Modeling Expert Knowledge in a Heuristic-Based Gin Rummy Agent

Sarah Larkin, William Collicott, Jason Hiebel
Michigan Technological University
1400 Townsend Drive
Houghton, MI 49931
selarkin@mtu.edu,wccollic@mtu.edu,jshiebel@mtu.edu

Abstract

We developed a heuristic-based reflex agent, Tonic, for the EAAI 2021 Undergraduate Research Challenge, which tasks competitors to create an autonomous player to play the card game gin rummy. Tonic's heuristics originate in expert knowledge and inform decision making for the three actions comprising a turn: drawing a card, discarding a card, and deciding when to knock. However, because these strategies are based in human intuition, there is often a lack of specificity to directly model them as algorithms. We developed parameterized models describing that intuition based on factors such as the number of turns played and an estimation of the opponent hand. To hone their performance, we conducted both manual analysis and parameter optimization (grid search) using self-play and play against a simple baseline agent. These heuristic models enable Tonic to win against the baseline agent at least 68% of the time.

Introduction

Gin rummy is a two-player card game played with a standard 52-card deck with aces having a value of one, face cards having a value of ten, and all other cards having the value printed on them. Gameplay consists of a series of hands comprised of alternating turns in which each player attempts to form melds of three or more cards of the same rank (a set) or three or more cards of adjacent ranks within the same suit (a run). The objective is to reduce the overall value of unmelded cards in their hands. Each player's turn consists of drawing, discarding, and deciding whether to end the current hand (knock) when their deadwood (unmelded card) sum is ten or less.

At the end of each hand, players' scores are updated based on their respective deadwood sums. First, the non-knocking player may reduce their deadwood by laying off cards onto the knocking player's hand to form melds. After laying off, the player with the least deadwood remaining scores the difference between theirs and their opponent's. If the knocking player's deadwood is greater than or equal to the opponent's, the opponent scores an additional undercut bonus of 25 points. In the case that a player knocks with 0 deadwood, meaning their hand is 100% melded, the player has "gone gin" and earns an additional 25 points. Once the scores are calculated, another hand begins unless one of the players has a total score of at least 100 points. If this is the case, the game is over and the player with 100 or more points is the winner.

This paper describes the construction of Tonic, a gin rummy agent using heuristics based on domain knowledge, for the EAAI 2021 Undergraduate Research Challenge (Neller 2020). To achieve this, we synthesized strategies discussed by experts and hobbyists. Experts often present strategies in a level of detail that does not dictate precisely in what circumstances those strategies are applicable. For example, Brown (2019) suggests: "If the deck is more than halfway gone, your opponent has probably gotten rid of the worst deadwood, so knocking with 10 points is an invitation to be undercut." A human player could dynamically interpret this and adjust play based on their human opponent and how a game is going. However, this is sufficiently vague enough to require analysis and experimentation to determine effective and implementable strategies. Tonic uses opponent hand estimation, pairs, and deadwood sum to dynamically determine the utility of a card at any time during the game.

We first give a high-level description of conventional gin rummy strategy. Next, we describe heuristics derived from expert and hobbyist strategies. Finally, we discuss an estimator for estimating confidence in card position based on observed play.

Related Work

Gin rummy is a sequential, imperfect information game. Sequential games consist of players alternating between actions, the outcomes of which they can remember and use for future decision making. Common sequential games include chess, checkers, and Go. However, gin rummy is different from perfect information games like chess and Go in that portions of the game state are hidden. For example, a player may not always know their opponent's cards or the next card they will draw from the face down deck. Other sequential, imperfect information games include poker, euchre, and Cluedo.

The most similar of these is poker, and considerable previous work has been published exploring different strategies. The most effective of these is Counterfactual Regret Minimization (CFR), which provides state-of-the-art performance for Texas Hold'em and other poker variants (Brown and Sandholm 2017; Moravčík et al. 2017a). Although this

Category	Type	Cards	Availability of other cards in deck
Broad	Adjacent	T♥, J♥	9♥ is in discard, Q♥ in deck
Broad	Adjacent	T♥, J♥	9♥ is in discard, Q♥ in opponent hand
Broad	Adjacent	Q♥, K♥	J♥ is in deck
Broad	Same Rank	T♥, T♦	T♠, T♣ is in deck
Broad	Inner Strait	J♣, K♣	Q♣ is in deck
Narrow	Adjacent	T♥, J♥	9♥, Q♥ both in deck
Narrow	Same Rank	K♥, K♦	K♠, K♣ both in deck
Not a pair	Adjacent	T♥, J♥	9♥, Q♥ both in discard
Not a pair	Same Rank	T♥, T♦	T♠, T♣ both in discard
Not a pair	Inner Strait	J♣, K♣	Q♣ is in discard

Table 1: Examples of pairs by category and type

is effective for poker, gin rummy has more complications. For example, gin rummy has significantly more sequential actions and many more choices per action than poker. This results in a much broader and deeper game tree for any naive implementation of CFR. Instead, simpler methods, such as those drawn from the heuristic decision making of experts, are more desirable in scope for the research challenge as agents are severely limited in computation time.

