External Regret Minimization in Extensive-form Games With Imperfect Information

Mahshid Rahmani Hanzaki [*1] **Elaheh Toulabinejad** [*1]

Abstract

We can use extensive-form games to model decision-making scenarios involving imperfect information. A common approach to solving such games is to compute the Nash Equilibrium. Previously, Linear Programming methods were used for calculating Nash Equilibrium, but they proved impractical for large-size games. Regret minimization offers an alternative approach for iteratively approximating the Nash Equilibrium in two-player zero-sum games. Minimizing counterfactual regret (CFR) emerges as one of the widely accepted methods for finding an approximate Nash Equilibrium in these games. In this survey, we explain the CFR algorithm and provide an overview of its limitations. We then discuss CFR variations and explain how they address the limitations of CFR. Finally, we conclude by highlighting some of the drawbacks of these methods and suggesting potential directions for future research.

1. Introduction

Extensive-form game models are used for multi-agent decision-making scenarios involving imperfect information. A natural solution for these types of games is Nash Equilibrium, where no player would gain utility by changing their strategy. To find the Nash Equilibrium in extensive-form games with perfect information, traversal of the game tree is needed. However, this strategy cannot be applied to imperfect information games; therefore, we require other methods to calculate the Nash Equilibrium for such games.

The earliest method of finding Nash Equilibrium in these settings was to convert it to a normal-form game and use Linear Programming to find an equilibrium. However, the number of strategies in the normal-form game grows rapidly with more information sets. This makes solving even a small game with 300 information sets impractical using this method.

Another proposed method to solve extensive-form games involves retaining the extensive-form structure and using Linear Programming to find Nash Equilibrium (Romanovskii, 1962). To use the extensive form of the game, a sequence-form representation of the strategy was introduced. With sequence-form Linear Programming (SFLP), there was no need to convert the game into its normal-form representation, which is exponential. In this approach, the computation time of SFLP increased polynomially with the size of the game representation. By using SFLP, larger games with many more information sets were solved. For example, with this method, the Rhode Island poker game—a game with 10^6 information sets—was solved

Regret minimization is another method that emerged to find the Nash Equilibrium in extensive-form games with imperfect information. Regret minimization is an online learning concept with an interesting connection to games. In games, the concept of regret is defined as the difference between the utility of the selected action and the utility of the best fixed strategy. In an extensive-form game, if all players use a regret minimization algorithm, the average strategy converges to a coarse correlated equilibrium (Hart & Mas-Colell, 2000). More interestingly, in a two-player zero-sum game, the average strategy converges to Nash Equilibrium. This is why regret minimization can be used to calculate Nash Equilibrium.

In 2007, an idea called counterfactual regret (CFR) was introduced to approximate Nash Equilibrium in imperfect information games by building upon the concept of regret minimization (Zinkevich et al., 2007). CFR is one of the most widely used methods in poker competitions to this day.

Using Vanilla CFR has drawbacks, including high memory and computation requirements for large-sized games, the need to traverse the entire game tree to calculate regret, and difficulties in generalizing. To address these issues, various variations of CFR have been introduced. In extensive-form games, there are situations where a player may not remember their own past actions. If every player

[*]Equal contribution [1]The University of Alberta, Edmonton, Canada. Correspondence to: Mahshid Rahmani Hanzaki <mahshid3@ualberta.ca>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

can remember their past actions and information sets, the game is considered to have perfect recall. In this survey, we will focus on counterfactual regret minimization methods for finding Nash Equilibrium in two-player zero-sum imperfect information extensive form games with perfect recall. We will discuss why each algorithm was developed and outline their respective downsides.

2. Background

In this section, we provide a formal definition of imperfect information extensive-form games and Nash Equilibrium. Moreover, we explain the general idea of regret minimization algorithms.

2.1. Extensive-form Games

Normal-form games cannot represent sequential decision-making, which is why extensive-form games are used to model sequential interactions. We model extensive-form games as a game tree where each terminal node represents the payoffs for each player and each non-terminal node is a point where one of the players chooses an action. Extensive-form games are modeled as a game tree, where each terminal node represents the payoffs for each player, and each non-terminal node is a point where one of the players chooses an action. In some situations, players might have partial or no knowledge about the actions taken by other players or the events in the game. Imperfect information extensive-form games model these situations.

