Automated Construction of Bounded-Loss Imperfect-Recall Abstractions in Extensive-Form Games (Extended Abstract[*])

Jiří Čermák , **Viliam Lisý**[†] and **Branislav Bošanský**
Artificial Intelligence Center, Department of Computer Science,
Faculty of Electrical Engineering, Czech Technical University in Prague
cermak@agents.fel.cvut.cz, {viliam.lisy, branislav.bosansky}@fel.cvut.cz,

Abstract

Information abstraction is one of the methods for tackling large extensive-form games (EFGs). Removing some information available to players reduces the memory required for computing and storing strategies. We present novel domain-independent abstraction methods for creating very coarse abstractions of EFGs that still compute strategies that are (near) optimal in the original game. First, the methods start with an arbitrary abstraction of the original game (domain-specific or the coarsest possible). Next, they iteratively detect which information is required in the abstract game so that a (near) optimal strategy in the original game can be found and include this information into the abstract game. Moreover, the methods are able to exploit *imperfect-recall* abstractions where players can even forget the history of their own actions. We present two algorithms that follow these steps – FPIRA, based on fictitious play, and CFR+IRA, based on counterfactual regret minimization. The experimental evaluation confirms that our methods can closely approximate Nash equilibrium of large games using abstraction with only 0.9% of information sets of the original game.

1 Introduction

Dynamic games with a finite number of moves can be modeled as extensive-form games (EFGs) – a game model capable of describing scenarios with stochastic events and imperfect information. EFGs can model recreational games (e.g., poker) as well as real-world situations in physical security, auctions, or medicine. EFGs are game trees, where nodes correspond to states of the game and edges to the actions of players. Imperfect information of players is represented by grouping indistinguishable states of a player into information sets, which form the decision points of the players.

The size of the extensive-form representation of games grows exponentially with the number of moves the players can play in a sequence (i.e., the horizon of the game). Therefore, models of many practical problems are very large. For example, the smallest version of poker played by people includes over 10^{14} information sets [Bowling *et al.*, 2015]. The memory required to store the strategy (a probability distribution over actions in each information set) is often a severe limitation in computing strategies in these models. There are two main approaches that tackle this issue: *online computation* and the use of *abstractions*.

Online strategy computation avoids computing the complete strategy explicitly before playing the game. Instead, the strategy is computed while playing the game and only for the situations encountered by the player. Recent online game playing algorithms provide performance guarantees [Moravčík *et al.*, 2017; Lisý *et al.*, 2015] and strong practical performance [Moravčík *et al.*, 2017; Brown and Sandholm, 2017], but they also have severe limitations. First of all, these algorithms require a substantial computational effort to make each decision. This is prohibitive in many applications, mainly in robotics and on embedded devices. Furthermore, the most successful methods exploit the specific structure of poker where all actions of the players are fully observable and the amount of hidden information is restricted. Showing in what way these algorithms can be generalized to games without these simplifying properties remains an open problem.

Abstraction methodology solves a smaller abstract game, which is a simplification of the large original game. An abstraction may consider distinct, but similar, to be the same or assume the players use only a subset of actions. The solution of the simplified game is then used for playing the original game. This methodology was, for a long time, in the center of attention of the computational poker community [Gilpin and Sandholm, 2007; Kroer and Sandholm, 2014; Brown and Sandholm, 2017] and even led to the first computer program that outperformed professional poker players in the smallest variant of the game played by people [Rehmeyer *et al.*, 2008]. However, if the original game is too large to be processed even with algorithms linear in the number of the nodes in the game tree, it is very hard to provide any guarantees on the performance of the strategy computed using abstractions. In many (e.g., security) applications, it is desirable to have worst-case guarantees on the performance of the computed strategy. Therefore, we focus on solving games where it is **feasible to traverse all nodes in the game**

[*] This paper is an extended abstract of an article published in Artificial Intelligence [Čermák *et al.*, 2020]

[†] Contact Author

tree, but we still want to **minimize the memory required to store the computed strategy**.