Conventional Gin Rummy Strategy

Every player decision modifies two stacks of cards shared by the players. The first is a set of face up discards and the second is a face down deck. When a hand begins, the cards not dealt to the players begin in the deck, with the top card of the deck immediately discarded. Over time, players will choose to draw from either pile and, in turn, will discard to the face up discard pile.

A turn consists of three actions. The first decision that needs to be made by the player is when it is appropriate to pick up the face up card and when to instead draw from the deck. There is a complex trade-off in this decision because the face up card presents a brief certainty, but the face down card presents a wider range of possibilities as to what card we could end up drawing. Conventional domain knowledge suggests best play results from only drawing from the discard if the card completes a meld (Brown 2019).

Once a card has been picked up or drawn, the player must then decide which card to discard. A player faces the constraint that they cannot discard a face up card that they have just drawn. A greedy strategy is to discard the highest deadwood card that is not a part of a meld. However, experts suggest that the player not break up pairs of cards that are one card away from becoming a meld (MacQuaid 2020). Another popular strategy is to avoid holding on to high deadwood pairs late into a hand (Brown 2019).

The final decision a player makes is to decide whether to knock (assuming their deadwood sums to 10 or less) or keep playing. Generally, players advocate knocking as soon as possible to prevent the opponent reducing their deadwood. However, as indicated by Brown (2019), as the hand progresses, knocking at the same target deadwood value allows the opponent to undercut.

Heuristics for Game Strategies

In this section, we introduce heuristics drawn from expert strategies which implement the actions of drawing cards, discarding cards, and choosing when to knock. These strategies are often nonspecific, which requires further modeling. In addition, expert strategies often refer to the idea of a "pair", two cards which require only one more card to form a meld. As such, we first describe pairs and categorize them. Next, we describe the heuristics. Finally, the actions of an opponent can suggest which cards the opponent may hold. As such, we attempt to model the likelihood that our opponent has a card given the cards that they have drawn and discarded.

Pair Formation

We define two categories of pairs to be used in different circumstances: broad and narrow. Examples of broad and narrow pairs can be found in Table 1.

Broad pairs are any two cards such that a third would form a meld with them. This could include a situation in which we held the [3♣, 5♣] and needed to pick up the 4♣, commonly called an inner strait by card players. However, if the card needed is no longer available (discarded by either player in a previous turn), the two cards no longer form a pair. Because an opponent may discard unmelded cards in future turns, we still consider the needed card to be available if it is in the opponent's hand or its location is unknown.

Narrow pairs are a subset of broad pairs for which there are two cards available to complete the meld. For example, if we had a 5♣ and 6♣ but knew that the 4♣ or 7♣ had previously been discarded, we would not consider the 5♣ and 6♣ to be a pair. The same idea applies to cards of the same rank, such that a 9♣, 9♠ would not constitute a pair if either the 9♥ or 9♦ were in the discard pile. Keeping an inner strait is poor strategy because it requires that we pick up one specific card to complete a meld instead of two, limiting our chances of meld formation (MacQuaid 2020). Similarly, pairs of the same suit that consist of Q, K are broad but not narrow because only one card will complete them into a meld. Our goal in pair formation is to limit the number of broad pairs and form narrow pairs because they offer greater melding opportunities.

Approach to Drawing Cards

Each turn in a Gin Rummy game begins by drawing a card either from the face up discard pile or the face down deck. Domain experts often advocate only drawing the face up card if it completes a meld (Brown 2019; MacQuaid 2020). The primary reason not to draw the face up card is to prevent the opponent from seeing what cards are in your hand. Drawing from the deck increases exploration and the likelihood of drawing a desirable card. This approach is implemented in the simple baseline player (SimpleAI) provided for the competition. However, this approach results in weak play in several circumstances such as when drawing the face up card would form a set, allow knocking, or permit pair formation early in the game. By adapting general discard strategies, we implemented models which draw the face up card in these additional situations.

