Applying Equivalence Class Methods in Contract Bridge

Sean Sutherland

Department of Computer Science
The University of British Columbia

Abstract

One of the challenges in analyzing the strategies in contract bridge stems from the fact that it is a Bayesian game with a very large type space, and that players' action sets depend on their type. This results in a the strategy space being of a size where brute force methods of analysis are not practical.

We present a general approach to simplifying this strategy space through the use of equivalence clases over types. The equivalence classes are chosen in such a way that the strategy needs to conditions on less information, thereby reducing the effective size and richness of the strategy.

A common subproblem in bridge, the finesse play, is used as an example in applying this approach. Several types of finesses are described and modelled formally, along with the corresponding equivalence classe. The reduction in the strategy space as a result of the equivalence class representation is demonstrated on these examples. We then discuss possible methods of improving or extending this approach to larger subproblems in bridge.

1 Introduction

This paper deals with the problem of developing and anaylsing the strategies required in the game of contract bridge. In a similar fashion to games such as chess, go, shogi, or poker, the outcome and strategy spaces for bridge are extremely large, making a brute force analysis of the game unreasonable. Bridge and poker also have the additional problem that they are Bayesian games, and as such the full details of the game are unknown, which must be reflected in the richness of the strategy space.

We begin by describing the game of bridge, both in a natural and game theoretic setting, followed by a brief discusssion of a number of previously used approaches to analyzing bridge play. We then discuss an idea borrowed from recent developments in poker playing agents, relating to the use of equivalnce classes over opponent types, and apply it in the context of bridge.

A specific subproblem of bridge is then selected, and a model for it is formally defined. Using this model we apply the equivalence class approach and show that this can lead to a significant reduction in the effective strategy space for a player.

2 Contract Bridge

2.1 Natural Language Description

Contract bridge is a game played with 52 standard playing cards, each with a rank and a suit. The suits are clubs, diamonds, hearts, and spades (♣, ♢, ♡, ♠). The ranks are 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King, Ace, in order from lowest to highest (the last five abbreviated T,J,Q,K,A).

There are four players: north, south, east, and west (N,S,E,W), seperated into two teams: N-S and E-W. Each player is dealt a 13 card hand at random. The game is comprised of a series of tricks, defined as each player playing one card from their hand. The order in which these cards are played is clockwise, with player names interpreted as cardinal directions, starting with the lead player. For example, if E is the lead player, the player order is E, S, W, N.

The lead player may play any card they wish. The other players must play cards of the same suit as the lead player's card, if possible. Otherwise, they may play any card from their hand. The winner of a trick is the player who played the highest ranked card in the suit of the lead player's card, unless there is a trump suit. However, for simplicity it is assumed that there is no trump suit. The winner of a trick then becomes the lead player for the following trick, and this is repeated until all the cards have been played, resulting in a total of 13 tricks.

Through a process known as bidding, one player is designated as the declarer, and is assigned a contract. The bidding process is not explained as it is not necessary for the analysis in this paper. A contract consists of a suit and a level. The suit dictates whether there is a trump suit, and what it is. As mentioned above, it is assumed that there is no trump suit. The level of contract indicates the number of tricks that the declarer's team must take in order for the team to achieve a positive score. The team opposing the declarer is known as the defenders. The defending team's score is always the negative of the declarer's team's score.

A team's score is strictly increasing in the number of tricks taken, though not linear. How this score translates into a player's utility depends on the variant being played, though it is also guaratneed to be strict increasing, and both players on a team will always recieve the same utility. For the analysis done in this paper we only need to consider that a player's utility is strictly increasing in the number of tricks

that player's team takes.

The player to the left of the declarer is the lead player for the first trick. After this player has played a card (called the opening lead), the declarer's partner reveals his hand to all of the players, and no longer interacts with the game. The declarer dictates what card is played from the revealed hand, which is referred to as the dummy or the table. In this way (apart from the bidding process), bridge is a three player game, in that only three players are making any decisions. However, it is still conceptually useful to remember that the declarer's team is two distinct hands being controlled by one player.

2.2 Game Theoretic Representation

We model bridge as being similar to a Bayesian game except that a player's action set depends directly on the player's type. The game is described with four players, one for each hand, because it is simplest to do so when consider the actual mechanics of the game. The fact that a single agent is in control of two hands does not need to be considered until speaking about strategies.