Definition 1 *According to Osborne & Rubinstein (1994), an extensive-form game has the following components:*

- *A finite set N representing the set of players*
- *a finite set H of sequences that satisfy these properties:*
  - *The empty sequence $\emptyset$ is a member of H.*
  - *Every prefix of a sequence in H is also in H.*
  - *$Z \subset H$ represents the terminal histories.*
- *A function P that assigns to each non-terminal history h a member of $N \cup c$. P is referred to as the* ***Player function****, where $P(h)$ denotes the player who takes an action after the history h.*
- *A function f_c that associates with every history h (for which $P(h) = c$) a probability measure $f_c(\cdot \mid h)$ on $A(h)$. Each of these probabilities is independent of the others.*
- *For each player $i \in N$ a partition $\mathcal{I}_i$ of $\{h \in H : P(h) = i\}$ with property that $A(h) = A(h\prime)$ whenever h and $h\prime$ are in the same member of the partition. For $I_i \in \mathcal{I}_i$ we denote by $A(I_i)$ the set $A(h)$ and by $P(I_i)$ the player $P(h)$ for any $h \in I_i$. ($\mathcal{I}_i$ is the* ***information partition*** *of player i; a set $I_i \in \mathcal{I}_i$ is an* ***information set*** *of player i.)*
- *For each player $i \in N$a utility function u_i from the terminal states Z to real numbers. In a* ***two-player zero-sum game*** *$u_1 = -u_2$.*
- *For each player $i \in N$ a preference $\succsim_i$ on lotteries over Z (the* ***preference relation*** *of player i) that can be represented as expected value of a payoff function defined as Z.*

2.2. Nash Equilibrium

A strategy profile of player i is denoted as σ_i where it assigns a distribution over $A(I_i)$. A strategy profile consists of $\sigma_1, \sigma_2, ..., \sigma_n$ where σ_{-i} is the strategies of every player except player i.

Definition 2 *(Shoham & Leyton-Brown, 2008) Player $i's$ best response to strategy profile σ_{-i} is a mixed strategy $\sigma^* \in \Sigma_i$ such that $u_i(\sigma^*, \sigma_{-i}) \leq u_i(\sigma_i, \sigma_{-i})$ for all strategies $\sigma_i \in \Sigma_i$.*

Definition 3 *(Shoham & Leyton-Brown, 2008) A strategy profile $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$ is a* ***Nash Equilibrium*** *if for all agents i, σ_i is a best response to σ_{-i}.*

Another solution concept is that players might not care to changing their strategy profile if the change in their utility is too small

Definition 4 *(Shoham & Leyton-Brown, 2008) Fix $\epsilon > 0$. A strategy profile is ϵ-Nash Equilibrium if for all agents i and all strategies $\sigma_i' \neq \sigma_i$, $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i}) - \epsilon$.*

2.3. Regret

Regret minimization is an online learning algorithm. Intuitively, regret is how much utility the player would lose by playing strategy σ_i instead of choosing the best fixed strategy (Zinkevich et al., 2007). We can define the average regret of player i at time T using Equation 1.

$$R_i^T = \frac{1}{T} \max_{\sigma_i^* \in \Sigma_i} \sum_{t=1}^{T} \left(u_i\left(\sigma_i^*, \sigma_{-i}^t\right) - u_i\left(\sigma^t\right) \right) \tag{1}$$

Moreover, we define the average strategy of player i for each information set as follows:

$$\bar{\sigma}_i^t(I)(a) = \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)\sigma^t(I)(a)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)}. \tag{2}$$

3. CFR Methods

In this section, we review the basic CFR algorithm and discuss several variations of CFR that aim to overcome its limitations by modifying different parts of the original algorithm.

3.1. The Vanilla CFR

CFR is a technique used to find approximate solutions for complex extensive-form games. It focuses on minimizing regret by introducing a novel concept called counterfactual regret. The concept involves breaking down overall regret into smaller regret terms for each information set. Zinkevich et al. (2007) proved that total regret is limited by the sum of these counterfactual regret terms. Therefore, if we minimize the sum of regret terms associated with each information set, we can minimize the overall regret.

To explain the concept of CFR simply, we outline the steps involved:

Step 1: Initialize the strategy profiles, regret values, and cumulative regrets at the start of the algorithm.

Step 2: Use the current strategy profile to play through the game. Traverse the entire game tree, making decisions at each information set according to the probabilities defined in the current strategy profile.

Step 3: For each information set and each action visited during gameplay, compute the regret value based on the game's outcomes.

