Automatic Public State Space Abstraction in Imperfect Information Games

Martin Schmid, Matej Moravcik, Milan Hladik
Charles University In Prague
{schmidm, moravcim, hladik}@kam.mff.cuni.cz,

Stephen J. Gaukrodger
Koypetition
stephen@koypetition.com

Abstract

Although techniques for finding Nash equilibria in extensive form games have become more powerful in recent years, many games that model real world interactions remain too large to be solved directly. The current approach is to create a smaller abstracted game, allowing the computation of an optimal solution. The strategy can then be used in the original game. Considering public information to create the abstraction can be strategically important, yet very few of the previous abstraction algorithms specifically consider public information or use an expert approach. In this paper, we show that the public information can be crucial, and we present a new, automatic technique for abstracting the public state space. We also present an experimental evaluation in the domain of Texas Holdem poker and show that it outperforms state-of-the-art abstraction algorithms.

Introduction

An extensive game with imperfect information is a general model for interactions of multiple agents in real world situations. Even though there has been substantial progress in solving techniques for these games (Zinkevich et al. 2007; Johanson et al. 2012), the size of many problems makes the computation intractable. For example, no-limit Texas Hold'em Poker (played at the Annual Computer Poker Competition) has approximately 10^{165} game states (Johanson 2013), making it impossible to even store such big strategy.

The current approach to deal with large extensive-form games is to create a smaller, abstracted game. The strategy computed in the abstracted game is then mapped back to the original game. Ideally, an equilibrium from the abstract game results in an equilibrium in the original game (lossless abstraction). Finding lossless abstractions for a subclass of extensive form games called game of ordered signals was studied in (Gilpin and Sandholm 2007).

Unfortunately, lossless abstraction can still produce a game that is far too large, necessitating lossy abstraction. In that case, the goal is to produce a lossy abstraction that retains some guarantees about the strategy's performance (exploitability bounds) in the original game. This was elaborated in (Kroer and Sandholm 2014), where authors present an algorithm that 1. given the maximum number of nodes for the abstraction, finds the abstraction having the minimal bound 2. given the desired bound, finds the smallest abstraction. Unfortunately, both of these algorithms are no easier than finding the equilibrium in the original game (and the fact that we cannot solve the original game is the very reason why we usually seek an abstracted game).

This leaves us with the framework of creating the abstracted game, computing the strategy for the abstracted game and finally checking the performance of the resulting (mapped) strategy of the original game. These works include (Johanson et al. 2013) (Gilpin, Sandholm, and Sørensen 2007) (Ganzfried and Sandholm 2014), all of them focusing on Texas Hold'em poker.

Abstraction techniques from these publications consider only the overall properties of the hand. This information does not correctly capture whether these properties come from the community cards (public information) or the agent's private cards. It is strategically important to distinguish information about the game state that is public (each player can see it), and the information that is private (known only by one player).

Surprisingly little attention was devoted to handling the public information. For example, at the Annual Computer Poker Competition (ACPC), most of the participants do not deal with public information at all, or use an abstraction created by human experts to do so. The only automatic public information abstraction, that we are aware of, appeared in (Waugh et al. 2009), but the used metric does not capture the public state distance well. As far as we know, the authors did not use this public information abstraction in their submission for the no-limit Hold'em Poker in ACPC (while our algorithm shows substantial performance gains in this particular game).

In this publication, we present new technique for abstracting public information. To examine our algorithm, we show that it outperforms state of the art techniques in the domain of no-limit Texas Hold'em poker.

Motivation

In this section, we argue that considering public information separately can be crucial to creating a good abstraction. Previous abstraction techniques evaluated only Texas Hold'em poker, and even within the poker domain it is possible to find

counterexamples in which these techniques fail (although the weaknesses of these approaches are not limited to the poker domain).

Consider the "stud" variants of poker. In stud poker games, players are dealt a combination of private and publicly visible cards, but there are no community cards. Betting rounds take place after every deal. For instance, there are five rounds in seven-card stud. In the first round, each player is dealt two private cards and one public. In the second, third and fourth rounds, each player is dealt one public card. Finally, in the fifth round, players are dealt one private card (making it seven cards in total, three private and four public).