Tonic will always draw the face up card if that card would form a meld. In addition to forming a meld, we choose to pick up the face up card if it brings the overall deadwood value low enough to knock. Furthermore, for the first two turns of each hand, we consider a third scenario: whether drawing it would form a pair with an existing card or cards in our hand and our hand has many unmelded cards and few pairs. We limit pair formation to two turns because experts advocate keeping face-card pairs for a short time into the hand (Brown 2019). The two-turn threshold was determined though a grid search of self-play at various values. The difference between setting the threshold to two or three turns is minimal, but a significant performance drop-off (> 10%) is seen if the threshold exceeds three turns.

We extended this advice to include never drawing a face-card to form a pair because exploring the deck is more productive than forming high-deadwood pairs. If the face up card does not meet any of these criteria, we draw from the deck. This has the advantage that, if the hand was dealt fully melded, the card we draw can be immediately discarded.

The decision to draw the face up card to form a pair depends on how many pairs we currently have, how many melds we currently have, and the deadwood sum of the face up card plus the card it would pair with. The maximum deadwood sum of a pair we are willing to form remains static at eighteen throughout the game. This allows us to maintain high pairs such as two nines but prevents us from cluttering the hand with highest deadwood face card pairs. However, the number of unmelded cards in our hand before drawing a card determines the maximum number of pairs we are attempting to form. For example, if we have nine unmelded cards and the face up card forms a pair, we will pick up the face up card if we have at most three pairs in our hand. Moreover, if we have five unmelded cards, we would draw the face up card only if we have one or no pairs in our hand. In general, we pick up the pair-forming face up card as long as fewer than approximately half of the unmelded cards in our hand are pairs. We chose to limit the pair formation in order avoid the situation in which most or all of our unmelded cards are paired. This scenario would otherwise cause us to prioritize pair formation, meaning we'd continue to draw the face up card when drawing from the deck would give us a better overall chance of forming a meld and thus lowering our deadwood long-term.

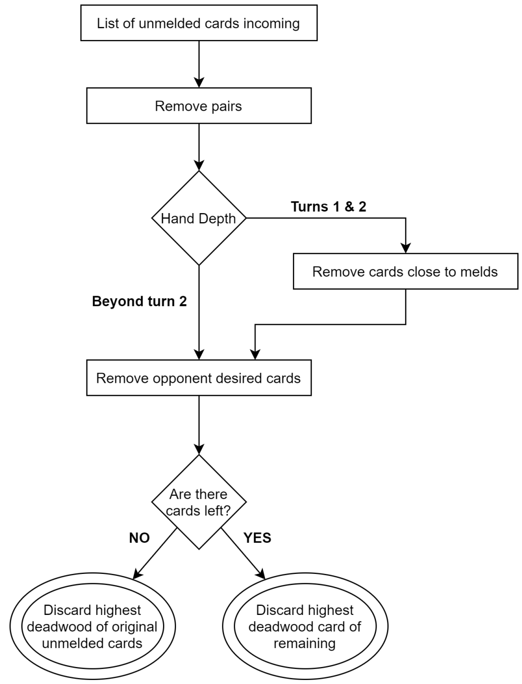


Figure 1: Overview of algorithm for discarding a card

Approach to Discarding Cards

The second action performed by any player in each turn is discarding a card. This is an area over which a player has much more control than drawing or knocking. For example, when drawing, there are only two choices: draw the face up card or draw the face down card. The face down card can be any of up to thirty unknown cards, depending on the current length of the hand. In contrast, when discarding a card, there are up to eleven cards to choose from and the player wants to determine which is least useful. This can be determined from a knowledge of what is in their hand and the discard pile, as well as an estimate of what their opponent holds.

We begin considering cards for discard by first choosing a set of melds in the hand to minimize deadwood. In the case that there are many ways to meld the hand with that minimum, we select one arbitrarily. Because of the multiplicity of unmelded cards we could discard, we next use a series of heuristic filters to increasingly narrow the candidates. The first filter handles the situation in which all eleven cards in our hand form melds. At this point, we would want to knock, but first we must choose one card to discard without breaking up any melds. If every single card in our current hand of eleven cards is part of a meld, we know that at least one meld must contain more than three cards. Therefore, if that meld is a run, we discard either the beginning or ending card. If it's a set, we can discard any card in the meld.