Step 4: Update the cumulative regret values for each action in the information set based on the computed regrets.

Step 5: Calculate the strategy for the next iteration using the updated regret values.

Step 6: Adjust the strategy probabilities to minimize the accumulated regrets over the iterations.

Step 7: After multiple iterations when it reaches the threshold, compute the average strategy. The average strategy is considered to approximate a Nash equilibrium.

Now, we describe how the method is formulated. Define $u_i(\sigma, h)$ as the expected utility when all players play strategy σ upon reaching history h. Moreover, $\pi^{\sigma}_{-i}(h)$ is the probability that all players contribute to the outcome except player i when history h occurs and $\pi^{\sigma}(h, h')$ represents the probability of transitioning from history h to history h' when the player plays strategy σ. Next, for a specific information set I and player i's decisions within that information set, we define **counterfactual regret** as follows:

$$u_i(\sigma, I) = \frac{\sum_{h \in I, h' \in Z} \pi^{\sigma}_{-i}(h) \pi^{\sigma}(h, h') \, u_i(h')}{\pi^{\sigma}_{-i}(I)} \tag{3}$$

Based on the computed value, the **immediate counterfactual regret** $R^T_{i,\text{imm}}(I)$ is calculated as:

$$\frac{1}{T} \max_{a \in A(I)} \sum_{t=1}^{T} \pi^{\sigma^t}_{-i}(I) \left(u_i\left(\sigma^t|_{I \to a}, I \right) - u_i\left(\sigma^t, I \right) \right) \tag{4}$$

Then, The authors emphasize their focus on the positive regret, defined as $R^{T,+}_{i,\text{imm}}(I) = \max\left(R^T_{i,\text{imm}}(I), 0\right)$. Applying this definition, they establish Theorem 1.

Theorem 1. $R^T_i \leq \sum_{I \in \mathcal{I}_i} R^{T,+}_{i,i\ \text{mm}}(I)$

Using the theorem above, they prove that by minimizing immediate counterfactual regret, we can minimize overall regret, which helps in approximating a Nash equilibrium. Therefore, for every information set and all possible actions within each information set, their goal is to minimize the following term by controlling $\sigma_i(I)$:

$$R^T_i(I, a) = \frac{1}{T} \sum_{t=1}^{T} \pi^{\sigma^t}_{-i}(I) \left(u_i\left(\sigma^t|_{I \to a}, I \right) - u_i\left(\sigma^t, I \right) \right) \tag{5}$$

In this algorithm, the actions are chosen based on the level of positive counterfactual regret associated with not selecting each action. Considering $R^{T,+}_i(I, a) = \max\left(R^T_i(I, a), 0\right)$, they choose the strategy for time $T+1$ using

$$\sigma_i^{T+1}(I)(a) = \begin{cases} \frac{R^{T,+}_i(I,a)}{\sum^{T,+}_{a \in A(I)}(I,a)} & \text{if } \sum_{a \in A(I)} R^{T,+}_i(I, a) > 0 \\ \frac{1}{|A(I)|} & \text{otherwise.} \end{cases} \tag{6}$$

In the absence of any actions showing positive counterfactual regret, a random action is selected.

3.2. MCCFR

CFR methods were effective in finding Nash equilibrium in two-player zero-sum imperfect information extensive-form games by minimizing regret per information set to reduce overall regret (Zinkevich et al., 2007). However, there are some drawbacks to using vanilla CFR. Vanilla CFR traverses the entire game tree in each iteration, leading to significant computational costs. As a result, it is not very practical for solving large extensive-form games like poker.

To address the computational cost of CFR in each iteration, Lanctot et al. (2009) introduced a general sampling framework for CFR. They introduced Monte Carlo Counterfactual Minimizing (MCCFR) algorithms that

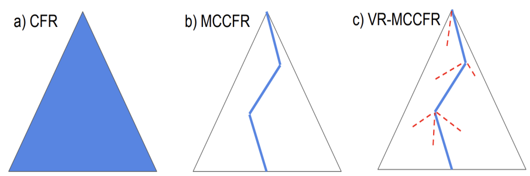

Figure 1: A high-level view of three algorithms a) CFR traverses the whole game tree b) MCCFR takes samples from game tree and calculates the counterfactual regret with samples iteratively c) VR-MCCFR uses baselines for both on-trajectory and off-trajectory node to reduce variance (Schmid et al., 2019).

use sampling methods instead of traversing the entire tree to calculate counterfactual regret. These algorithms vary based on the specific sampling method employed. Using a sampling method reduces the cost per iteration but increases the number of iterations required. The authors introduce two sampling methods from the MCCFR family: outcome sampling, where only a single play of the game is sampled in each iteration, and external sampling, which involves sampling chance nodes and opponent's actions.