Having equilibrium solving algorithms with small memory requirements is practical for several reasons. First, it allows for solving larger games with more commonly available hardware. Second, a small computed strategy is much more practical in applications. Besides being easier to store and transfer over a network, it is also faster to query during the gameplay. Third, a small strategy is easier to use in portfolio-based approaches, where we want to store multiple different strategies for a game in order to play better [Brown *et al.*, 2018] or exploit suboptimal opponents [Bard *et al.*, 2013].

The problem of reducing the amount of memory required for computing a strategy was addressed by several recent algorithms since the size of required memory is an important bottleneck for scaling up the computation [Bowling *et al.*, 2015]. CFR-BR [Johanson *et al.*, 2012] allows computing a strategy in one-quarter of the memory required by CFR by replacing the updates of one of the players by a best response computation. CFR-D [Burch *et al.*, 2014] allows for using a quadratic computation time to compute a strategy close to the beginning of the game, as a trade-off for requiring only in the order of square root of the storage space. DOEFG [Bošanský *et al.*, 2014] initially stores data only about a small part of the game in which players can use only small subsets of their actions. This restricted game is iteratively expanded with new actions, which can improve players' expected utility until the equilibrium is provably found. All these algorithms assume it is possible to traverse the whole game tree for at least one of the players.

1.1 Contribution

In our paper [Čermák *et al.*, 2020], we propose algorithms that reduce the memory required for computing and representing a (near) optimal strategy for a game using automatically-constructed imperfect-recall abstractions created by domain-independent algorithms. The abstraction considers distinct situations in the game to be equivalent.

Domain independent. Most existing methods for automatically constructing abstractions in extensive-form games were designed primarily for poker. They explicitly work with concepts like cards and rounds of the game [Shi and Littman, 2000; Billings *et al.*, 2003], or at least assume that the actions are publicly observable [Brown and Sandholm, 2015] and ordered [Gilpin *et al.*, 2007]. This is not true in many other domains (e.g., in security). The algorithms proposed in this paper are completely domain-independent and applicable to any extensive-form game.

Imperfect recall. Computationally efficient algorithms for computing (near) optimal strategies in extensive-form games [von Stengel, 1996; Zinkevich *et al.*, 2007; Hoda *et al.*, 2010] require players to remember all the information gained during the game – a property denoted as *perfect recall*. Therefore, the automated abstraction methods designed to be used with these algorithms [Gilpin *et al.*, 2007; Brown and Sandholm, 2015] must construct perfect-recall abstractions to provide performance guarantees. Requiring perfect recall has, however, a significant disadvantage – the number of decision points and hence both the memory required during the computation and the memory required to store the resulting strategy grows exponentially with the number of moves. To achieve additional memory savings, the assumption of perfect recall may need to be violated in the abstract game resulting in *imperfect recall*. Using imperfect recall abstractions can bring exponential savings in memory and these abstractions are particularly useful in games in which exact knowledge about the past is not required for playing optimally. While it may be easy to identify specific examples of imperfect recall abstractions for some games, it is unknown how to systematically and algorithmically identify which information is required for solving the original game and which can be removed. For example, it is usually important to estimate the opponent's cards in imperfect information card games. While past events generally reveal some information, it is not clear which exact event is relevant or not.

The only method for automatically constructing imperfect-recall abstractions with qualitative bounds is presented in [Kroer and Sandholm, 2016]. It considers only a very restricted class of imperfect-recall abstractions. The information sets can be merged only if they satisfy strict properties on the history of actions and there is a mapping between the applicable actions in these information sets such that future courses of the game and possible rewards are similar. In our work, we take a different approach and instead of constraining which information sets can be merged, we design algorithms that start with a very coarse abstraction and refine information sets where necessary. Our approach does not require any specific structure of the abstract game or refined information sets. We introduce two domain-independent algorithms that, starting from an arbitrary imperfect recall abstraction of the original two-player zero-sum perfect recall EFG, simultaneously solve the abstract game, detect the missing information causing problems, and refine the abstraction to include this information. This process is repeated until provable convergence to the desired approximation of the Nash equilibrium of the **original game**.

