002.

Darse Billings, Neil Burch, Aaaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 661–668, 2003a.

Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *IJCAI*, volume 3, page 661, 2003b.

Michael Bowling, Michael Johanson, Neil Burch, and Duane Szafron. Strategy evaluation in extensive games with importance sampling. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML)*, pages 72–79, 2008.

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

Michael H Bowling, Nicholas Abou Risk, Nolan Bard, Darse Billings, Neil Burch, Joshua Davidson, John Alexander Hawkin, Robert Holte, Michael Johanson, Morgan Kan, et al. A demonstration of the Polaris poker system. In *AAMAS (2)*, pages 1391–1392. Citeseer, 2009.

Branislav Bošanský, Christopher Kiekintveld, Viliam Lisý, Jiri Cermak, and Michal Pechoucek. Double-oracle algorithm for computing an exact Nash equilibrium in zero-sum extensive-form games. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 335–342. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

J. Bronowski. The ascent of man, 1973. Documentary (1973). Episode 13.

Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365 (6456):885–890, 2019.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. In *Thirty-fourth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Neil Burch. *Time and Space: Why Imperfect Information Games are Hard*. PhD thesis, University of Alberta, 2017.

Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. In *AAAI*, pages 602–608, 2014.

Neil Burch, Martin Schmid, Matej Moravčík, Dustin Morill, and Michael Bowling. Aivat: A new variance reduction technique for agent evaluation in imperfect information games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Neil Burch, Matej Moravčík, and Martin Schmid. Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

Katherine Chen and Michael Bowling. Tractable objectives for robust policy optimization. volume 25, 2012.

cmu. Brains Vs. AI. http://www.cs.cmu.edu/brains-vs-ai, 2015.

B Jack Copeland. *The essential turing*. Clarendon Press, 2004.

George B. Dantzig. *A proof of the equivalence of the programming problem and the game problem*, page 330–335. Wiley, New York, NY, 1951.

Economist. Poker: A big deal. *The Economist*, pages 31–38, 2007.

Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Online convex optimization for sequential decision processes and extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1917–1925, 2019.

David A Ferrucci. Introduction to “this is watson”. *IBM Journal of Research and Development*, 56(3.4):1–1, 2012.

Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2017.

Ian Frank and David Basin. Search in games with incomplete information: A case study using bridge card play. *Artificial Intelligence*, 100(1-2):87–123, 1998.

Ian Frank, David A Basin, and Hitoshi Matsubara. Finding optimal strategies for imperfect information games. In *AAAI/IAAI*, pages 500–507, 1998.

Sam Ganzfried and Tuomas Sandholm. Endgame solving in large imperfect-information games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 37–45, 2015.

Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 871–878. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

Richard Gibson. Regret minimization in games and the development of champion multiplayer computer poker-playing agents. *Ph.D. Dissertation, University of Alberta*, 2014.

Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold’em poker. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 50. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press, 2007.

Jonathan Gray, Adam Lerer, Anton Bakhtin, and Noam Brown. Human-level performance in no-press Diplomacy via equilibrium search. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

GTO Wizzard Development Team. GTO Wizzard, 2023. URL https://gtowizard.com/en/.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015.

Feng-Hsiung Hsu. *Behind Deep Blue: Building the Computer that Defeated the World Chess Championship*. Princeton University Press, 2006.

M. Johanson. Measuring the size of large no-limit poker games. Technical Report TR13-01, Department of Computing Science, University of Alberta, 2013.

Michael Johanson, Nolan Bard, Marc Lanctot, Richard Gibson, and Michael Bowling. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2012.

Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. Evaluating state-space abstractions in extensive-form games. volume 1, page 271–278, 05 2013.

Michael Bradley Johanson. *Robust Strategies and Counter-Strategies: From Superhuman to Optimal Play*. PhD thesis, University of Alberta, 2016. URL http://johanson.ca/publications/theses/2016-johanson-phd-thesis/2016-johanson-phd-thesis.pdf.

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *In: ECML-06. Number 4212 in LNCS*, pages 282–293. Springer, 2006.

Daphne Koller and Avi Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94:167–215, 1997.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. *Advances in neural information processing systems*, 22, 2009.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Adam Lerer, Hengyuan Hu, Jakob Foerster, and Noam Brown. Improving policies via search in cooperative partially observable games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Viliam Lisý, Marc Lanctot, and Michael Bowling. Online Monte Carlo counterfactual regret minimization for search in imperfect information games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 27–36. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

V. Lisý and M. Bowling. Equilibrium approximation quality of current no-limit poker bots. In *Proceedings of the AAAI-17 Workshop on Computer Poker and Imperfect Information Games*, 2017a. https://arxiv.org/abs/1612.07547.

