Practice Deals & Opening Leads

Pete Matthews Jr – https://3nt.xyz – © February 29, 2020

Dealmaster Pro can be used to generate deals to practice bidding conventions and for bridge simulations. This is an aged program with a quirky graphical user interface and a questionable dealing algorithm. I have used it to study many types of problems, but some situations are difficult or impossible.

The Deal program by Thomas Andrews is a more powerful and free alternative. It does require writing computer programs in the Tcl language (pronounced "tickle"), possibly making it more difficult to use.

This article describes using Deal and other tools to deal hands for bidding practice and to study opening lead problems.

Required Software

First, obtain and install this required software on a Windows computer:

- ActivePerl, free at http://www.activestate.com/.
- Deal, free at http://bridge.thomasoandrews.com/deal/.
- SPIRAL.tcl & practice.pl, free at https://3nt.xyz/bridge.htm#Articles, associated with the current article.
- Vim (optional), free at https://www.vim.org/download.php.
- Cards Font (optional), free at https://www.pagat.com/com/cardsttf.html.
- Bridge Composer (BC), free trial at bridgecomposer.com. If all you want is rudimentary printing of practice deals, that can be done in Deal.

Perl and Vim are native to Linux, and Deal works fine there. However, BC is Windows-only, and this article assumes you are working on Windows.

ActivePerl: Download the latest version of ActivePerl from https://www.activestate.com/products/perl/downloads/. (I'm running version 5.28.1.) Perform a "typical" install.

Deal is distributed with the runtime library for Tcl 8.5 (tcl85.dll), and that is what Deal requires. Any other installation of Tcl appears to be irrelevant to Deal, but ActiveTcl is available if you want it.

Deal: Download the Windows binary release (actually a DOS program) from http://bridge.thomasoandrews.com/deal/downloading.html. Version 3.1.9 has been current since 2013 or earlier, so what you get will probably be the single file deal319win.zip. Put it in a safe place, and unzip the file there, creating a **deal319** folder.

Copy the deal319 folder to your Documents folder, and rename it to Bidding Practice (**Documents\Bidding Practice**). All files used with Deal, and all components of Deal itself, should be in this folder.

practice.pl, SPIRAL.tcl & KANTAR.tcl: place into the Bidding Practice folder.

Vim: Download and install the Vim text editor, the modern version of the old Unix text editor. Even if you wind up editing with Notepad, Vim can be useful because it understands Tcl, color coding various entities on the screen.

Cards Font: Download and install the Cards font, if you want Bridge Composer (or MS Word) to be able to use outline (hollow) heart and diamond symbols. This is best for documents that will use mostly black print. I don't know which offends me more, solid black heart and diamond symbols, or solid red heart and diamond symbols with mostly black print.

Bridge Composer: Download and run the installer file, which would currently be called Bridge.Composer5-77.msi. This is a great tool that I use all the time to write about deals, to create random deals for dealing machines, to perform double-dummy analysis, to run bridge simuations with GIB or WBridge5 software, and in this case, to print curtain cards for practice bidding. Here is how I configured the defaults in BC:

File > Page Layout
Change Left from 1 to 0.5.
Change Right from 1 to 0.5.
Change Top from 1 to 0.5.
Change Bottom from 1 to 0.5.
OK

Format > Board Layout > Card Table
Change the Fill color from 008000 (loud green) to cccccc (gray).
Change the Border color from aaaaaa to 999999 (darker gray).

Format > Board Layout > Pips (tab)
Change heart and diamond from FF0000 (red) to 000000 (black).
Change Pip font from Symbol to Cards.

Format > Board Layout > HTML (tab)
Change the Pip Rendering from Entity to Native for ⦿ Spades.
Repeat for ⦿ Hearts, ⦿ Diamonds and ⦿ Clubs.

OK

Format > Layout > Save as Default

Generate Practice Deals

Create a NAME.tcl Script in the Bidding Practice folder

Create a NAME.tcl script specifically for this problem. Choose your own NAME, and capitalize it for consistency with the practice.pl script. Use the Tcl language and tools from the Deal package, as documented in the html folder in your Bidding Practice folder (or on the web). https://www.tcl.tk/man/tcl8.5/ is the manual for this version of Tcl.

Deal comes with some examples for NAME.tcl. Deal will generate a random deal and call the `main` procedure in NAME.tcl, which will either `accept` or `reject` that deal. Dealing continues until the requested number of deals has been accepted.

The `write_deal` procedure in the script writes each accepted deal. When using practice.pl, it writes the deal to NAME.txt, one line per deal. practice.pl then reads that file and creates a NAME.pbn file for use with Bridge Composer. Here is a trivial TEST.tcl file that accepts every deal generated:

```
#
# 2020-02-06  PDM  TEST.tcl
#

#
# write each deal on one line of PBN, South (S:) first, clockwise
#
proc write_deal {} {
  foreach hand {south west north east} {
    set fmt($hand) \
        "[$hand spades].[$hand hearts].[$hand diamonds].[$hand clubs]"
    }
  puts "\[Deal \"S:$fmt(south) $fmt(west) $fmt(north) $fmt(east)\"\]"
  }

#
# Start Here
#
main {
  accept
  }
```

The `write_deal` procedure shown above must be included in NAME.tcl for practice.pl to work correctly for bidding practice.

