

Bridge Solver Online - Interface

By: John Goacher

1. Overview

Bridge Solver Online (BSOL) is a web based interactive double dummy solver application which utilises Bo Haglund's double dummy solver module (DDS). The user interface of the application is written in html and javascript, so can run on any device with a modern web browser. It communicates with a server based cgi component, hosted on dds.bridgewebs.com, that embeds the DDS library.

BSOL is invoked to "play" a hand via an http GET request with a number of parameters, representing the board number, deal to be played, the dealer, vulnerability etc. The target of the request can be a separate browser tab or window, or an iframe within the current browser window. Any subsequent user interactions within BSOL such as clicking "Analyse" to calculate makeable contracts and optimum contract(s)/score, selecting a declarer/suit to play, or playing a card, is handled within BSOL via ajax requests to the cgi process.

Clicking on a declarer/suit combination to play the contract generates a request to the cgi that establishes a new play session and calculates the number of makeable tricks for each possible opening lead. Clicking on the "Stop" button, or clicking on another declarer/suit combination in the makeable contracts box, will terminate the current session. If the user navigates away from the page or closes the browser window without terminating the session, or does not continue playing the hand, the session will be timed out about 20 minutes after the most recent user action.

2. Software Interface

BSOL is invoked by an http request of the form:

```
https://dds.bridgewebs.com/bsol2/ddummy.htm?<parameter string>
```

Some parameters are compulsory and others are optional. All parameters are validated. A validation failure of one of the compulsory parameters will cause a javascript alert describing the error and BSOL will not be invoked. A validation failure of one of the optional parameters will not be reported, and the parameter will not be passed on to BSOL.

All parameters names may be upper or lower case.

2.1 Method 1 - Including all the hand details as explicit parameters

2.1.1 Compulsory Parameters

board

The board name, which will normally be a number. However, it can be any character string (arbitrarily limited to 15 characters).

dealer

Must be one of N,S,E,W (upper or lower case)

vul

Must be one of NS, EW, All, or None (upper or lower case)

north, south, east, west

These four parameters hold the dealt cards for each player for this board. Each parameter string holds the cards for all four suits, in the order Spades, Hearts, Diamonds, Clubs, with the individual strings separated by a "." character. The cards within a suit are represented by their face values with a 10 being denoted as "T". An example would be:

Q853.KJT82.K5.J5 or .KJT876542.K5.J5

Checks are made to ensure that all four hands are present, that each hand contains 13 cards, and that no card is duplicated in the deal.

club,event

Although BSOL will work if these two parameters are missing, they should be included in order to allow usage statistics to be gathered, and for diagnostic purposes in the event of any problems. The values are not validated but will be recorded in the apache log entries.

example: **club=acblunit132&event=20150707_1**

2.1.2 Optional Parameters

dd

If present this parameter contains double dummy tricks information, i.e. the number of makeable tricks for this board for each declarer/suit combination, a total of 20 values. These are represented by 20 hex digits in the range "0" to "D" (upper or lower case). This information is normally sourced from the PBN file produced by the program that generated the random deal. However, the representation of this data within PBN files is not rigorously defined. Some files only show number of tricks information for makeable contracts (i.e. 7 or more tricks), and may show 0 or 1 for non-makeable contracts. BSOL will assume that the values 0 and 1 are genuine values if any of the 20 positions contains a trick count in the range 2 to 6 inclusive, otherwise values 0 and 1 are assumed to mean "unknown" and the corresponding

positions in the BSOL makeable contracts panel will be displayed as a "-" character (or a "*" if the user has chosen to display number of trick in the panel rather than makeable contracts). The dd parameter value may also contain the "-" or "*" values to explicitly indicate "not makeable" or "unknown " respectively for a particular declarer/suit combination.

Within the dd parameter the values appear firstly for north for NT, Spades, Hearts, Diamonds, Clubs in that order, followed by the corresponding group of 5 values for each of the south, east, west positions in turn. Thus an example of a complete makeable contracts string is:

```
3413534135a9ba899b98
```

In this example the makeable contracts are:

East: 2 Clubs, 4 Diamonds, 5 Hearts, 3 Spades, 4 NT

West: 2 Clubs, 3 Diamonds, 5 Hearts, 3 Spades, 3 NT

If the dd parameter is omitted, BSOL will display an "*" in each position in the makeable contracts panel.

optimumscore

If this parameter is present it shows the optimum contract(s) and/or optimum score for this board. This parameter is simply a text string and no validation is performed. It is displayed in the top left quadrant of the BSOL display.

Some examples of the optimumscore representation are:

```
EW 6S; -980
```

```
W 2H,EW 2D+1; -110
```

```
W 2H,EW 2D+1; -110
```

```
NS 3D+1,NS 3C+1; +130
```

The polarity of the score is always shown from the point of view of NS, even if it's an EW contract. Occasionally it's possible that NS and EW can both make the same optimum contract score (e.g. they can both make 1NT depending which sides it first). In that case the NS and EW strings should be supplied with a "
" in between to force them to display on two separate lines.

contract and **declarer**

These two optional parameters can be used to supply the actual declarer and contract played in a session, as recorded on the scorecard for a particular pair of players. If these optional parameters are present the corresponding button in the

BSOL makeable contracts panel is displayed bright yellow. If the user elects to play that contract in BSOL the contract originally played will be displayed in the top right column of the BSOL display, underneath the makeable contract.