The four players in the game are $\mathbb{N} = \{N, S, E, W\}$. The set of cards is $\mathbb{C} = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\} \times \{2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A\}$, with the subset of cards of suit x denoted $\mathbb{C}_x$. An agent's type (or hand) is $\theta_i \subset \mathbb{C}$ such that $|\theta_i| = 13$. The distribution over types is uniform in the sense that, given no additional information, $P(C_1 \in \theta_i) = P(C_2 \in \theta_i) \forall C_1, C_2 \in \mathbb{C}$. However, the samples for each player are very much not independant, as each card must be a member of exactly one player's type. Mathematically, we could write this as $\cap_i \theta_i = \emptyset$ and $\cup_i \theta_i = \mathbb{C}$.

Because the only action players can take is to play cards, the set of all possible actions available to players is the set of cards. Determining a specific player i's action set at node j, $A_i(j)$, is done in a number of refinement steps. First, player i must have had the card to begin with: $A_i(j) \subset \theta_i$. Second, a card cannot be played twice: $A_i(j) \subset \theta_i \backslash H_i(j)$, where $H_i(j)$ denotes player i's history of actions up until node j.

If player i is the lead player at node j, then no further restrictions are necessary and $A_i(j) = \theta_i \backslash H_i(j)$. If player i is not the lead player at node j, then assuming the lead card is of suit x, player i must play a card of suit x: $A_i(j) = (\theta_i \backslash H_i(j)) \cap \mathbb{C}_x$. If this set is empty, meaning player i no longer holds any cards of suit x, or player i is the lead player, then we remove this restriction: $A_i(j) = \theta_i \backslash H_i(j)$.

There are a few interesting consequences of this formulation. While in general player i does not know the action set of player j, he does know a subset of actions that player j cannot make. This is also directly tied to player i's beliefs about player j's type. In addition, as the game progresses player's make partial observervations on each other's types, refining their beliefs. By the end of the game, every player knows every other player's type.

As mentioned above, we do not require an explicit utility function for the analysis that is done in this paper. Even if we assume that utility is equal to team score, explaining the scoring system would unnecessarily complicate the problem. In particular, we would like to be able to speak in the context of a single trick, and the effect that it has on the score is dependant on everything else that happens in the game. Therefore we restrict our information about the utility function to be the fact that it is strictly increasing in the number of tricks taken by the team.

This representation does not encapsulate the fact that the declarer has control of two hands. Once we have the game as defined above, we can simply label all of the action nodes of the dummy as those of the declarer instead, while keeping note of which hand the node corresponds to. While this game is just as well defined, describing it in this way to begin with would have been notationally tedious.

3 Past Work

3.1 Sampling Methods

In the past, a number of bridge playing agents were developed based on a sampling approach [Ginsberg 1999]. The basis for this method is to first develop a method to find an optimal strategy in the case where every player's type is known. This is often referred to as "double dummy" bridge, as it is equivalent to both teams having a dummy, and can be solved reasonable quickly. Then, at each trick, opponent types are sampled from the type distribution, with biases based on information gained so far. The optimal play is determined for each of the samples, and the action chosen is the one which maximizes the expected utility based on these samples.

One of the drawbacks of this method is that the quality of the approximation depends on the number of samples taken, and the amount of computation required is directly proprotional to the number of samples. Another drawback is that Monte Carlo methods such as this tend to avoid information gathering actions, causing difficult or risky decisions to be deffered as much as possible. This has lead to some noticable mistakes in the play.

3.2 Planning Methods

Both of these papers speak only about how to play bridge from the point of the view of the declarer. The planning approach lends itself well to playing as declarer, as both of the team's hands are visible. As a defender, not knowing what cards your partner has can make it very difficult to predict what might happen later on, making a planning approach less effective.

In 1992, Frank et al. wrote a paper that presented a new approach to solving the problem of declarer play in bridge [Frank et al. 1992]. They hypothesized that the main reason

that human players can perform so well at declarer play in bridge despite its complexity is that each hand can be broken into subproblems, which, when solved individually, can be linked together in an attempt to find an optimal strategy. Then the main difficulty lies within identifying which subproblems to solve, and how they are related. Their method uses ideas from the area of automated proof generation for mathematical theorems, which characterizes a similar form of problem.