The SPIRAL.tcl file is a working example displaying my rudimentary Tcl skills. It generates deals to be used to practice Spiral Raises. A set of such deals, ready for printing, is posted with my “Semi-Natural Spiral Raises” article at https://3nt.xyz/bridge.htm#Articles. Perhaps DM Pro could have dealt these hands, but I have no idea how.

Open a Command Prompt Window (DOS Box)

Change into the Bidding Practice folder:

```
cd  "Documents\Bidding Practice"
```

Run the Programs

Try to generate just two deals, at the start, for example, using SPIRAL.tcl:

```
perl  practice.pl  SPIRAL  2
```

practice.pl uppercases the name (SPIRAL here). All output files will use that name, so use an uppercase name for NAME.tcl to be consistent. When the programs run to completion, it prints:

```
#ARGV=1
  TCLFILE=SPIRAL.tcl, WORKFILE=SPIRAL.txt, Count=2
  WORKFILE=SPIRAL.txt, OUTFILE=SPIRAL.pbn
  Board=2
SPIRAL.pbn complete.
```

practice.pl has built-in help that is displayed by “perl practice.pl” alone.

Debug the Programs

SPIRAL.tcl is sprinkled liberally with debugging statements. There is definitely a more elegant way, but this worked for me. Later, once I got the script working, I “set DEBUG 0” in three places, and all the debug printing disappeared. “set DEBUG 1” would turn it back on.

I’ll have to leave debugging to you. Let me suggest, however, that if the Tcl program executes the wrong code, or skips code, you probably have mis-matched curly braces. Color-coding in Vim can help with this and other syntax issues of a similar nature. Put the cursor on one brace, and it will highlight the matching brace, if it is on the screen.

Check the Results

```
type  SPIRAL.txt
```

This intermediate file should contain the deals, one per line. See if the number of deals is correct, and if they actually meet your needs. The hands are listed South, West, North, East on each line.

Print the Results

Open SPIRAL.pbn with Bridge Composer. If the deals meet your needs, change to Curtain Cards:

- File > Page Layout…
  Change Layout from whatever to Curtain Card
  OK

- File > Print – or – File > Save As PDF…

and print the results. Cut each sheet in half, vertically. If you print one-sided, the left side has the North and East hands, the right side, South and West. If you print two-sided, the left side will have North and East on the front, South and West on the back. This turns out to be just fine for switching positions every five deals. Staple each group of sheets in the upper left corner.

Now print a hand record as well:

- Tools > Double Dummy All Boards
- If necessary, enter the NAME (SPIRAL, for example) in the Event field.
- File > Page Layout…
  Change Layout from whatever to Hand Record
  OK
- File > Print – or – File > Save As PDF… [choose a second file name]

Print this two-sided and staple in the upper left corner.

Alternative Methods

I'm much better at Perl than Tcl, so I wrote practice.pl in Perl. I adapted it from a script I used in 2013 when running a large set of simulations with Deal.

It should be possible to rewrite it in Tcl (practice.tcl), but there is little reason to do so. It may also be possible to merge that function into the NAME.tcl script, but then those bits (not just `write_deal`) would have to be included in every such script. There is value to the separation.

Opening Lead Problems

Required Software

We need mostly the same tools to study opening lead problems. The NAME.pbn output file is now optional, and the NAME.txt file contains the desired results. The updated `write_deal` calculates the number of tricks that declarer can take (double dummy) for each opening lead specified, and appends those numbers to the line for that deal.

Microsoft Excel is used to extract these results from NAME.txt and tally them. This could also be done by another spreadsheet application, a Perl or Tcl script, or by a shell script under Linux or Cygwin.

Bridge Composer is needed only for easy verification that the deals are suitable, and possibly to write up the results.   KANTAR.tcl solves an opening lead problem. https://3nt.xyz/bridge/session_notes/2020-92029_KANTAR.pdf shows how I wrote up the KANTAR results with BC.

Create a NAME.tcl Script in the Bidding Practice folder

You'll want to start with a copy of the KANTAR.tcl script for your lead problem; give it your own name. It contains the `write_deal` procedure you will need, shown on the next page. Change the values of `lead` and `trump` for your problem.

Near the top of your NAME.tcl script, "stack" your West hand for all deals:

```
# stack the west hand
west is A653 KJ753 9 832
```

Deal will used these cards for West and randomly distribute all others.

`write_deal` in the script writes each accepted deal. However, before it does so, double dummy analysis is performed on the deal for each of the opening leads specified. (Due to my inexperience with Tcl, I put this code in this location to avoid possible scope problems; that is, to assure variables can be seen where needed.)

Before `write_deal` writes the line for a deal to NAME.txt, it appends to that line the number of declarer's double dummy tricks for each specified lead.