They demonstrated that with a reasonable sampling policy, any member of the MCCFR family can be used for equilibrium computation.

3.3. VR-MCCFR

MCCFR methods offer a solution where CFR does not need to traverse the entire game tree to calculate counterfactual regret when it uses a reasonable sampling method. However, there is a downside to sampling methods: The counterfactual regret calculated from one sample may differ from the actual value calculated from the entire game tree.

The idea of MCCFR is that the counterfactual regret calculated for each sample differs from the counterfactual regret for the entire tree, but the expectation of the samples remains the same. This introduces variance to the algorithm and reduces the rate of convergence.

In (Schmid et al., 2019), the authors reformulated the problem to reduce variance without introducing bias. This method is called Variance Reduction in Monte Carlo Counterfactual Regret Minimization (VR-MCCFR). They introduced a baseline to decrease the variance of sampling. Subtracting a state-action-dependent baseline from the observed return helps to reduce variance. This idea is similar to what was previously seen in actor-critic reinforcement learning methods. The experiments performed by Schmid et al. (2019) show that by reducing the variance they could achieve two orders of improvement compared to MCCFR in the game of Leduc Poker. Figure 1 illustrates a high-level view of all three algorithms, vanilla CFR, MCCFR, and VR-MCCFR.

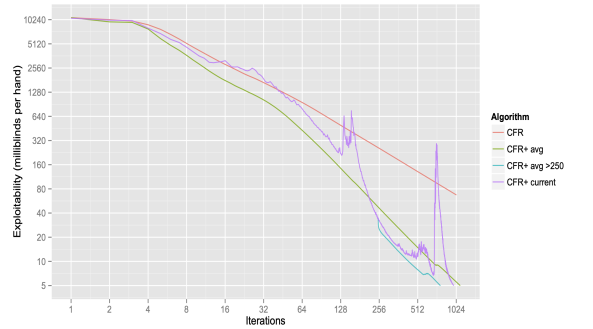

Figure 2: Convergence of exploitability for CFR and different variations of CFR+ for No Limit Texas Hold'em flop subgame (Tammelin, 2014)

3.4. CFR+

Another drawback of using vanilla CFR is the memory and computation issue. CFR needs to store the solution and accumulate regret for each information set. Heads-up limit poker, for example, has around 10^{14} information sets, making it impractical to store accumulating regret for each one. Additionally, computation increases linearly with the number of information sets in vanilla CFR.

To address this problem, a new algorithm based on CFR was introduced called CFR+ (Tammelin, 2014). CFR and its variants use regret matching (Hart & Mas-Colell, 2000) to select strategies. Regret matching selects strategies proportional to the positive cumulative regret. This implies that an action with negative cumulative regret will not be chosen, even if the immediate regret is zero.

CFR+ addresses this problem by employing a variation of regret matching called regret matching+ (Tammelin, 2014). With regret matching+, an action is chosen immediately after demonstrating that it can be beneficial (i.e. when the immediate regret is positive).

In Figure 2, the performance of various CFR+ variations is illustrated and compared to vanilla CFR. CFR+ methods converge approximately an order of magnitude faster than CFR and require less memory. To the best of our knowledge, CFR+ methods represent the state-of-the-art for finding a Nash Equilibrium in imperfect information extensive-form games.

3.5. Double Neural CFR

Tabular-based CFR methods require two large memory structures to record cumulative regret and average strategy. This tabular representation makes these methods challenging to apply in large extensive-form games with limited time and memory.