2 Proposed Algorithms

Our algorithms can be initialized by an arbitrary abstraction since the choice of the initial abstraction does not affect their convergence guarantees. Hence, for example, in poker, we can use the existing state-of-the-art abstractions used by the top poker bots. Even though these abstractions have no guarantees that they allow solving the original poker to optimality, our algorithms will further refine these abstractions where necessary and provide the desired approximation of the Nash equilibrium in the original game. If there is no suitable abstraction available for the solved game, the algorithms can start with a simple coarse imperfect recall abstraction (we provide a domain-independent algorithm for constructing such abstraction) and again update the abstraction until it allows approximation of the Nash equilibrium of the original game to the desired precision.

The first algorithm is Fictitious Play for Imperfect Recall Abstractions (FPIRA)[1] that is based on Fictitious Play (FP).

[1] An earlier version of FPIRA appeared in [Čermák *et al.*, 2017].

In each iteration, it computes a best response to the opponent's average strategy in the original perfect recall game. Since it is a pure behavioral strategy stored only in the reachable parts of the game, its size is usually small. If the pure strategy cannot be represented in the current abstraction of the original game, the abstraction is refined. For example, that happens when two different actions are supposed to be played in the same abstract information set. Furthermore, we need to detect whether adding this pure strategy to the average of the strategies played by the player in past iterations can be represented in the current abstraction. We base this detection on the difference between the quality of the strategies expected from computing the average directly on the original two-player zero-sum EFG with perfect recall and the result obtained from applying it to the abstraction. Finally, we formally prove that the guarantee of convergence of FP to the Nash equilibrium of the original two-player zero-sum EFG with perfect recall directly translates to the guarantee of convergence of FPIRA.

The second algorithm is CFR+ for Imperfect Recall Abstractions (CFR+IRA) where FP is replaced by the CFR+ algorithm [Tammelin, 2014] that is known to have a significantly faster empirical convergence to a Nash equilibrium. To update the abstraction, we compare the expected theoretical speed of convergence of CFR+ in the original game and the convergence achieved in the abstraction. The algorithm solves the game using CFR+, traversing the whole unabstracted game tree in each iteration. All regrets and average strategies computed as a part of CFR+ are stored in the information set structure of the abstraction. There are two procedures for updating the abstraction. (1) First, the abstraction is updated to guarantee the convergence of the algorithm to the Nash equilibrium. As a part of this abstraction update, CFR+IRA samples a subset of k_b unabstracted information sets of the original game. It checks the immediate regret in these monitored information sets for a well-chosen number of iterations before sampling a new subset. If the immediate regret in any of the monitored information sets decreases slower than guaranteed by the no-regret algorithm used for CFR+, we know that the abstraction is too coarse and needs to be refined, so that the monitored information set is detached from its abstract information set. (2) Second, the abstraction is updated using a heuristic update, which significantly improves the empirical convergence of the algorithm and does not break the convergence guarantees. In each CFR+ update, the heuristic samples a subset of k_h unabstracted information sets that belong to the same abstract information sets. If the actions with the highest regrets are different for different unabstracted information in the same abstracted information set, the abstraction is refined. We provide a bound on the average external regret of CFR+IRA and hence show that CFR+IRA is guaranteed to converge to a Nash equilibrium of the original two-player zero-sum EFG with perfect recall.

Both algorithms are conceptually similar to the Double Oracle algorithm (DOEFG, [Bošanský *et al.*, 2014]) since they create a smaller version of the original game and repeatedly refine it until the desired approximation of the Nash equilibrium of the original game is found. Our algorithms, however, use imperfect recall information abstractions during the computation, while DOEFG uses a restricted perfect recall game, where the players are allowed to play only a subset of their actions. Hence, the algorithms introduced in this article exploit a completely different type of sparseness than DOEFG.