Our next three filters are applied in concert to determine which cards we want to keep as shown in Figure 1. The first of these ensures that we keep existing pairs together in the hand. Pairs qualify to be kept if their deadwood sum is below

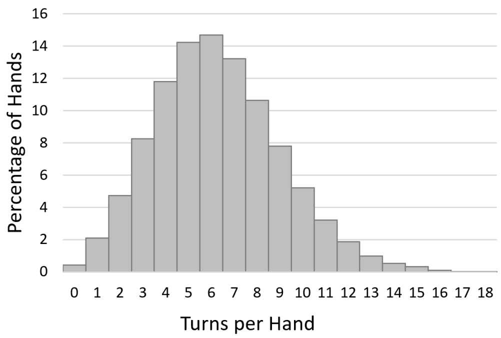


Figure 2: Distribution of hands by number of turns per hand

a certain variable threshold which decreases based on the number of turns played. For the first two turns, every pair is kept regardless of its deadwood sum. As the hand goes on, the threshold decreases from twelve on turn three to ten on turn five, meaning that fewer pairs kept. After five turns, no pairs are intentionally kept in the hand because the average hand length for SimpleAI self-play is six turns (Figure 2).

The second filter keeps cards close to existing melds for the first two turns only. We define these cards as being either two ranks less than the smallest card in a run or two ranks greater than the largest. Much like pair formation, we only keep these cards if the cards that would attach them to the meld have not been previously discarded. For example, if we have [**5♣**, **6♣**, **7♣**] and we also have either or both the **3♣** and **9♣**, we would choose to hold onto them only if it was possible to obtain either the **4♣** or **8♣**, respectively, in a future turn. After the first two turns, we no longer apply this logic because it results in holding too many unpaired/unmelded high-deadwood cards late in the hand.

The third filter keeps cards in our hand that are desired by the opponent. A card is classified as being desired by the opponent if, in the case where the opponent obtains it, it would form a meld in their hand. This filter executes on every turn. Furthermore, it does not consider the deadwood cost of keeping a card that the opponent would want.

Our final filter determines which remaining card we will discard. This process is summarized in Figure 3. We first use the list of cards remaining after the previous filters, but if this list is empty or if the only remaining card is the face up card which we just picked up, we use the original list of unmelded cards before we applied any filters. While we could naively pick the highest deadwood card remaining in the list, that causes complications. In some cases, there is more than one highest remaining deadwood card, but only one that is the best to discard.

In the case that there are face cards, we want to discard the highest-ranking face card that is not part of a broad pair. If all face cards are part of broad pairs, we discard the one that is part of the fewest. This maintains high-valued pairs rather than a collection of high-valued cards that are unpaired. This enables drawing of a face up card that could complete the meld later in the game when our opponent begins to reduce its deadwood. If no face cards are present, we discard the highest ranked card in our hand.

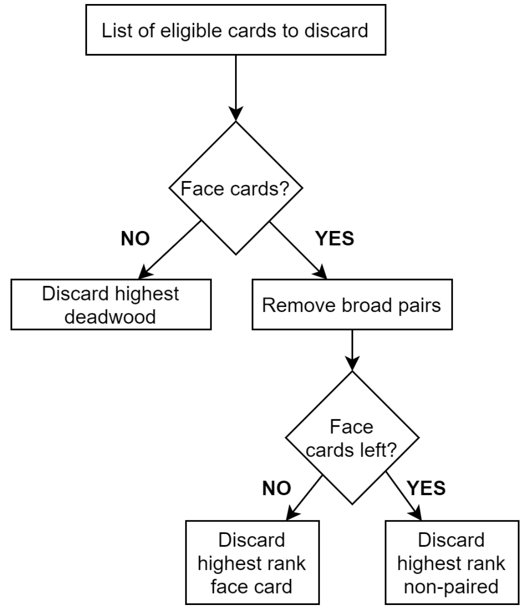
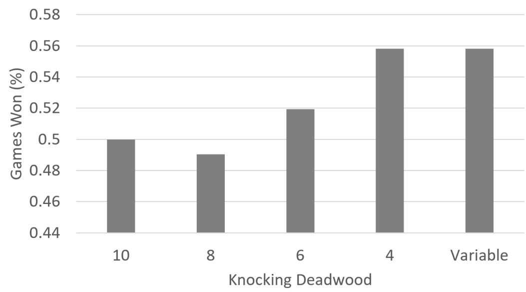