A valid contract is a digit in the range 1 to 7 followed by one of S,H,D,C,(N or NT). It may be doubled or redoubled, and the double character can be "x", "X", or "*". The parameter string may be upper or lower case.

A valid declarer is one of N,S,E,W (upper or lower case).

leadcard

If contract and declarer parameters are supplied then leadcard may optionally be supplied. If present, and the user opts to play the same/declarer suit combination in BSOL, the lead card will be marked with an "*" when the opening lead position is displayed in BSOL. The parameter value may be in upper or lower case and is a two character field represented in the UK convention with face value first followed by the suit, e.g. 3C. The face value is one of "23456789TJQKA", and suit is one of "CDHS".

title

The optional title parameter value is a text string. No validation of this string is performed. If present it will be displayed above the board in the BSOL display frame or window. It may contain embedded html markup.

analyse

If this parameter is present and is set to the value "true" (upper or lower case) then BSOL will automatically call the ddummy cgi to calculate makeable contracts and optimum contract(s)/score after displaying the board. This function is performed in any case if the hand definition contains no makeable contracts information. However, it may be useful to include this parameter if the hand definition does not contain the optimum score field, or if the makeable contract information is incomplete (i.e. only contains the value of makeable tricks for declarer/suit combinations where the number of makeable tricks is 7 or more, and contains the value 1 or 0 for all other cases).

The results of the analyse function are cached on the server for a number of days so that the calculation does not have to be repeated if BSOL is subsequently invoked with the same hand.

2.2 Method 2 - Calling Bridge Solver Online with a lin parameter

It is possible to call BSOL with a BBO LIN formatted string (i.e. lin=string-in-BBO-LIN-format). This can be used to supply all the relevant information for a single board - player names, board number, the deal definition, dealer, vulnerability, a bidding

sequence, and a played card sequence. When the lin parameter is included, any other parameters except for **title**, **club**, and **event** are ignored. Makeable contracts and optimum contract/score will be calculated automatically providing that all 52 cards are defined in the deal.

A benefit of this method of calling BSOL is that it allows inclusion of a bidding sequence and a suggested line of play (complete or partial). The user may step forwards and backwards through the suggested line of play or may choose to play a different card at any point. The user may also choose to play a different declarer/suit combination, in which case the recorded line of play is not available.

Although a lin string can be coded manually it is easier to either download the lin file of a played hand from BBO, or use the BBO Hand Editor to enter all the information.

2.3 Method 3 - Calling Bridge Solver Online with a file parameter

As an alternative to the above methods it is possible to supply the url of a single PBN, DLM, or bridgebase online LIN file. BSOL will then load all the boards from that file. If the board number parameter is also included BSOL will initially display that particular board, otherwise it will display the first board in the file. However, it is possible to navigate between boards via an additional "GoTo..." button that will appear alongside the other buttons at the bottom of the BSOL display.

In addition to the file parameter and (optionally) the board parameter, the event and club parameters should also be included for stats collection and diagnostic purposes.

An example of calling BSOL in this way could be:

```
https://dds.bridgewebs.com/bsol2/ddummy.htm?  
board=3&club=myclub&event=20160223_1&file=https://www.myclub.com/  
pbn_files/20160223_1.pbn
```

There are three important considerations:

- the name of the file must end with the .pbn, .dln, or .lin extension
- if the url of the file contains any characters which have special significance in a url, e.g. '?' and '&', then the url of the file must be encoded. For example, in javascript this can be performed using the standard function encodeURIComponent.
- the file must have the appropriate access controls to permit it to be accessed from javascript loaded from a different domain. This can be achieved by putting a .htaccess file in the directory containing the pbn/dln/lin files, providing that the web server is configured to allow use of .htaccess files to override default settings.

The .htaccess file should contain the following lines:

Header add Access-Control-Allow-Origin <https://dds.bridgewebs.com>
Header add Access-Control-Allow-Headers "origin, x-requested-with, content-type"
Header add Access-Control-Allow-Methods "GET"

Notes:

i) The LIN file format is undocumented, and therefore BSOL's decoding routines have had to be developed by reverse engineering the format from examples of LIN files. This decoder works on files downloaded from Bridge Base Online (Vugraph Archives, Tournament Archives, and Hand Records), but has been known to fail on old LIN files from other sources.

ii) BSOL can display the bidding and card play sequences contained within LIN files. PBN files can also contain this information, but BSOL currently ignores it. A workaround is to convert the PBN file to a LIN file using BBO's software, downloadable at https://online.bridgebase.com/intro/installation_guide_for_bbo.php