A Bridge Bidding Practice System

John Galatti, Robert Hackman, North Hinkle, Theresa Reese, Vanessa Simpson

Seidenberg School of CSIS, Pace University, White Plains, NY, USA
jgalatti@twilightparkrock.com
{rh75789w, nh35718w, tr77771w, vs25229w}@pace.edu

Abstract

The Bridge Bidding Practice System allows bridge players to practice and refine their bidding strategies. Users are able to practice bidding with their partners as well as by themselves. The system has been designed so that the user can create decks (sets of four hands) in which they assign rule(s) to specific players’ hands. Once a deck has been dealt, each player can place his or her bids. After the bidding phase is completed, the practicing partners can view each other’s hands. Also, statistics are generated based on the bids to determine all possible outcomes for the number of tricks won per team. These statistics are calculated using a Generic Algorithm.

Key words: Game of Bridge, Bridge Bidding Practice System.

Introduction

Bridge is a card game that is played with four players. By convention these four players are referred to as North, South, East and West. Partners sit opposite each other, thus North and South are partnered, as are East and West. Bridge is played with a deck of 52 cards. There are four suits: spades, hearts, diamonds, and clubs. The cards are ranked Ace, King, Queen, Jack, 10 down through deuce. In a traditional bridge deal the cards are shuffled and dealt out one at a time starting from the dealer’s left and continuing until the deck is exhausted. Each player then has 13 cards, which constitutes their hand.

The bridge hand is given specific attributes for valuation. The Ace, King, Queen, and Jack are considered high cards and are respectively attributed 4, 3, 2, and 1 high card points. The 10 through 2 are not attributed high card value (i.e., 0 high card points). There are a total of 40 high card points in a deck. The hand is also valued with respect to distribution – that is, the relative number of cards of each suit in the hand. The ACBL (American Contract Bridge League) uses length points with 1 point being given for five cards in a suit and another point is added for each additional card in that suit. There is an alternate method of valuing distribution which is by short suits – in this case a two card suit (doubleton) is 1 point, a one card suit (singleton) is two points, and a void is three points. Distribution is also distinguished by what is known as a balanced hand; there are three suit distributions that are considered balanced: 4-3-3-3, 4-4-3-2, and 5-3-3-2, while all other distributions are considered unbalanced.

After the cards are dealt, the bidding phase begins. Bidding is the process of the two teams making bids for the number of tricks they can take. The lowest level that can be bid is the one level – this level is a bid to make 7 tricks (the first six tricks are the book, the seventh trick is the first level above book). The suit is the denomination – higher denominations are considered higher bids than lower denominations at the same level. When suits are bid, they are the trump. The bid specifying no suit as trump (denomination = No Trump) is the highest bid at any particular level. The highest level of bidding is the seventh level, which is the bid to take all seven tricks above book (also known as a Grand Slam). One requirement of bidding in bridge is that each successive bid be at a higher denomination or level.
The dealer, say West, starts the bidding by making a bid or passing. North follows by making a bid that is higher than the previous bid or passing. East follows next, then south. This continues until there are three passes in sequence or four passes in the first round. When bidding concludes, the team making the last bid wins the contract. The contract is defined to be book (6 tricks) plus the level of the bid. Once the bid phase concludes, the play phase begins [2].

The Bridge Bidding Practice System allows users to practice and refine their bridge bidding strategies. The current system is a refinement of an earlier system [1]. This system has a significant increase in functionality, and a more efficient algorithm for validating the rules and creating hands based on those rules. The primary objectives of the refined system were to make this system user-friendly, maintainable, and extensible, while preserving its core functionality.

**Methodology**

The Bridge Bidding Practice System deals only with the bidding phase of the Bridge Game. The new system supports the standard bidding practices as defined by the ACBL Bridge Series book “Bidding – The Club Series” [2]. While not comprehensive, this book lays a common basis for the description of hands in a manner that will be understood by the majority of bridge players. The design of the Bridge Bidding Practice System is such that it can support other bidding systems. The ACBL system is being used as the minimal basis; that is, if it cannot support that system, then it is insufficient. Additionally, since it supports the ACBL system, then that system can be used as the basis for the computer generated bids in the bidding screen.

The system is divided into two parts, client-side and server-side, as described below.