Figure 3: Overview of algorithm for discarding a face card

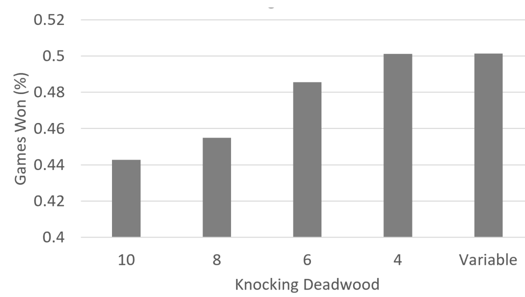
Approach to Knocking

Experts disagree about the best strategy for when to knock. Some advocate knocking as soon as possible to prevent the opponent from undercutting (MacQuaid 2020), while others suggest waiting with the hope of undercutting the opponent (Brown 2019). The second approach assumes that knocking at ten deadwood late into the hand increases the chances that we are undercut. Our strategy balances both the desire to end games soon with the desire to reduce deadwood below ten points in long hands. As we approach and exceed the estimated average game length of six turns, we gradually decrease the deadwood threshold so that we knock later. This approach mitigates the risk of being undercut by knocking at ten deadwood late into a hand while still performing well against other agents that knock immediately at ten deadwood.

To evaluate our knocking strategy, we implemented a suite of knocking fixed knocking strategies into SimpleAI: knocking a 10 (default), 8, 6, and 4 deadwood points. In addition, we implemented Tonic's variable deadwood thresholds into SimpleAI. Figure 4a illustrates the average win rate across 100,000 games in play between SimpleAI (default) and each other knocking strategy. Figure 4b illustrates the average win rate across 100,000 games in play between SimpleAI (variable) and each other knocking strategy. In general, reducing the deadwood threshold increases win rate. Further the win rate of SimpleAI using a fixed threshold of 4 deadwood points and the win rate of SimpleAI using the variable threshold were approximately equivalent. However, a fixed deadwood threshold of 4 leads to being frequently undercut by a more complex agent such as Tonic (Tonic wins against SimpleAI (4) 68% of the time). By compari-



(a) SimpleAI (Default) vs All



(b) SimpleAI (Variable) vs All

Figure 4: Win rates for different fixed and variable knocking strategies using SimpleAI

son, a variable threshold is undercut less frequently (Tonic wins against SimpleAI (variable) 60% of the time).

Opponent Hand Estimation

Unlike humans playing gin rummy, a computer has perfect memory within the limits imposed by the competition. This means all agents are able to card count and estimate the opponent's hand. One factor in the accuracy of the estimation is that most hands are only six turns long, meaning there is little data to base the estimation upon. We use the estimator for two main tasks: tracking the discarded cards and determining what cards the opponent likely has.

Every card is assigned a value in the range $[-\infty, \infty]$ which is then scaled to a likelihood estimate in the range $[0, 1]$ using a logistic function. All cards not in Tonic's hand are initially set to a value of 0 and thus a likelihood of 0.5. All cards in Tonic's hand are set to a value of $-\infty$ and thus a likelihood of 0. Likelihoods are then updated when Tonic draws a card, the opponent draws a card, or the opponent discards a card. This is adapted from the estimator discussed by AI Factory (Rollason 2007).

When the opponent draws the face up card, we set the likelihood to be one because we are certain that the card is in their hand. We also increment the values of the surrounding cards by a fixed increment amount. This includes both cards that can form a run or a set with the drawn card. To update runs, we perform an additional increment. Starting with the card two ranks lower, we add the average of the adjacent cards' values. For example, if the opponent drew a **5♣**, we would first perform the increment. Then we would average the values of the **2♣** and **4♣** and add that value to the **3♣**. We would repeat this, adding the average of the values of **3♣** and **5♣** to **4♣**, and so forth through **7♣**. This same procedure is performed following an opponent discarding a card.

When the opponent draws the face down card, we update the estimator based on their declining to draw the face up card. We assume the opponent would not decline a face up card if it formed a meld. For example, if the face up card is a **7♥**, we know that they are unlikely to have other sevens or the **6♥** or **8♥**. Thus, we decrease the value of these cards by a fixed decrement amount. To allow for the possibility of the opponent attempting to form an inner strait, we also decrease the value of the **5♥** and **9♥**, but by a smaller amount. In the case where the opponent discards a card, we set its likelihood to 0, and decrement the likelihoods of the surrounding cards.

The update amounts were established through a grid search, playing five rounds of 10,000 games each against SimpleAI and choosing the tuple of values which maximizes performance. In general, we found that an increment amount exceeding the decrement amount always led to better performance. This difference in value is because we can be certain that any rational agent will pick up a face up card that is adjacent to a card in their hand.