The distinguishing property between Texas Hold'em and stud from the public information perspective is that some of the cards dealt to the player are visible to the opponent. Consider a situation where the player holds $(A\spadesuit A\diamondsuit) - 2\heartsuit 7\heartsuit$ at the second round (the first two cards are private, the other ones public). Current bucketing techniques would merge this situation with $(2\heartsuit 7\heartsuit) - A\spadesuit A\diamondsuit$, since they consider only the overall situation. This is clearly wrong, since these situations are actually extremely different (in the first one the opponent knows that the player holds a pair of aces, in the second one the player also holds a pair of aces, but the opponent does not know it).

In this example, the necessity to consider the public information comes from the need to distinguish situations that differ only in the knowledge that the opponent have about them. This property is not limited to these card games, and our abstraction algorithm would be a good choice in these cases.

In the games where the opponent can see some of the player cards (such as the stud example), abstraction algorithm that can separate states based on the public situation would be superior to the current approaches. Looking back to the Texas Hold'em, one could still benefit from the public information - the community cards in this case. The board structure can provide some information about the opponent (although not as directly as seeing the opponent's card), and considering this information can lead to better performing strategies as we will show in the experimental section.

Overview of Our Approach

Our work naturally falls to the class of abstraction algorithms referred to as the *abstraction as clustering* (Johanson et al. 2013). In these settings, creating the abstraction falls down to 1. definition of a distance measure 2. computing the desired number of clusters using a clustering algorithm

Public State Distance

We view the distance between two public states as a distance between two sets. The set members are all possible games states sharing the same public information. In the poker, the set members would be all hands that the player can hold on a specific board.

To compute the distance between two sets, we use the **Earth Movers Distance (EMD)**, which naturally extends the notion of distance between elements to distance between entire sets. The distance between two sets is then a function of distances between any two elements from these sets.

It is crucial to define a measure of the distance between two elements (the **ground distance**) that is meaningful in the target domain. In this paper we examined two different choices of ground distance for no-limit Texas Hold'em Poker

Our Clustering Technique

Although most of the state space abstractions use k-means as the clustering algorithm (Johanson et al. 2013; Gilpin, Sandholm, and Sørensen 2007), our approach uses **k-medoids** clustering. In our settings, the number of nodes (sets) is relatively small, while computing the distance between two nodes is not a cheap operation. This contrasts with previous approaches, where the number of nodes is large, while the distance function is cheap. Using k-medoids, we can precompute the distance table (containing the distances between any two public states) and then compute the clusters easily.

Public State Distance

Earth Mover's Distance

The EMD defines a distance between two signatures (a signature is a set of tuples $(element, weight)$) and is based on a solution of well-known transportation problem. The transportation problem is used for signature matching by defining one signature as the supplier and the other as the consumer, and solving the optimal transportation of weight using the cost matrix D. $D = [d_{i,j}]$ is called the **ground distance matrix**, where the element $d_{i,j}$ is the **ground distance** between element i in the first signature and element j in the second.

Intuitively, EMD measures the "minimum work" needed to change one signature into another. If we think of a signature as a mass of earth properly spread in space , then a unit of work is defined as moving a unit of earth by a unit of ground distance. For the formal definition see (Rubner, Tomasi, and Guibas 2000).

The EMD naturally extends the notion of distance between single elements to distance between sets of elements. If the ground distance is a metric and the total weights of two signatures are equal, the resulting distance also defines a metric. EMD was successfully used in the poker domain by state-of-the-art poker abstractions (Johanson et al. 2013; Ganzfried and Sandholm 2014).

Public State Distance via EMD

The way we use EMD for a distance between the public states, the signature corresponds to one public state and the elements in that signature are all information sets from that public state. An arbitrary constant w is used for the weights of signature's elements, same for each element.

More specifically, in the case of poker boards, signature represents a specific board and the signature's elements are are all hands which a player can hold on that board. Meaningful ground distance definition is crucial, and we present two ground distances later in the text.

Computing the EMD

To compute EMD, we can use linear programing (note that the transportation problem can be formulated as a system of linear equations). This optimization problem is also an instance of the **minimum-cost flow problem** and, when the weight and the count of elements in signature are constant, also an instance of the **minimum-weight perfect matching problem**. There are fast, polynomial time, combinatorial algorithms to address both of these problems.

Poker Application

Poker has become a standard test bed for large imperfect-information games, with Annual Computer Poker Competition the leading evaluation framework for the computer agents (ACPC 2014). We decided to evaluate our algorithm in this game as well.

While our approach would probably perform very well in the stud-like games (see the motivation section), there are no other automatic abstraction approaches published for this domain. Since this would make empirical evaluation impractical, we decided to limit ourselves to the well studied and active field of Texas Hold'em Poker. We test our implementation against current abstraction algorithms to assess the impact of considering public information.