Viliam Lisý and Michael Bowling. Eqilibrium approximation quality of current no-limit poker bots. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017b.

Viliam Lisý, Trevor Davis, and Michael Bowling. Counterfactual regret minimization in sequential security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163. Morgan Kaufmann, 1994.

Edward Lockhart, Neil Burch, Nolan Bard, Sebastian Borgeaud, Tom Eccles, Lucas Smaira, and Ray Smith. Human-agent cooperation in bridge bidding. In *Proceedings of the Cooperative AI Workshop at 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1928–1937, 2016.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Matej Moravčík, Martin Schmid, Karel Ha, Milan Hladik, and Stephen J Gaukrodger. Refining subgames in large imperfect information games. In *AAAI*, pages 572–578, 2016.

Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

Martin Müller. Computer Go. *Artificial Intelligence*, 134(1):145–179, 2002.

Martin Müller and Ralph Gasser. Experiments in computer Go endgames. *Games of No Chance*, pages 273–284, 1996.

J v Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

James Pita, Manish Jain, Fernando Ordónez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Using game theory for Los Angeles airport security. *AI Magazine*, 30(1):43, 2009.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition, 2003.

Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44:206–226, 1959.

Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *Ai Magazine*, 31(4):13–32, 2010.

Jonathan Schaeffer, Robert Lake, Paul Lu, and Martin Bryant. Chinook the world man-machine checkers champion. *AI magazine*, 17(1):21–21, 1996.

Martin Schmid. *Search in Imperfect Information Games*. PhD thesis, 2021. URL https://arxiv.org/abs/2111.05884.

Martin Schmid, Matej Moravčík, and Milan Hladik. Bounding the support size in extensive form games with imperfect information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravčík, Rudolf Kadlec, and Michael Bowling. Variance reduction in Monte Carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In *Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

Martin Schmid, Matej Moravčík, Neil Burch, Rudolf Kadlec, Josh Davidson, Kevin Waugh, Nolan Bard, Finbarr Timbers, Marc Lanctot, Zach Holland, et al. Player of games. *arXiv preprint arXiv:2112.03178*, 2021.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, Go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Jack Serrino, Max Kleiman-Weiner, David C. Parkes, and Joshua B. Tenenbaum. Finding friend and foe in multi-agent games. In *Proceedings of the Thirty-third Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Claude E Shannon. XXII. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.

Jiefu Shi and Michael L Littman. Abstraction methods for game theoretic poker. In *Computers and Games: Second International Conference, CG 2000 Hamamatsu, Japan, October 26–28, 2000 Revised Papers 2*, pages 333–345. Springer, 2001.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017b.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 632(6419):1140–1144, 2018.

Michal Šustr, Vojtěch Kovařík, and Viliam Lisý. Monte Carlo continual resolving for online strategy computation in imperfect information games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 224–232, 2019.

Michal Šustr, Martin Schmid, Matej Moravčík, Neil Burch, Marc Lanctot, and Michael Bowling. Sound algorithms in imperfect information games. *arXiv preprint arXiv:2006.08740*, 2020.

O. Tammelin. Solving large imperfect information games using CFR+. *CoRR*, abs/1407.5042, 2014.

Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit texas hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015.

Gerald Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Comput.*, 6(2):215–219, March 1994.

Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

The Stockfish Development Team. Stockfish: Open source chess engine, 2021. URL https://stockfishchess.org/.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. *arXiv preprint arXiv:1802.10031*, 2018.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1): 295–320, 1928.

Michal Šustr, Martin Schmid, Matej Moravčík, Neil Burch, Marc Lanctot, and Michael Bowling. Sound search in imperfect information games. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020.

Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 781–788, 2009.

Kevin Waugh, Dustin Morrill, J. Andrew Bagnell, and Michael Bowling. Solving games with functional regret estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Martha White and Michael H. Bowling. Learning a value analysis tool for agent evaluation. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1976–1981, 2009.

Daniel Whitehouse. *Monte Carlo tree search for games with hidden information and uncertainty*. PhD thesis, University of York, 2014.

J. Wood. Doug Polk and team beat Claudico to win $100,000 from Microsoft & The Rivers Casino. http://pokerfuse.com/news/media-and-software/26854-doug-polk-and-team-beat-claudico-win-100000-microsoft/, 2015.

David J Wu. Designing a winning Arimaa program. *ICGA Journal*, 38(1):19–40, 2015.

M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 905–912, 2007.

Martin Zinkevich and Michael Littman. The AAAI computer poker competition. *Journal of the International Computer Games Association*, 29, 2006. News item.