Often, when filtering cards that we want to keep or consider for discard, we want to know whether or not a card is in the opponent's hand. To begin each hand, we only know the specific locations of eleven cards. After this, we have a definite likelihood of 0 when we draw a card or the opponent discards. We only have a definite likelihood of 1 when the opponent draws the face up card. During play, the likelihoods of the uncertain cards typically range from 0.3 to 0.9. To determine whether or not we consider a card in the opponent's hand, we must establish two thresholds: one as the upper bound for being confident that a card is not in the opponent's hand, and the other as the lower bound at which we are confident that the card is in the opponent's hand. These values were set at 0 and 0.86 respectively, found through grid search of games played between Tonic and SimpleAI.

Conclusion

This paper describes the construction of a gin rummy agent using heuristics based on domain knowledge. To achieve this, we synthesize strategies discussed by experts and hobbyists. This synthesis requires the development of algorithmically precise models originating in vague domain knowledge. First, we approach the problem of when to draw from the deck or the discard pile. Expanding beyond conventional domain knowledge to draw from the discard in cases of completing a meld, we concluded that it is also beneficial to draw from the discard pile to reduce deadwood low enough to knock or form pairs early in the hand. We also developed a filter-based heuristic to determine the best discard of eleven cards in the hand. A key element of this heuristic is using a pair's deadwood sum to determine how far into the hand it should be kept. Finally, we decrease the likelihood of being undercut when knocking by using the depth of the current hand (how many turns have passed) to determine the target deadwood value for knocking. Using these heuristics, Tonic wins against SimpleAI at least 68% of the time.

Future Research Directions

There are many future directions this research could take from small improvements to our current heuristics and opponent hand estimation to creating a new agent using counterfactual regret minimization based on our current heuristics.

Improvements to Current Heuristics A current weak point in Tonic is that it discards an arbitrary unmelded card if no cards are left after applying all filters. In the fallback case, our strategy does not consider the opponent's hand, and may provide it with a useful card. Instead, applying a subset of filters in this case may provide more logically sound set of candidate cards for discard. Similarly, we might reorder the hierarchy of filters based on individual filter impact. Finally, we might prevent the opponent from laying off against high pairs by breaking them immediately before knocking.

Opponent Hand Estimation At present, Tonic keeps static the increment, decrement, and decline values for an opponent picking up or declining the face up card. Instead of keeping them the same throughout the hand, it is likely that hand depth also has an impact on how these likelihood estimates should be adjusted. This is also likely to have an impact on Tonic's ability to predict opponent knocking by tracking their probable deadwood. At the moment, Tonic always keeps cards in its hand that an opponent is interested in. Tonic would likely see an impact on its performance if it used hand depth to determine which card(s) the opponent is interested in and which to make available to discard.

Counterfactual Regret Minimization One technique which was suggested for the EAAI research challenge was to apply counterfactual regret minimization, or CFR (Neller 2020). CFR is a method which has recently been the focus of sequential, partial information games such as Poker (Moravčík et al. 2017b). The main challenge of applying CFR to gin rummy is that tracking each card of the deck individually would result in a combinatorial explosion in the dimensionality of the game state, with a game tree branching factor too large to reasonably compute fully. By representing the game state using experts' considerations such as pairs, deadwood at varying hand depths, and situations in which to draw the face up card, we can reduce the dimensionality to computationally reasonable levels.

References

Brown, N.; and Sandholm, T. 2017. Libratus: The Superhuman AI for No-Limit Poker. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, 5226–5228.

Brown, S. 2019. 7 Simple Strategy Tips for Gin Rummy. https://www.thesprucecrafts.com/gin-rummy-card-game-strategy-and-tips-412364 (accessed August 27, 2020).

MacQuaid, M. 2020. 8 Gin Rummy Strategy Tips to Help You Win. https://bargames101.com/gin-rummy-strategies/ (accessed July 18, 2020).

Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. 2017a. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356(6337): 508–513.

Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. 2017b. DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker. *Science* 356(6337): 508–513.

Neller, T. 2020. Gin Rummy EAAI Undergraduate Research Challenge. http://cs.gettysburg.edu/~tneller/games/ginrummy/eaai/ (accessed May 4, 2020).

Rollason, J. 2007. Predicting Game States in Imperfect Information Games. https://www.aifactory.co.uk/newsletter/2007_02_imperfect_info.htm.