We used our public state distance to cluster the public states on the second round (flop) and existing abstraction algorithms for the subsequent rounds (turn and river).

When computing the distance between two boards using the EMD, each signature represents a specific board and the elements are all of the hands the player can possibly hold on that board. For example, one board/signature is $(3\blacklozenge 2\spadesuit 2\clubsuit)$ and the elements are $\{(A\spadesuit A\heartsuit), (A\spadesuit K\heartsuit), \ldots (7\blacklozenge 7\clubsuit), \ldots\}$

For the EMD computation, we need to specify a ground distance between hands and we will now discuss the two ground distances.

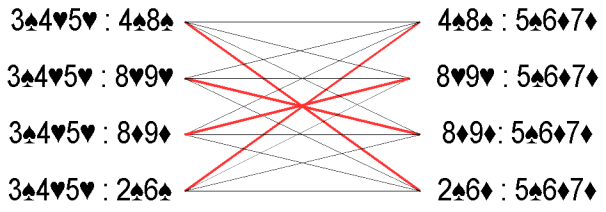

Figure 1: Board distances viewed as the minimum-weight perfect matching problem. Edge's weight corresponds to the ground distance. The board on the left is $(3\spadesuit 4\heartsuit 5\heartsuit)$ and $(5\spadesuit 6\blacklozenge 7\blacklozenge)$ on the right. The bold red edges show how this small matching problem should be solved. For example, $4\spadesuit 8\spadesuit$ from the left is paired with $2\spadesuit 6\spadesuit$, since these hands form a straight on corresponding flops.

Distribution Aware Ground Distance

The first ground distance we examined was presented in (Johanson et al. 2013). They defined the distance between a pair of hands as the EMD between their hand strength distributions.

A **hand strength distribution** is a histogram that summarizes the distribution over possible end-game hand winning probabilities against the random opponent hand distribution.

Unfortunately in the case of Texas Hold'em, the resulting clusters had a poor quality. To leverage the public information in card games where the player's cards are not visible, we examined a different distance. The second ground distance between two hands we evaluated was based not only on the current properties of the hand, but on the hand properties on all previous rounds as well.

Distribution History Aware Ground Distance

While the EMD between hand strength distributions is adequate for describing a hand's properties on the current round of the game, it makes sense to include consideration of how its potential has changed. To capture the hand properties for previous rounds, we will not represent hands with a single hand strength histogram, but rather as a vector of several histograms, one for each preceding round and one more for current one. We call this vector the **distribution history vector**.

We define the ground distance d between two distribution history vectors $p = (p_i, .., p_n)$ and $q = (q_i, ..., q_n)$ as the mean of corresponding EMD distances.

$$d_{p,q} = \sum_{i=1...n} \frac{EMD(p_i, q_i)}{n} \quad (1)$$

and call it **distribution history aware distance** . Using this distance as a ground distance resulted in good empirical results in the domain of no-limit Texas Hold'em poker.

Final Abstraction

Computing the distance for all boards and clustering the boards based on this distance is the first step in creating the abstraction for the flop round. Once we have all boards clustered, we still need to bucket the hands within these clusters.

For this purpose, an existing bucketing algorithm can be used and we chose the **distribution aware bucketing** due to its good performance, low computational cost and easy implementation (Johanson et al. 2013).

Implementation

To compute the EMD, we model the problem as the minimum-weight max-flow problem. Since there are 1176 private hands the players can hold after the flop, the final graph has 2352 nodes (+2 for source, target) and 1385328 edges. To solve this combinatorial problem, we used the lemon library (Dezső, Jüttner, and Kovács 2011), which is a C++ library providing efficient implementations of combinatorial optimization tasks.

Running the minimum-weight perfect matching on the graph of this size took around 100ms on average. Since we need to compute this distance between any two boards, it would take around 200 cpu-days to compute all these distances between non-isomorphic flops. Fortunately, these