A few years later, Smith et al. published a paper on another planning-based technique [Smith et al. 1996]. The approach is task oriented, using hand crafted tasks with specific pre- and post-conditions to represent the choices that the declarer can make, rather than specific actions. The method then uses decision theory and planning techniques from the literature to find an optimal path through the game. The techniques had to be modified to accomodate for uncertainty in the system based on card distributions and opponent strategies.

3.3 Approximate Optimality

In 2003, the University of Alberta Poker Research Group published a paper about approximating optimal behaviour in poker playing agents [Billings et al. 2003]. While this is not a paper on bridge, some of the ideas presented are nevertheless relevant. In particular, the authors discuss the idea of partitioning the set of possible hands into equivalence classes, based on a sense of strategic similarity. Hands are said to be strategically similar if they can be played with the same strategy and yield a similar outcome or utility.

Using this, approximately optimal strategies can be found more easily, because the strategy space is reduced drastically. This direction of thinking is particularly important in card-based games, where considering all possible combinations of cards is generally computationally infeasible.

The planning papers take on a similar approach, as they avoid having to explicity consider strategically equivalent actions by encapsulating strategies into tasks and only considering the relationships between these tasks.

4 Problem Description

In this paper we only consider the problem of playing as the declarer. As stated above, playing from the point of view of the declarer makes predicting possible partial outcomes much simpler, as the trick taking capabilities of the team are entirely known. However, the other reason this restriction is made is that defender strategies have an additional complication: communication.

It is allowed within the rules of bridge that actions relay information to one's partner, as long as: 1. The action is legal within the game, 2. The meaning of an action is agreed upon beforehand, and 3. the opponents are fully aware of this agreement (all easily enforcable from a game theoretic point of view). Since there is no partner for the declarer to relay information to, this is only really a concern for defenders.

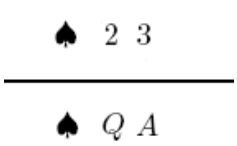


Figure 1: An example of when a finesse play can be used.

While declarer can still consider what information is being relayed by defenders' actions, this is only a reactionary consideration, used to refine the type distributions. On the other hand, this can have a large strategic impact for a defender. An action must be considered not just for its impact on the game itself, but also for what information it may relay to his partner. While building this into the strategic model for defender is not an insurmountable task, it does outline another fundamental difference between playing as the declarer or a defender.

4.1 Subproblems and Equivalence

The approach described later uses ideas borrowed heavily from the ideas presented in [Billings et al. 2003], but rather than partitioning entire hands into classes, we group individual cards. Generally, cards exhibit strategic equivalence when the cards are either of a low enough rank to be unlikely to win a trick, or a single player holds cards of consecutive rank.

Determining how cards should be grouped into equivalence classes in a general situation is a daunting task, especially because the strategic equivalence of cards can change drastically over the course of the game. Instead, we turn to the idea presented in the planning methods of focusing on the analysis of a single subproblem in bridge: the finesse play. Even modelling this single subproblem is difficult to do in complete generality. We further narrow down the problem by presenting models for a few specific forms, and later discussing some steps that could be taken towards generalization.

4.2 The Finesse Play

The finesse play is a technique used to win a trick using a card that is not the top ranked card in that suit. The basic idea of a finesse is to take advantage of the fact that players must play cards in a certain turn order. Therefore some players will be able to condition their actions on cards that have already been played in the current trick.

Figure 1 depicts a simple example of when a finesse play can be used, where the cards above the line represent the dummy's hand, and the the cards below represent the declarer's hand. In this example, there are only two tricks left

to be played, and it is known that the opponents hold the 4, 5, 6, and K of spades.

It is evident that the A can take a trick, but it is less obvious how declarer can make the Q win as well. If the dummy is on lead, declarer can lead the 2 of spades, and then see what E plays. If E plays the K, then declarer trivially takes two tricks by playing the A and then the Q. If E plays anything else, then the choice may not be as clear. If declarer plays the Q here, then he is still guaranteed two tricks as long as E has the K.