**Client Side**

One of the problems with the original site was the process of creating rules. The user first had to create specific rules on a screen called “Bridge Token Rules”. If the user wanted to combine two or more specific rules into a larger rule, they would have to do so on another screen called “Bridge Rules”. Next, to assign rules to specific players and specify the number of deals to generate using that rule set, the user would go to a third screen called “Bridge Setup”. There was much room for user error in this process because user input was not regulated. In the “Bridge Token Rules” screen, for example, a user could specify a minimum value for total card points that was greater than the maximum value. In the “Bridge Rules” screen a user could combine conflicting rules without receiving an error message. Also, once a rule was created in the “Bridge Token Rules” screen, there was no way for the user to see its specification. This made it difficult for a user to distinguish between rules on the “Bridge Rules” and “Bridge Setup” screens.

Given all of these problems, our approach was to merge the three screens into one “Deck Setup” screen (see Figure 1). In this screen, the user is able to assign sets of rules to specific players and specify the number of deals to generate. This screen incorporates JavaScript error checking that prevents the user from creating invalid hands for a selected player. It also prevents the user from utilizing conflicting rule sets within a particular deal.

Figures 2-7 display all the options that can be selected for a particular player’s rule set. The user can modify the Total Points (Figure 2), High Card Points (per hand and per suit, Figure 3), Distribution Points (Figure 4), Denomination Length (Figure 5), Denomination High Card Points (Figure 6), and Specific Cards for each player (Figure 7).
Figure 1. Deck Setup page.

Figure 2. Total Points.

Figure 3. High Card Points.

Figure 4. Distribution Points

Figure 5. Denomination length.

D6.3
The Client-Side was developed using HTML design applications and graphic design tools such as Photoshop and Freehand MX. The design is focused on a user-friendly presentation of the Bridge Bidding System. This was accomplished by using drop-down menus, check boxes and radio buttons as much as possible to facilitate use and avoid invalid input. The following is a list of the software used for the design:

Development Software:
- Microsoft Front Page 2003 (HTML and PHP pages)
- Macromedia Freehand MX 2004
- Macromedia Dreamweaver MX 2004 (Design of CSS Style Sheet)
- Adobe Photoshop CS

Server-side

Figure 8 illustrates the major components of the Bridge Bidding Practice System infrastructure. The user interface for bidding and the “Deck Setup” screen is interactive and managed by the Apache Web server, which uses PHP to communicate with the MySQL database. The other major components are C programs that perform the analysis of bid hands. These C programs are scheduled through a Linux Cron task, which checks the database every few minutes for new bid hands to analyze.

The server is composed of an object tier. Figure 9 illustrates the middle object tier of the web server. The web interface communicates to this object tier.
The Middle Object tier consists of the following:
When a person logs in, a Bridge User object is created. This user is belongs to teams and has decks associated to him/her. The deck object is created when the user goes into "Deck Setup". This deck object is composed of hands and the hands have rules. Aside from the hands, the deck object also has suits, which have cards. And rules apply to the hands, which are satisfied by cards and given to the deck by suits. Once the hands are defined, the user can create a deal. This deal object is composed of dealt deals and dealt decks, which it gets by requesting decks from the deck object. The deal then has dealt hands that belong also to the bridge user. The other object associated with the dealt hand is the dealt deal bids.

Hand Validation Algorithm

In the deck setup screen, we allow people to certain a set of rules. These rules can be classified as simple and specific rules. Simple rules are easily validated because they are based on the number of cards in a deck available to the user. Specific rules, on the other hand, are more difficult to validate because if the user request a specific card (ace of spades), that card must be made available. But, a specific rule, such as high point cards, which can be made up of any of four cards in any of 4 suits (total of 16 cards). Therefore, if the user specifies a specific amount within a suit, there are 4 cards that can answer the request with values 1-4.
Figure 10 is an illustration of the Rule Validation Strategy. The database processes all requests at the same time and determines if the deck is valid or not. A deck is invalid when there is no solutions found for a request. For example, if the rules assigned to North and South do not conflict but East's does, the deck is invalid. In this case, the Points based on a previous rule which were defined to East conflict and create restrictions, which prevent the deck from being generated. This allows the user to go back and change the rules assigned to East to generate a valid deck.