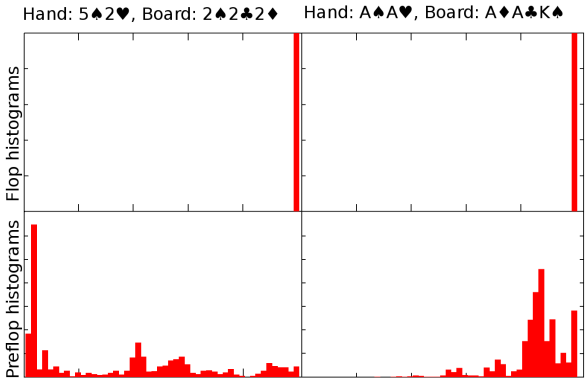

Figure 2: Hands strength histograms for two poker hands and two rounds of Texas Hold'em poker. Histograms for the flop round are on the top row, histogram for the preflop round on the bottom row. Each column represents a single combination of private hand and board cards. Both hands form the best possible combination on the flop round, consequently resulting flop histograms are the same and the EMD between these histograms is 0. Thus also distribution aware ground distance would consider these two hands as same situation. However, the starting hands were very different. The 5♠2♥ is one of the weakest starting combination with low potential to improve. The A♠A♥ is the strongest possible starting hand. Therefore also their preflop round hand strength histograms are very different as we can see in the second row of figure. The EMD of these two histograms is 50.854. Thus the resulting distribution history aware ground distance between the hands is 25.427 and they are considered as very different. It is very important to capture this difference since both hands would be played very differently on the preflop round by any reasonable strategy.

distances can be computed independently, and we ran 20 instances in parallel, finishing the computation in approximately 10 days.

Since we computed the distances for less than 2000 boards, running the k-medoids algorithm using this distance table is very fast. We calculated the final 20 clusters in less than a minute.

Resulting Clusters

The ultimate reason for doing our public state space clustering is to improve the abstraction performance. However, having the resulting clusters, it's interesting to see whether the clusters have any human interpretation. To create these clusters manually, domain experts typically write a set of rules to cluster the flop (same colors, pair on the board, high card, . . .).

Investigating the Figure 3, we see that the clusters indeed have an easy domain interpretation. For example the cluster number 2 consists of flops forming a possible straight. Cluster 12 consists of flops where the cards have the same color, and cluster 15 contains pairs and a high card. Clusters 19 and 20 look similar (ace and a high card), but the later one contains flops having a higher third card.

1	3♦2♠2♣	4♦4♠2♦	5♦5♠4♣	5♦5♠4♠
2	5♦4♠3♠	6♦4♠2♦	6♦5♠3♣	6♦5♠4♦
3	8♦3♦2♣	8♦4♠3♠	9♦3♦2♠	9♦4♠2♣
4	8♦6♦3♠	9♦5♦3♠	9♦6♦2♠	9♦6♦4♣
5	8♦4♠4♣	T♦2♦2♣	T♦T♠8♦	J♦2♦2♣
6	9♦8♠4♣	9♦8♠4♦	9♦8♦7♣	T♦7♠4♣
7	T♦7♠5♠	J♦7♠4♦	J♦7♠5♣	J♦7♠5♠
8	J♦8♦6♣	J♦8♦7♠	J♦9♠8♦	J♦9♦6♠
9	5♦5♠5♣	9♦9♠6♦	T♦T♠2♠	T♦T♠3♠
10	J♦6♦3♠	Q♦5♠3♦	Q♦5♠4♠	Q♦5♦3♣
11	J♦9♦3♣	J♦9♦3♠	J♦9♦4♣	J♦9♦4♠
12	7♦5♦4♦	T♦6♦2♦	J♦7♦2♦	J♦8♦5♦
13	J♦9♠3♦	J♦T♠3♠	Q♦T♦3♣	Q♦J♠4♣
14	J♦T♠7♦	Q♦J♠8♣	Q♦J♦9♠	K♦T♠7♦
15	Q♦Q♠J♦	K♦4♠4♣	K♦6♦6♣	K♦7♠7♣
16	Q♦6♠4♣	K♦6♠5♣	A♦3♦2♣	A♦6♠5♦
17	K♦6♠2♦	A♦7♠2♦	A♦7♦3♣	A♦8♠2♣
18	K♦7♠6♦	K♦9♠6♠	K♦9♦6♣	A♦7♠6♦
19	A♦J♠4♦	A♦J♦6♣	A♦Q♠5♦	A♦Q♦3♣
20	A♦T♦9♠	A♦J♠8♦	A♦K♠9♠	A♦K♠8♠

Figure 3: The resulting 20 clusters. Four flops from every cluster were randomly sampled to create this table.