This is called "finessing the Q," because it attempts to use the Q to win a trick by taking advantage of it's position relative to the K. It is also important that the dummy was on lead. If declarer had simply lead the Q from his hand, then regardless of who had the K the Q would not win (overtaking the Q with the K is a dominant strategy for whoever has it). For the finesse to succeed declarer needs to lead from the dummy and hope that E has the K. If it turns out that W had the K, then there is no way for declarer to guarantee winning with the Q, regardless of who is on the lead, unless the K is the only spade W holds. In this particular example, this is not possible.

Note that it is not necessary to model the defenders' behaviour in order to speak about finesses. In this example it was convenient that the defenders (in particular, W) had dominant strategies that we could assume would be followed. However, in general the defenders may not have dominant strategies, or they may be very difficult to find. Nonetheless the finesse is a valid technique to use, as the true goal of the play does not rely on the defenders' strategy: if E holds the K, using the finesse *guarantees* that the declarer will win with the Q.

5 The Model

In an effort to keep the model as simple as possible, we will only be talking about finessing within the context of a single trick. This, in addition to the fact that there is no trump suit, implies that the only relevant suit is the one being lead. For notational simplicity, we will assume that the suit in question is always spades.

First we define a relation between two cards, which we call similarity. We define the predicate $\text{similar}(x, y, S)$ to be true if the card X and the card Y both belong to set S, and every card between them also belongs to set S. For example, if declarer holds the A,K,and Q of a suit in his hand, then the A and Q are similar. This represents the notion that as far as that player is concerned, these cards are almost identical.

5.1 Equivalence Classes

We are now ready to define some equivalence classes over cards in the players' hands. The first is the class of "top cards" which we will denote $\mathbb{T}$.

$$\begin{aligned} \text{if} \quad & \spadesuit A \notin \theta_N \cup \theta_S \quad \text{then} \quad \mathbb{T} = \emptyset \\ \text{otherwise} \quad & \mathbb{T} = \{x | \text{similar}(x, \spadesuit A, \theta_N \cup \theta_S)\} \end{aligned}$$

This is the set of cards that the declarer is in control of that no card the opponents hold can overtake. If one of these is played, then the declarer or the dummy is guaranteed to take the trick. In the example above, $\mathbb{T} = \{\spadesuit A\}$ In addition, we define $\mathbb{T}_N = \mathbb{T} \cap \theta_N$ and $\mathbb{T}_S = \mathbb{T} \cap \theta_S$.

The next class we define is that of the "finesse cards", which we denote $\mathbb{F}$. This class is defined as follows.

$$\mathbb{F}_{\text{max}} = \spadesuit x \text{ where } x = \max\{y | (\spadesuit y \in \theta_N \cup \theta_S) \wedge (\spadesuit y \notin \mathbb{T})\}$$
$$\mathbb{F} = \{x | \text{similar}(x, \mathbb{F}_{\text{max}}, \theta_N \cup \theta_S)\}$$

The finesse cards are the set of higest ranked declarer controlled cards that are not top cards. In the example, $\mathbb{F} = \{\spadesuit Q\}$. The card that the declarer is finessing will be an element of this set. As above, we define $\mathbb{F}_N = \mathbb{F} \cap \theta_N$ and $\mathbb{F}_S = \mathbb{F} \cap \theta_S$.

We now define an equivalence class over some of the defenders' cards. This is the class of "critical cards", which we denote $\mathbb{C}$. This corresponds to the set of cards that are lower ranked than cards in $\mathbb{T}$, but higher than cards in $\mathbb{F}$.

$$\mathbb{C}_{\text{max}} = \spadesuit x \text{ where } x = \max\{y | \spadesuit y \in \theta_E \cup \theta_W\} \wedge \spadesuit y > \mathbb{F}_{\text{max}}$$
$$\mathbb{C} = \{x | \text{similar}(x, \mathbb{C}_{\text{max}}, \theta_E \cup \theta_W)\}$$

Conceptually, these are the cards that the declarer is concerned about when attempting to finesse one of the cards in $\mathbb{F}$. In the example above, $\mathbb{C} = \{\spadesuit K\}$. As with the classes of declarer's cards, we also define $\mathbb{C}_E = \mathbb{C} \cap \theta_E$ and $\mathbb{C}_W = \mathbb{C} \cap \theta_W$. Note that declarer has full knowledge of what cards are in $\mathbb{C}$, but not of $\mathbb{C}_E$ and $\mathbb{C}_W$ in general.