3 Experiments

In the experimental evaluation, we compare the memory requirements and runtime of CFR+IRA, FPIRA, CFR+, and DOEFG. For CFR+IRA, we denote by BxHyCFR+IRA the version with $k_b = x$ and $k_h = y$. We present the evaluation on various domains: graph pursuit-evasion game parametrized by the allowed number of moves (GPx), imperfect information version of Goofspiel with x cards per player (GSx), and poker variants with varying the number of bets b, raise sizes r, and consecutive raises allowed c (Pbrc). We demonstrate that CFR+IRA requires at least an order of magnitude less memory than DOEFG and FPIRA to solve a diverse set of domains. Hence it is the most suitable algorithm for (approximately) solving games with limiting memory requirements. We show that even if CFR+IRA is initialized with a trivial automatically built abstraction, it requires building information abstractions with as few as 0.9% of information sets of the original game to find a good approximation of the Nash equilibrium of the original game. Moreover, the results suggest that the relative size of the abstraction built by CFR+IRA will further decrease as the size of the solved game increases. From the runtime perspective, we demonstrate that the CFR+IRA may converge similarly fast to CFR+ applied directly to the original game.

In Figure 1 we present the results showing the abstraction size for GP6, GS6 and P224. We depict the results for CFR+IRA as averages with the standard error over 5 runs with different seeds (the standard error is too small to be visible). The CFR+IRA is capable of solving the games using abstractions with significantly fewer information sets than the rest of the algorithms. For exploitability of the resulting strategies 0.05, the B100H900CFR+IRA uses on average 2.0%, 2.4%, 0.9% of information sets of GP6, GS6 and P224, while DOEFG uses 13.3%, 12.0%, 4.0%. A slow runtime prevented FPIRA from convergence to strategies with exploitability below 0.05 in the given time. The experiments presented in the full paper show that it converges in smaller games. In case of GP6, GS6 and P224, $k_h + k_b = 100$ corresponds to storing regrets required for the abstraction update only in 0.09%, 0.06%, 0.03% of information sets of the whole game respectively.

The plots in Figure 2 show the runtime comparison in seconds. These plots further confirm the runtime dominance of CFR+IRA over FPIRA. The high runtime of FPIRA is the cause for omitting the results of FPIRA for smaller exploitabilities of the resulting strategies, as the time required to compute them becomes prohibitive. Furthermore, the results show that the runtime is worse for CFR+IRA where $k_b + k_h = 100$ compared to the case with $k_b + k_h = 1000$.

We provide an improved version that can use a significantly smaller initial abstraction. Additionally, we significantly extend the experimental evaluation of the algorithm.

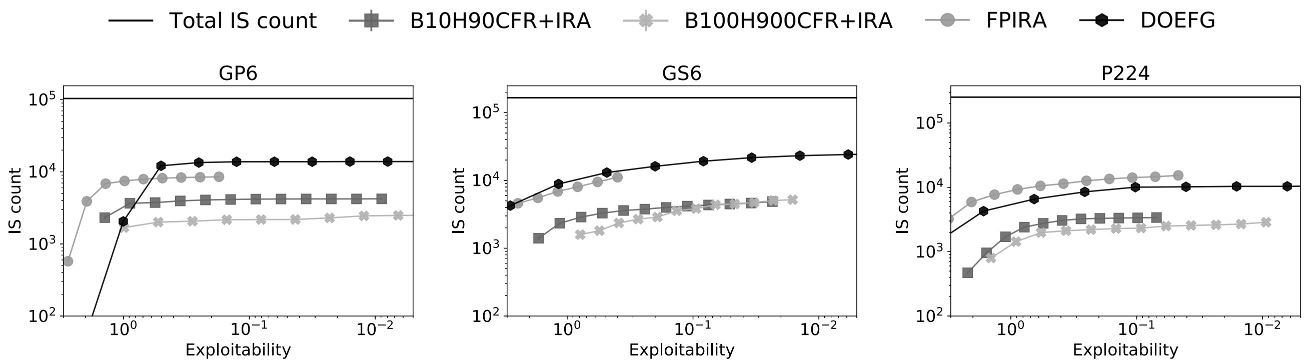

Figure 1: The plots depicting the number of information sets (log y-axis) used by algorithms to compute strategies with the sum of their exploitabilities depicted on the log x-axis for GP6, GS6 and P224.