Run, Debug and Check the Programs

Use the methods above to run the programs, for ten deals:

```
perl  practice.pl  KANTAR  10
```

5,000 deals are necessary for accurate simulation results. This takes almost ten minutes on my relatively fast desktop computer. If your computer is slower, it could take much longer. To speed things up, we use the –n switch to tell practice.pl not to create KANTAR.pbn:

```
perl  practice.pl  -n  SPIRAL  5000
```

```
#
# write each deal on one line of PBN, South (S:) first, clockwise
#
proc write_deal {} {
#
# DDS is set up and called here to avoid potential variable scoping issues
#

# Declarer always South, leader always West.
# Set these for this specific problem:
# lead = the list of opening leads to be analyzed
set lead {"5H" "9D"}
# trump = clubs, diamonds, hearts, spades or notrump
set trump "spades"
# 1 = debug printing, 0 = none
set DEBUG 0


# required initialization
set result ""
set eff "-noreuse"

  foreach hand {south west north east} {
    # format the hand for PBN
    set fmt($hand) \
        "[$hand spades].[$hand hearts].[$hand diamonds].[$hand clubs]"
    }

  # set up the full hand diagram
  set dgm [full_deal]
  if {$DEBUG} {puts stderr "   diagram=$dgm"}

  # calculate the double dummy result for each lead
  foreach card $lead {
    if {$DEBUG} {puts stderr "   trump=$trump eff=$eff, card=$card"}
    set tricks \
      [dds $eff -leader west -diagram $dgm -trick $card south $trump]
    # append the result to string
    set result "$result $tricks"
    # reuse data for same deal and strain (better efficiency) [skipped]
    # set eff "-reuse"
  }

  puts \
  "\[Deal \"S:$fmt(south) $fmt(west) $fmt(north) $fmt(east)\"\]$result"
}
```

Just let it cook. When it’s done, check the results again. Our example, KANTAR.txt, begins with:

```
[Deal "S:KQJ74.A62.J7.KJ6 … T92.T4.AQ2.AT975 8.Q98.KT86543.Q4"] 11 11
[Deal "S:KJ842.9.AQJ72.K6 … T97.T642.K3.AQJT Q.AQ8.T8654.9754"] 10 11
[Deal "S:KQT72.9.K765.AQT … J84.AT64.J4.KJ64 9.Q82.AQT832.975"] 10 10
[Deal "S:K9842.4.AKJ63.A7 … QJT.AT2.T42.KT94 7.Q986.Q875.QJ65"] 11 11
```

```
[Deal "S:QT982.Q8.AK52.KJ … J74.AT9.Q87.A975 K.642.JT643.QT64"] 11 10
[Deal "S:K8742.QT4.AK8.QJ … QJT.2.QJ43.AT754 9.A986.T7652.K96"] 9 8
```

where "…" is the stacked West hand, `A653.KJ753.9.832`.

We see that on some deals, the opening lead makes no difference to the double dummy results: on the first deal, declarer can make 11 tricks on either opening lead.

On the second deal, declarer can be held 10 tricks on the five of hearts (5H) lead, but can make 11 tricks on the nine of diamonds (9D) lead (as specified by the value of the `lead` variable in KANTAR.tcl).

If no deals show a difference in results for the opening lead, then either there is a bug in the scripts or the problem is insignificant. For example, it's unlikely that the fourth best or fifth best lead of the five or three of hearts would make any difference.

Open Excel and Bring in the Results

Right-click in the Bidding Practice folder > New > Microsoft Excel Spreadsheet

Rename the file with NAME, KANTAR.xlsx in this case.

Open the file, and select cell A1

Data > From Text > Browse to the Bidding Practice folder

Select KANTAR.txt > Import > ⦿ Delimited > Next

Delimiters ☑ Space > Next

Check numbers are in columns > Finish > ⦿ Existing Worksheet: =A1 > OK

The deals are irrelevant; if the numbers make sense, save the file. If not, Ctrl-Z and try again.

Now scroll down to line 5000 (the number of deals). Enter titles under the deals and enter the formulas both next to them and below the first column of numbers:

5001	Total Tricks for Defense	=65000-SUM(C1:C5000)
5002	Average Tricks for Defense	=C5001/5000
5003	Lead Sets 4S Contract	=COUNTIF(C1:C5000,"<10")/5000

65,000 is the maximum total tricks for 5,000 deals. We subtract the total tricks for declarer from that, to get total defensive tricks. The average tricks for the defense is the key statistic for a normal contract played at matchpoints.

The third formula displays the fraction of times that the lead will set the contract (declarer can be held to fewer than 10 tricks), the key statistic for IMP scoring or total points.

Right click the 5002 cell > Format Cells... > Number, 2 Decimal Places > OK

Right click the 5003 cell > Format Cells... > Percentage, 1 Decimal Places > OK

Then select rows 5001 to 5003 for all the other columns of numbers, press the Ctrl key, and hit R. This will duplicate the three formulas to the right, into the other columns. Here is what this did in KANTAR.xlsx:

Total Tricks for Defense	12814	14540
Average Tricks for Defense	2.56	2.91
Lead Sets 4S Contract	13.1%	26.8%

All that remains is to copy and paste the results into the file or email where you want them, and add headings for the opening lead (H5, D9).

A repeat run of the simulation would use a different collection of 5,000 deals. The results might differ, but only slightly.