Finally, we also have the class of "low cards", which we denote $\mathbb{L}$. This is simply the set of cards that do not belong to $\mathbb{T}$, $\mathbb{F}$, or $\mathbb{C}$.

$$\mathbb{L} = (\theta_N \cup \theta_S \cup \theta_E \cup \theta_W) \setminus \mathbb{T} \setminus \mathbb{F} \setminus \mathbb{C}$$

Notice that this set is not limited to spade cards. The set is meant to represent the cards that are too low in rank to be relevant to the finesse play (at least according to this model). Therefore cards from other suits must be considered in this set, as in some sense they are the lowest ranked cards, in the context of a spade trick. Similar to before, we also define $\mathbb{L}_N = \mathbb{L} \cap \theta_N$, $\mathbb{L}_S = \mathbb{L} \cap \theta_S$, $\mathbb{L}_E = \mathbb{L} \cap \theta_E$, $\mathbb{L}_W = \mathbb{L} \cap \theta_W$, and $\mathbb{L}_{EW} = \mathbb{L} \cap (\theta_E \cup \theta_W)$.

6 Application

Let us first consider a game tree that could be used to represent the example finesse given above, depicted in Figure 2. This tree represents just the finesse strategy, as indicated by the fact that there is only one branch at each of declarer's

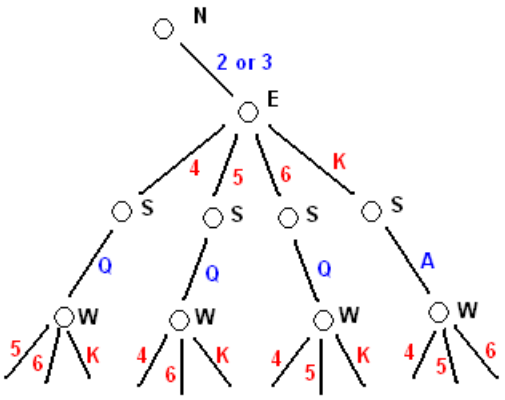


Figure 2: The strategy tree for the example finesse in Section 4.2.

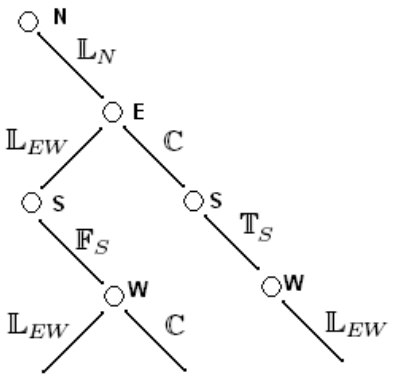


Figure 3: The same tree as in Figure 2 using the equivalence class notation.

choice nodes. While this does not reflect the actual game being played (as some of the branches do not exist), it is a good representation of the size of the strategy.

In general, the declarer is unaware of the set of actions available to a defender, but is aware of a superset of actions which are plausible. Therefore a given strategy for declarer must condition on all possible actions the defenders *may* be able to take. The branching factor of the strategy tree can therefore be quite large, even for simple problems such as the one presented above.

Instead, we can represent this tree using the equivalence clases define above, shown in Figure 3. The action (or card) corresponding to a branch is replaced with a equivalence class, indicating that the card played could be any card from that equivalence class. For example, if E plays from the class $\mathbb{L} = \{\spadesuit 4, \spadesuit 5, \spadesuit 6\}$, then declarer will follow with a card from $\mathbb{F}_S = \{\spadesuit Q\}$. This tree represents an equivalent strategy to the one in Figure 2, but with a smaller branching factor for opponent nodes.

In this example the reduction in the size of the tree is not significant. In particular, if the number of spades remaining in play was increased, then the branching factor of the full strategy tree would be proportionally larger. However, the tree in Figure 3 represents a strategy that can be used in more general situations, as will be demonstrated below.