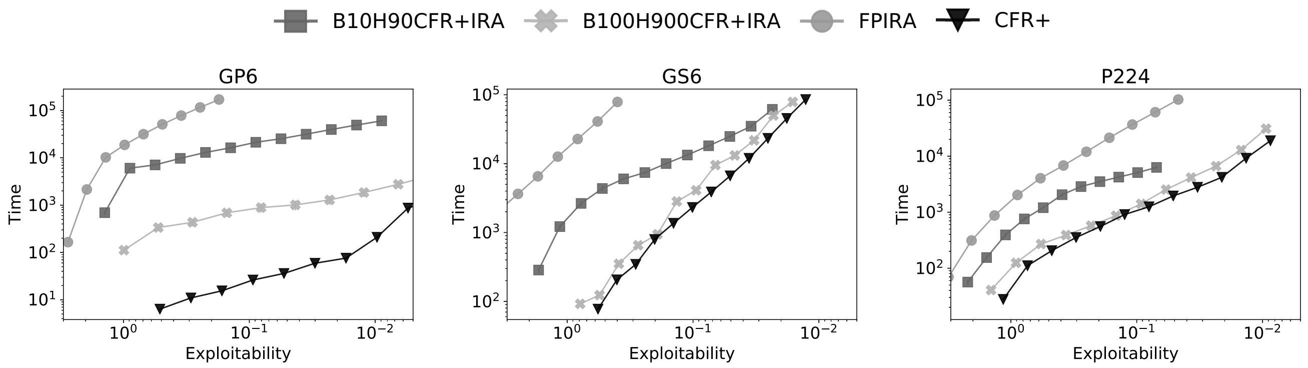

Figure 2: The plots showing runtime of FPIRA and CFR+IRA in seconds (log y-axis) required to reach the given sum of exploitabilities of resulting strategies of player 1 and 2 (log x-axis) for GP6, GS6 and P224 respectivelly.

Additionally, there is a more profound difference between the CFR+IRA runtime and the runtime of CFR+. Both observations are expected since using $k_h + k_b = 100$ and $k_h + k_b = 1000$ means that the algorithm uses less than 0.1% and 1% of information sets for the abstraction update in all 3 domains. Hence it takes longer to refine the abstraction to allow strategies with a smaller exploitability.

4 Conclusions

The imperfect recall abstraction methodology can significantly reduce the memory required to solve large extensive-form games and the size of the final solution. However, solving the resulting imperfect recall abstract games is a computationally hard problem and the standard algorithms for solving extensive-form games are not applicable or lose their convergence guarantees. Hence, there is only a limited amount of work that focuses on using imperfect recall abstractions. Previous works use either very restrictive subclasses of imperfect recall abstractions, heuristic approaches, or use computationally complex algorithms to solve the imperfect recall abstracted game.

In our full article [Čermák *et al.*, 2020], we propose a novel approach to imperfect recall information abstraction, which does not require any specific structure of the imperfect recall abstraction of a game nor does it use computationally complex algorithms to solve it. We introduce two domain-independent algorithms FPIRA and CFR+IRA which can start with an arbitrary imperfect recall abstraction of the solved two-player zero-sum perfect recall extensive-form game. The algorithms simultaneously solve the abstracted game, detect the missing information causing problems and refine the abstraction to include it. This process is repeated until provable convergence to the desired approximation of the Nash equilibrium of the original game.