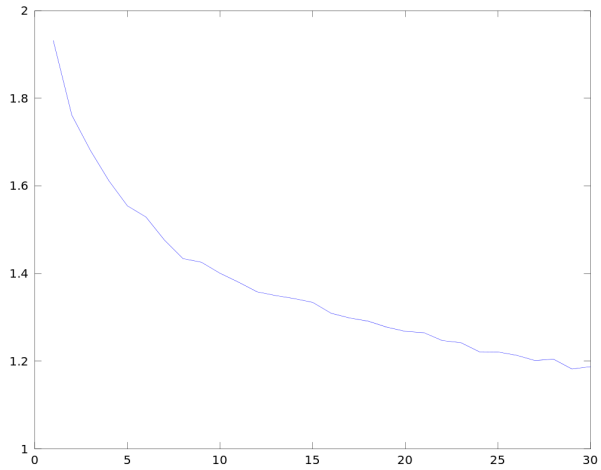

Figure 4: Error function of k-medoids using different number of clusters. For each k ranging from 1 to 30, we ran 100 random initialization of the algorithm. Non-existence of clear "elbow" point suggests that increasing the number of clusters could further improve the abstraction performance.

Summary of Our Approach

First step in our approach is to cluster flops into 20 clusters. For that, we need to compute the distance matrix between all flops.

Each board card combination is represented as a set of hands which a player can hold on that board. The distance between boards is defined as EMD between these sets. We used minimum-cost flow solver for EMD computation, and

distribution history aware distance as the ground distance. Once the distance matrix is computed, public board combinations are clustered into the buckets with k-medoids algorithm.

When public bucketing was created, arbitrary hand clustering algorithm can be used to cluster private hands. We used distribution aware bucketing for this purpose.

Experiments

There are many options available for constructing abstractions in large games, such as Texas Hold'em, and typically even the best abstraction techniques do not have any theoretical guarantees. Therefore, the majority of progress in abstraction generation has been established through empirical evaluation. This involves creating abstract games, finding the optimal strategy (Nash Equilibrium) in these games, and evaluating resulting strategy in the real game.

There are multiple methods for evaluating the performance of the resulting strategies. These include in-game performance against other agents (one-on-one play), in-game performance against an unabstracted Nash equilibrium, and exploitability in the real game. (Johanson et al. 2013) As the second and third methods are not yet tractable in no-limit Texas Hold'em due to its size, we compared strategies created with different abstraction methods using the one-on-one play approach.

We computed strategies using our in-house implementation of the CFR algorithm (Zinkevich et al. 2007), and ran the resulting strategies against each other in the unabstracted game.

This comparison method is currently used to compare abstraction algorithms in no-limit Texas Hold'em poker (Ganzfried and Sandholm 2014), and it produces results comparable to the more sophisticated methods (Johanson et al. 2013).

We chose a combination of **distribution aware** abstraction and **opponent hand strength** clustering as the baseline for the experiment. To make the experiment fair, both abstractions, baseline and our new one, used identical betting abstraction and equal number of buckets on each round.

Since we only computed our public state clusters for the flop round, we also used the same bucketing for the preflop, turn and river rounds in both abstractions.

The Abstracted Game

When comparing two abstractions using one-on-one play, there are several game parameters affecting the final results. In the case of Texas Hold'em poker, these include size of the state space abstraction (number of buckets), stack size and betting abstraction.

In the game we chose for the evaluation, the starting bets were 100 chips for the big blind and 50 chips for the small blind. These numbers were borrowed from the ACPC competition, the most established platform for the computer poker. In contrast to the ACPC, where the players have $20,000$ chips at the start of the game, we chose smaller number to make the abstracted game smaller and easier to solve. In our experiment, the starting stack size for both players was $2,500$ chips. This fact also reduced variance during the evaluation, allowing us to obtain very tight confidence intervals.

State Space Abstraction

As has become prevalent in most recent poker abstractions, we used imperfect recall for the state space abstraction. This has shown to outperform perfect recall abstractions in this particular domain (Waugh et al. 2009). At the start of each round, the player forgets all information from previous rounds.

This property made it both valid and easy to replace the original flop bucketing in the baseline abstraction with our new public bucketing, while keeping the abstraction unchanged in all other rounds.

To evaluate the effect of the abstraction size, we created abstractions of two sizes having 1000 and 2000 buckets for flop, turn and river (there were 169 preflop buckets in both abstraction).

In both cases, the baseline strategy used lossless abstraction for the preflop round, distribution aware bucketing (Johanson et al. 2013) for the flop and turn rounds and finally the opponent cluster hand strength bucketing (Johanson et al. 2013) for the last river round. This type of abstraction is currently used by some of the top computer poker agents.