6.1 Types of Finesses

There are many different types of play that classify as a finesse. Here we will describe three types of finesse plays and model them using the equivalence class representation described above. Each of the finesses in modelled as the dummy being on lead as the North player, but the representation for the finesse when the declarer is on lead is identical if we simply exchange the labels for each pair of players ($N \leftrightarrow S$ and $E \leftrightarrow W$).

6.1.1 Simple Finesse

This is the type of finesse shown in the example in Section 4.2. It is a finesse in which a low card is lead toward the hand with a finesse card. If the first opponent to act plays a critical card, then declarer overtakes with a top card, otherwise declarer plays a finesse card. Using the notation we have defined, we can represent this finesse as follows.

- Preconditions:
  - $\mathbb{L}_N \neq \emptyset$
  - $\mathbb{F}_S \neq \emptyset$
  - $\mathbb{T}_S \neq \emptyset$
  - $|\mathbb{C}| = 1$
- Strategy tree shown in Figure 3
- if $\mathbb{C} \subset \theta_E$ (finesse successful):
  S wins the trick
- otherwise:
  W or S wins the trick, W has the choice

6.1.2 Indirect Finesse

The indirect finesse is very similar to the simple finesse except that S does not require any top cards. The reason that this small difference warrants a different category of finesse is that in the indirect finesse, the opponents may win the trick even if the finesse is successful. We define the indirect finesse here using the equivalence class notation.

- Preconditions:
  - $\mathbb{L}_N \neq \emptyset$
  - $\mathbb{F}_S \neq \emptyset$
  - $\mathbb{T}_S = \emptyset$

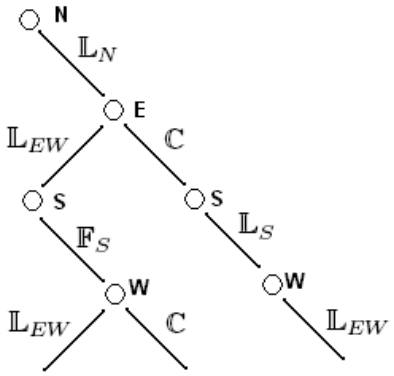


Figure 4: A strategy tree representing the indirect finesse.

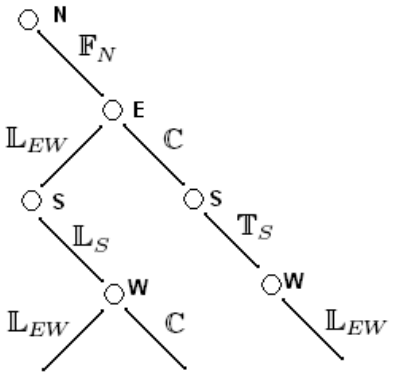


Figure 5: A strategy tree representing the high card finesse.

  - $|\theta_S| \geq 2$
  - $|\mathbb{C}| = 1$
- Strategy tree shown in Figure 4
- if $\mathbb{C} \subset \theta_E$ (finesse successful):
  E or S wins the trick, E has the choice
- otherwise:
  W or S wins the trick, W has the choice

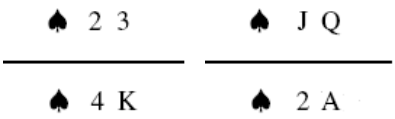
An example of when an indirect finesse could be used is shown in Figure 6a.

Notice that in the event that E wins the trick using the critical card, the finesse is still considered a success. This is because declarer's finesse card has now been "promoted" to a top card. While there is no guarantee that declarer will be able to win at a later time using that promoted card, this is an issue regarding whether the finesse should be attempted at all.

The indirect finesse is often used when not just $\mathbb{T}_S = \emptyset$, but $\mathbb{T}_N = \emptyset$ as well. It is then a method of attempting to create top cards in a suit when declarer does not have any. It is still a valid finesse when $\mathbb{T}_N \neq \emptyset$, but in this case there may also be a high card finesse to attempt from the South hand, if declarer can get it on lead by some other means.

6.1.3 High Card Finesse

A high card finesse is one in which a finesse card is led. This type of play is different from other types of finesses because even when it is successful, the finesse card may neither win nor be promoted. However, it is guaranteed that when the finesse is successful, either the finesse card being played takes the trick or all *other* finesse cards the declarer holds are promoted to top cards. The opponent has the choice between these two outcomes. As a result, for this to be of any real use to the declarer, he must be in control of at least two finesse cards.