In the validation of hands, a select can be performed against a set of card combinations that are valid. This hand validation also determines when the deck should be dealt and also which decks will actually generate the hands that fit the point and distribution rules. This validation process also recognizes when there is only one possible solution for a generation and provides that solution or if there are multiple solutions, it provides them randomly.

Since we are using all the possible patterns for point cards distribution, we know there are enough variants. This way when a solution is returned the number of variants are tracked just for information and stored with the validation. It is important to state that this hand validation algorithm doesn't guarantee the distribution. When a hand is dealt, one card can happen in any of 13 places and in terms of distribution this makes a difference. This does not consider the order in which the cards are dealt. What can be done to guarantee distribution is to place a weight associated with a particular solution. In other words, out of the 65,398 possible point solutions, this form occurs how often. This way, we are able to determine not only if it is a possible solution but also if you selected according to its weight compared to other possible solutions. At this point, we can determine what the probably deals are rather than what are solving now which are the possible deals.

When validation is complete, the select set of cards is assigned to the specified player. The remaining cards (broken down by suits) that weren't specifically required of the deck is available to randomly distribute to players to complete their hands.

When the user goes in to "Deck Setup", a new deck is created with 4 hands that have no restrictions on them. As the user adds rules to each hand, it will pass through validation to determine if the request can be granted. Once the deck is validated, then it can be dealt with the amount of hands requested by the user.

![Diagram](image)

**Figure 11**

Figure 11 is an illustration of the “Deck Setup” process. This method of generating deals was selected because of the sheer volume of possible deals, which is approximately 56 octillion. For instance, if a user requests a specific hand, they may or may not receive it, but if the hand is fairly common they might receive it rapidly. But let's suppose the users requests a very rare hand that maybe can occur about 1,000 times, it could take a long time to receive that hand. This hand may never be generated. For this reason, we have developed a process that first finds the possible
hands that will solve the request and then randomly picks the hand out of a set that we know will solve it. This new process provides a solution that we know will come back.

**Server-side Hardware and Software**

**Hardware:**
An x86 or x64 Server with a minimum of 512MB ram. Production scaling tests have not been performed.

**Software:**
Linux: The application uses LAMP to maintain the ability to support the analysis executables that run outside the web site environment.

Linux: The development was done on OpenSuse 10.1, but any current Linux distribution should work.

Apache: Version 2.2, Pre-Fork

MySQL: Version 5.0


**Backend Development Software:**


Sun Java 1.5_0_9: http://java.sun.com/javase/downloads/index.jsp

PHPEclipse 1.1.8: http://sourceforge.net/project/showfiles.php?group_id=57621

PHP Dbg 2.13.1: http://sourceforge.net/projects/dbg2/


For development, PHP 5.1.3 and the required modules were recompiled with debug enabled. This combined versions of PHPEclipse, Eclipse, Sun Java, and PHP Ddbg allows for interactive debugging of the php code on the server. The header files for the Pre-Fork version of Apache that you are using will be required – most Linux distributions support header file installation through their integrated setup tool (YAST, etc).

**Case Study**

The following is an example of a user interacting with the “Deck Setup” screen to create a Rule with an exact number of "Denomination High Card Points" for any of the following "Major", "Minor", "Spades", "Diamonds", "Hearts", and "Clubs" you would like the selected player to receive.

**Main Success Scenario:**

1. User North clicks on "Deck Setup" Screen

2. User North selects "Create New Deck"

3. User North completes "Step 1: Create a name for this deck" by inputting a name in the text field.

4. User North proceeds to "Step 2" and selects the "Player" he/she would like to create a rule for from the drop down box entitled Player"

5. User North selects "Denomination High Card Points" by clicking the "Select" button next to "Denomination HCP"
6. The denomination length which User North can select from is between 0 - 20 for "Major" and "Minor" and 0-10 for each of the following "Spades", Diamonds", "Hearts" and "Clubs" User North selects "Exact Number" he would like the system to generate to the player

7. User North Selects "Done"

8. User North selects the number of deals he/she would like to generate from the "Create Deals" drop down box.

9. User North saves the newly generated rule by proceeding to "Step 4" and selecting "Save Deck"

**Conclusion**

The newly developed Bridge Bidding System is more user friendly than it predecessor. The rules and Deck Setup screen were a significant improvement over the original system’s three screens. The system can benefit from additional functionality, such as “new user registration”, “modify user profile”, and “create teams”.

**References**