The experimental evaluation shows that CFR+IRA requires at least an order of magnitude less memory than FPIRA and the Double Oracle algorithm (DOEFG, [Bošanský *et al.*, 2014]) to solve given domains. Even when using trivial automatically build initial imperfect recall abstraction, CFR+IRA is capable of closely approximating Nash equilibrium of large extensive-form games using abstraction with as little as 0.9% of information sets of the original game. Furthermore, the results suggest that the relative size of the abstraction used by CFR+IRA will further decrease as the size of the solved game increases.

Acknowledgements

A part of the original research and preparation of this extended abstract was supported by the Czech Science Foundation (grant no. 18-27483Y and 19-24384Y).

References

[Bard *et al.*, 2013] Nolan Bard, Michael Johanson, Neil Burch, and Michael Bowling. Online implicit agent modelling. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 255–262. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[Billings *et al.*, 2003] 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*, pages 661–668, 2003.

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

[Bošanský *et al.*, 2014] Branislav Bošanský, Christopher Kiekintveld, Viliam Lisý, and Michal Pěchouček. An Exact Double-Oracle Algorithm for Zero-Sum Extensive-Form Games with Imperfect Information. *Journal of Artificial Intelligence Research*, 51:829–866, 2014.

[Brown and Sandholm, 2015] Noam Brown and Tuomas Sandholm. Simultaneous abstraction and equilibrium finding in games. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 489–496. AAAI Press, 2015.

[Brown and Sandholm, 2017] Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, page eaao1733, 2017.

[Brown *et al.*, 2018] Noam Brown, Tuomas Sandholm, and Brandon Amos. Depth-limited solving for imperfect-information games. *arXiv preprint arXiv:1805.08195*, 2018.

[Burch *et al.*, 2014] Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 602–608. AAAI Press, 2014.

[Čermák *et al.*, 2017] Jiří Čermák, Branislav Bošansky, and Viliam Lisy. An algorithm for constructing and solving imperfect recall abstractions of large extensive-form games. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 936–942. AAAI Press, 2017.

[Čermák *et al.*, 2020] Jiří Čermák, Viliam Lisý, and Branislav Bošanský. Automated construction of bounded-loss imperfect-recall abstractions in extensive-form games. *Artificial Intelligence*, 282(103248), 2020.

[Gilpin and Sandholm, 2007] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM (JACM)*, 54(5):25, 2007.

[Gilpin *et al.*, 2007] 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, 2007.

[Hoda *et al.*, 2010] Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. Smoothing Techniques for Computing Nash Equilibria of Sequential Games. *Mathematics of Operations Research*, 35(2):494–512, May 2010.

[Johanson *et al.*, 2012] Michael Johanson, Nolan Bard, Neil Burch, and Michael Bowling. Finding optimal abstract strategies in extensive-form games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1371–1379. AAAI Press, 2012.

[Kroer and Sandholm, 2014] Christian Kroer and Tuomas Sandholm. Extensive-Form Game Abstraction with Bounds. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 621–638. ACM, 2014.

[Kroer and Sandholm, 2016] Christian Kroer and Tuomas Sandholm. Imperfect-Recall Abstractions with Bounds in Games. In *Proceedings of the seventeenth ACM conference on Economics and computation*, pages 459–476. ACM, 2016.

[Lisý *et al.*, 2015] 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.

[Moravčík *et al.*, 2017] 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*, 2017.

[Rehmeyer *et al.*, 2008] Julie Rehmeyer, Nathan Fox, and Renzi Rico. Ante up, human: The adventures of polaris the poker-playing robot. *Wired*, 16.12:186–191, December 2008.

[Shi and Littman, 2000] Jiefu Shi and Michael L Littman. Abstraction methods for game theoretic poker. *Computers and Games*, 2063:333–345, 2000.

[Tammelin, 2014] Oskari Tammelin. Cfr+. *CoRR, abs/1407.5042*, 2014.

[von Stengel, 1996] Bernhard von Stengel. Efficient Computation of Behavior Strategies. *Games and Economic Behavior*, 14:220–246, 1996.

[Zinkevich *et al.*, 2007] Martin Zinkevich, Michael Johanson, Michael H Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, pages 1729–1736, 2007.