(a) indirect (b) highcard

Figure 6: Simple examples where different types of finesses can be used.

- Preconditions:
  - $|\mathbb{F}_N| \geq 2$
  - $\mathbb{T}_S \neq \emptyset$
  - $\mathbb{L}_S \neq \emptyset$
  - $|\mathbb{C}| = 1$
- Strategy tree shown in Figure **??**
- if $\mathbb{C} \subset \theta_E$ (finesse successful):
  N or S takes the trick, choice by E
- otherwise:
  W or S takes the trick, choice by W

An example of situation in which a high card finesse could be used is shown in Figure 6b.

7 Discussion

7.1 Generalized Finesse

There are many other forms of play that can be considered finesses, and many of them are obvious extensions or generalizations to the ones presented above. This is supported by the fact that the strategy tree representations for the differet types of finesses are very similar. For example, one could imagine a finesse where both N and S might play a finesse

card. Then, depending on the relative ranks of these cards, the finesse will either behave like a simple finesse or a high card finesse.

Another example would be to relax the constraint that $|\mathbb{C}| = 1$. However, in the simple finesse case success becomes very unlikely, since we require $\mathbb{C} \subset \theta_E$. In this case, we would usually reqire that $|\mathbb{F}_S| >= |\mathbb{C}|$ and that declarer attempt multiple finesses, though this requires adapting the model to accomodate more than one trick (which are not necessarily consecutive).

It is evident that attempting to generalize the finesse strategy could also involve other extensions to the underlying model, including manipulating the equivalence class definitions, or adding more classes. While a more general finesse could have been used in the previous sections, the reasoning behind them would have become less intuitive. This would have resulted in requiring a deeper analysis into bridge strategy without adding much to the overall purpose of this paper, which is to explore the equivalence class approach.

Though the model becomes more complicated as it tries to cover a more general range of situations, it is not difficult to see that the advantage of using equivalence classes is still present. In any finesse play (and more generally, in any bridge technique), there are always cards which are key to the success of the play, and those that are not. By reducing the number of apparent actions, and thus reducing the strategy space, the task of finding the desired strategy is made easier.

7.2 Perfect vs. Imperfect Information

It is difficult to cleanly classify bridge as being either a perfect or imperfect information game. The distinction is perfectly well defined for non-Bayesian games, and so the difficulty lies in deciding which conceptual features of the definition should be carried over to the Bayesian setting.

However, a deterministic approximation of the game can be considered instead, equivalent to the strategy trees from the previous section. In this representation, opponent agents' choice nodes have actions sets corresponding to cards that they might be able to play, from the point of the declarer, regardless of whethter it is actually possible or not. While this is not an accurate representation of the game, it is useful in expressing the full richness of the strategy space of the declarer.

If the game is considered in this way, it is clear that the resulting deterministic game is one of perfect information. Each agent is always exactly aware of every action previously taken by every agent, specifying a unique location in the tree.

With this representation in mind, we can interpret the idea of using equivalence classes over cards in a new way. Grouping together cards (actions) into classes and conditioning strategies only on those classes is very similar to the notion of information sets.

Information sets generally are meant to represent an agent's inability to differentiate between nodes of a game tree. However, they could instead be used to represent an agent's *indifference* between nodes. By imposing this lack of information upon hiself, the declarer can greatly simplify the strategy space that needs to be considered.

The strength of this interpretation is that because this information restriction is self-imposed, the agent has the ability to choose how and when this information is partitioned. In the example finesse, the declarer knows exactly which cards were played once the trick is over. This information only needed to be ignored during the trick to simplify the strategy.

7.3 Global Strategy

7.3.1 Hierarchical Methods

There are many ways to approach the problem of extending the simple examples shown above into a method to simplify larger subproblems or groups of subproblems. One promising direction that has been tested in bridge AI before is that of a hierarchical structure [Smith et al. 1998].

The basis of the approach is that the full details do not need to be known for every feature or subproblem we are trying to describe. It is often the case that a number of low-level, specific pieces of information can be grouped together to form a higher level concept.