Moreover, the earlier methods involved simplifying a game using abstraction methods to make it smaller and more

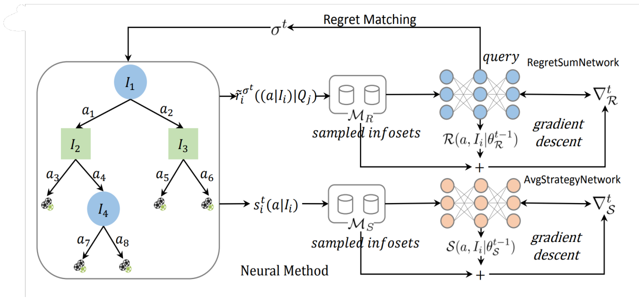

Figure 3: The Double Neural CFR architecture. (Li et al., 2018)

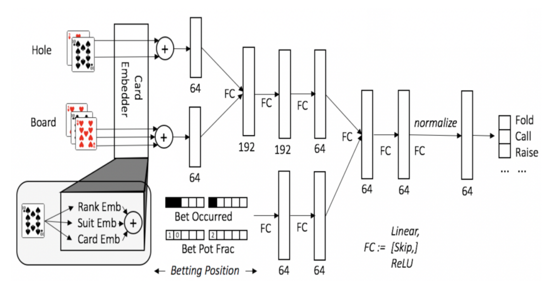

Figure 4: The Deep CFR neural networks architecture. (Brown et al., 2018)

manageable for the tabular CFR algorithm. After solving the simplified game, the solution is applied to the original full game. However, abstracting the game in this way can lead to loss of important details and nuances from the original game. Additionally, game abstractions typically require manual effort and are specific to each game domain.

Li et al. (2018) tackle this problem by introducing a fully neural end-to-end approach using a novel double neural counterfactual regret minimization algorithm. Their algorithm achieves performance comparable to tabular-based counterfactual regret minimization algorithms without relying on two memory structures. It also demonstrates excellent generalization and compression capabilities. Additionally, they introduce a new sampling method that exhibits lower variance compared to outcome sampling and is more memory-efficient than external sampling. This improved sampling method contributes to better algorithm convergence.

As illustrated in Figure 3, they use two neural networks. The RegretSumNetwork (RSN), denoted as $\mathcal{R}\left(a, I_i \mid \theta_{\mathcal{R}}^t\right)$, focuses on calculating the cumulative regret $R^t(a|I_i)$ for a given information set I_i and action a where $\theta_{\mathcal{R}}^t$ represents the network parameters. The cumulative regret helps to find the current strategy. The AvgStrategyNetwork (ASN), denoted as $\mathcal{S}\left(a \mid \theta_{\mathcal{S}}^t\right)$, produces the cumulative strategy $S^t(a \mid I_i)$. The normalization of this cumulative strategy, which involves calculating the weighted average of all previous behavioral strategies up to t iterations, is used to approximate a Nash Equilibrium.

The $\mathcal{R}$ and $\mathcal{S}$ networks are designed for extensive-form games where players take turns based on their observed history. To handle variable-length action sequences, the paper uses recurrent neural networks (RNNs) where each action corresponds to a cell of the RNN. An attention mechanism is added to the RNN architecture to improve decision-making by highlighting important information at different positions in the sequence. This mechanism helps the networks better capture and use key details during gameplay.

3.6. Deep CFR

Brown et al. (2018) critique previous work that combined deep learning and CFR, Double Neural CFR, for potentially lacking theoretical rigor and being limited to small-size games. Therefore, they propose an alternative approach with distinct methods for training data collection and a novel approach for approximating CFR behavior.

Deep CFR simplifies the approximation of the behavior of CFR by using deep neural networks to generalize across similar information sets, thus avoiding the need to calculate cumulative regret for every individual information set in the game. This paper builds upon an MCCFR variant called external sampling and and demonstrates its convergence to an ε-Nash equilibrium in two-player zero-sum games.

The Deep CFR algorithm is detailed in Algorithm 1. This algorithm is structured around several key components. First, separate advantage networks are initialized for each player. These networks denoted as $V(I, a\|\theta_p)$, employ deep neural networks to estimate the advantage of a specific action a at an information set I for a player p. They utilize the neural network architecture depicted in Figure 4 for both the value network, responsible for computing advantages for each player, and the network used to approximate the final average strategy. The advantage is calculated using reservoir-sampled advantage memories, which are also set up for each player along with a strategy memory. Reservoir sampling is utilized to manage these memories, ensuring that they maintain a fixed size and discard the oldest samples when new ones are added.

The algorithm operates within three nested loops. The outer loop iterates over iterations t. Within each iteration, there's a loop that cycles through the players taking turns, and an inner loop performs a constant number of partial traversals (K). During these traversals, the path of traversal is determined using a function called TRAVERSE, which implements the MCCFR external sampling strategy.