Our new abstraction differed only in the flop round, where we used our public flop clusters. The game states were initially clustered using 20 top level clusters based on the community cards. For each top level cluster, we used distribution aware bucketing to create 50 inner buckets (for the 1000 total bucket abstraction), and 100 inner buckets (for the 2000 total buckets abstraction).

The number of top level clusters was chosen using a very little experimental evaluation. The best ratio of top level vs inner level buckets can vary from domain to domain. Figure 4 suggests that increasing the granularity of the top level buckets improves the quality of public information clustering in our domain, but one would have to evaluate the resulting bucketing to see if increasing the number of top level buckets leads to improved performance of the final agent.

Results

Results of the matches together with confidence intervals are displayed in Figure 5, values are in milli big blinds per hand (mbb/h).

1000 buckets	$3.473 mbb/h \pm 0.4$ (95% conf. interval)
2000 buckets	$4.366 mbb/h \pm 0.04$ (95% conf. interval)

Figure 5: The experimental results. The decimal point difference in confidence intervals is not a typo, we ran many more iterations in the second case.

Our abstraction outperformed the baseline abstraction in both evaluated games, suggesting that it is beneficial to consider the public information when creating abstractions for no-limit poker.

The winnings in the larger abstraction are greater, which is somewhat intuitive. Once the hand strength distribution

abstraction is fine-grained, the gain from the additional public information is much more significant.

It would be very interesting to compare our automatic approach with human expert abstraction used by the top ACPC competitors, but unfortunately, none of these is publicly available.

Conclusion

While previous publications examined the effect of imperfect recall (Waugh et al. 2009) or hand potential (Ganzfried and Sandholm 2014) on the domain of no-limit poker, this is the first publication dealing with the effect of public information in this domain.

We also presented a new public information abstraction technique, which is a natural member of the "abstraction as a clustering" class of algorithms.

Our algorithm improves the generality of the current state-of-the-art toolkit for solving imperfect information games. Applying this toolkit to a new domain consists of two simple steps - creating a game abstraction and solving the abstracted game. Current state space abstraction algorithms would fail to create well performing abstraction in games where the public information plays a crucial role, thus our algorithm should be very good choice for these games. Our experimental results also showed a significant improvement in the standard test bed for games with imperfect information, the no-limit Texas Hold'em Poker.

Interestingly enough, our clusters for public poker boards presented in the paper displayed an easily interpretable domain structure. This structure can be of interest to domain experts.

Future Work

It would be very interesting to evaluate performance of our new abstraction on other imperfect information games. There is also lot of space for improvement in the no-limit poker, since we implemented our technique only on the flop round. Applying our technique for the later rounds could significantly increase the performance of the resulting abstraction.

To make these experiments possible, we are planing to speed up our algorithms by sampling the data, and by using approximations and heuristics for the EMD computations, presented in (Ganzfried and Sandholm 2014).

References

ACPC. 2014. Acpc 2014 rules, computerpokercompetition.org.

Dezső, B.; Jüttner, A.; and Kovács, P. 2011. Lemon–an open source c++ graph template library. *Electronic Notes in Theoretical Computer Science* 264(5):23–45.

Ganzfried, S., and Sandholm, T. 2014. Potential-aware imperfect-recall abstraction with earth movers distance in imperfect-information games. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Gilpin, A., and Sandholm, T. 2007. Lossless abstraction of imperfect information games. *Journal of the ACM (JACM)* 54(5):25.

Gilpin, A.; Sandholm, T.; and Sørensen, T. B. 2007. 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, 50. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Johanson, M.; Bard, N.; Lanctot, M.; Gibson, R.; and Bowling, M. 2012. Efficient nash equilibrium approximation through monte carlo counterfactual regret minimization. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 837–846. International Foundation for Autonomous Agents and Multiagent Systems.

Johanson, M.; Burch, N.; Valenzano, R.; and Bowling, M. 2013. Evaluating state-space abstractions in extensive-form games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 271–278. International Foundation for Autonomous Agents and Multiagent Systems.

Johanson, M. 2013. Measuring the size of large no-limit poker games. *arXiv preprint arXiv:1302.7008*.

Kroer, C., and Sandholm, T. 2014. Extensive-form game abstraction with bounds. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 621–638. ACM.

Rubner, Y.; Tomasi, C.; and Guibas, L. J. 2000. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2):99–121.

Waugh, K.; Zinkevich, M.; Johanson, M.; Kan, M.; Schnizlein, D.; and Bowling, M. H. 2009. A practical use of imperfect recall. In *SARA*.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, 1729–1736.