In the context of the equivalence class representation we have presented, this corresponds to broadening the classes as much as possible until refinement is necessary, in an attemptto keep the strategy tree as simple as possible at any given time.

This was already done to some extent in the examples discussed above. It is not obvious (and often false) that the cards in $\mathbb{L}$ are strategically equivalent for the rest of the game. They are grouped into the same class only because they are not relevant in the outcome of the current trick when the technique in question is being attempted. Once the finesse has been completed (either in theoretical extrapolation or in practice), then situation would have to be reevaluated, and the cards in some classes, such as $\mathbb{L}$, may be refined into new classes.

It is not only refinement that can occur, it is also the case that as the game unfolds, classes may merge. For instance, in a finesse, if the opponents play their critical card(s) then $\mathbb{F}$ and $\mathbb{T}$ would merge into a new top cards class. The idea is that the grouping of cards is done in a dynamic way so as to keep the tree simplified optimally at all times.

7.3.2 Probability

It is unlikely that it is possible to reach a reasonable solution to a Bayesian game without speaking about probabilites. Having chosen not to concretely define a utility function, it is difficult to form any arguments based on expected utility.

However, even with only the knowledge that maximizing the number of tricks is our goal, some worthwhile observations can still be made.

For instance, a decision often made by declarers is whether to finesse or to "drop," which is simply to play the top cards hoping that the defenders will have no choice but to play the critical cards due to lack of choice. In both cases, a sucess means the promotion of at least a finesse card to a winner. The declarer must compute the probability of success of each technique, while considering the consequences of failure, and come to a optimal answer on expectation.

Hierarchical type refinement can come into play here as well, as these probabilities can change every time an observation is made. If a technique is defined is robustly enough to allow for it to be aborted in favour of another technique, then it may be found that the optimal play involves attempting a technique part way, and then reevaluating whether to continue or try something different.

8 Conclusion

In this paper we presented a method of using strategic equivalence to help simplify the analysis of contract bridge. This approach takes advantage of the fact that in many situations the choice between two different actions can be largely inconsequential, at least in the specific context in which it is presented. Such actions can be grouped together into equivalence classes, potentially simplifying the strategy space of a player.

The approach is analyzed from the point of view of the declarer, where the combined set of cards that both defenders hold is known, but the specific locations are unknown. As this results in the defender's action sets having uncertainty, the actual game tree is not known. In this case, it is necessary to instead consider the strategy tree, which accounts for all the actions that a defender might be able to take.

By applying the equivalence class approach to the strategy tree representation, we are able to significantly reduce the branching factor of the tree, simplifying its analysis. The method is shown explicitly for a common subproblem in bridge, the finesse, and the benefits of the representation are evident.

While we only present a very specific application of the approach, there is much potential for generalization to larger and more complex subproblems. The reduction and simplification of the effective strategy tree is an invaluable tool regardless of the direction in which research of this field is carried.

8.1 Citations

References

ABRAHAMS, G. 1977. *Brains in Bridge*. Hodder and Stoughton, Sevenoaks, Kent, Great Britain.

BILLINGS, D., BURCH, N., DAVIDSON, A., HOLTE, R., SCHAEFFER, J., SCHAUENBERG, T., AND SZAFRON, D. 2003. Approximating game-theoretic optimal strategies for full-scale poker. *International Joint Conferences on Artificial Intelligence*.

FRANK, I., BASIN, D., AND BUNDY, A. 1992. An adaptation of proof-planning to declarer play in bridge. *Proceedings of the 10th European conference on Artificial intelligence*.

GINSBERG, M. L. 1999. Gib: Steps toward an expert-level bridge-playing program. *International Joint Conferences on Artificial Intelligence*.

KELSEY, H. W. 1968. *Advanced play at Bridge*. Hart Publishing Company, New York City.

SMITH, S. J. J., NAU, D. S., AND THROOP, T. A. 1996. A planning approach to declarer play in contract bridge. *Computational Intelligence 12*.

SMITH, S. J. J., NAU, D. S., AND THROOP, T. A. 1998. Success in spades: Using ai planning techniques to win the world championship of computer bridge. *Association for the Advancement of Artificial Intelligence*.