At each information set encountered, the algorithm employs a strategy determined by regret matching using the output

Algorithm 1 Deep Counterfactual Regret Minimization (Brown et al., 2018)

function DEEPCFR
  Initialize each player's advantage network $V(I, a \mid \theta_p)$ with parameters θ_p so that it returns 0 for all inputs.
  Initialize reservoir-sampled advantage memories $\mathcal{M}_{V,1}, \mathcal{M}_{V,2}$ and strategy memory $\mathcal{M}_{\Pi}$.
  for CFR iteration $t = 1$ to T **do**
    for each player p **do**
      for traversal $k = 1$ to K **do**
        TRAVERSE($\emptyset, p, \theta_1, \theta_2, \mathcal{M}_{V,p}, \mathcal{M}_{\Pi}$) {Collect data from a game traversal with external sampling}
      end for
      Train θ_p from scratch on loss $\mathcal{L}(\theta_p) = \mathbb{E}_{(I,t',\tilde{r}')\sim\mathcal{M}_{V,p}}[t' \sum_a (\tilde{r}'(a) - V(I, a \mid \theta_p))^2]$
    end for
  end for
  Train θ_Π on loss $\mathcal{L}(\theta_\Pi) = \mathbb{E}_{(I,t',\sigma')\sim\mathcal{M}_\Pi}[t' \sum_a (\sigma'(a) - \Pi(I, a \mid \theta_\Pi))^2]$
  return θ_Π

of the neural network V. The objective of this neural network is to approximate the regret that would have been generated by tabular representations, aiming for a proportional relationship.

After completing a set of traversals (denoted as K), the advantage network parameters are updated from scratch based on the reservoir-sampled advantage memories. This update involves training the advantage network to minimize the mean squared error (MSE) between the predicted advantages from the neural network and the sampled advantages stored in the reservoir.

Once all iterations are completed, the average strategy is trained using the strategy memory. This training process is critical for the strategy to converge towards a Nash equilibrium.

4. Limitations and Future Work

Most research has concentrated on using CFR-based algorithms for a limited range of games, mainly Poker. While CFR and its variations perform exceptionally well in Poker and have even beaten humans at the game, there is no guarantee that these algorithms will perform similarly in different scenarios. Therefore, exploring how these algorithms can be applied to a wider range of games and multi-agent settings beyond Poker is an important area for investigation.

These papers have primarily focused on two-player zero-sum games with perfect recall. Broadening this scope, exploring the application of these algorithms to multi-agent games with imperfect recall as well as alternative solution methods beyond Nash equilibrium presents another promising avenue for research.

In the variations of CFR that we discussed, the authors attempt to solve the game without needing to use the entire game tree. Previous methods used abstraction to simplify the game, but this led to losing some nuances and details of the game. Additionally, using neural networks introduces errors because they are not always perfect at predicting outputs. Therefore, reducing these errors is an important area for future work.

5. Conclusion

In this research survey, we emphasized the significance of employing CFR to approximate a Nash equilibrium in two-player zero-sum extensive-form games. We then discussed vanilla CFR and its variants, which aim to overcome limitations such as handling large games within constrained time, space, memory, and computational resources, as well as avoiding the need to traverse the entire game tree and minimizing loss of game details during abstraction. Finally, we concluded by summarizing existing challenges and potential future research directions.

References

Brown, N., Lerer, A., Gross, S., and Sandholm, T. Deep counterfactual regret minimization. In *International Conference on Machine Learning*, 2018.

Hart, S. and Mas-Colell, A. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte carlo sampling for regret minimization in extensive games. *Advances in neural information processing systems*, 22, 2009.

Li, H., Hu, K., Ge, Z., Jiang, T., Qi, Y., and Song, L. Double neural counterfactual regret minimization. *ArXiv*, abs/1812.10607, 2018.

Osborne, M. J. and Rubinstein, A. *A course in game theory*. MIT press, 1994.

Romanovskii, I. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3:678–681, 1962.

Schmid, M., Burch, N., Lanctot, M., Moravcik, M., Kadlec, R., and Bowling, M. Variance reduction in monte carlo counterfactual regret minimization (vr-mccfr) for extensive form games using baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2157–2164, 2019.

Shoham, Y. and Leyton-Brown, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Tammelin, O. Solving large imperfect information games using cfr+. *arXiv preprint arXiv:1407.5042*, 2014.

Zinkevich, M. A., Johanson, M. B., Bowling, M., and Piccione, C. Regret minimization in games with incomplete information. In *Neural Information Processing Systems*, 2007.