Binky Points: Optimal Evaluation

Introduction

In the crudest forms of bidding, the goal is to add the value of your hand to the value of partner's hand and determine what you can make. You make these evaluations independently, sum them together, and that directs you to the appropriate level.

That is, we seek to find a valuation, v, such that:

v(south)+v(north)

corresponds to the number of tricks north and south can take.

Evaluations

As with my prior evaluation article, we can seperate out notrump and suit evaluations, as they are quite different. We can also examine offense and defense, (which is fine, but offense is the most interesting, from a bidding point of view.)

Now, let's say we have a valuation v which satisfies the property that v(south)+v(north) is a good estimate of the number of tricks available to the north and south pair. What more can we say about this?

Well, if Tr(south) is the number of tricks south expects to take, total, on average, just looking at his hand, then we'd expect:

Tr(south)=Average(north,v(south)+v(north)) =
  v(south) + Average(v(north))

where the average is taken over all north hands which are disjoint from the south hand.

But Tr is just the values from my prior article. Let's say we restrict ourselves to valuations v which have the following form:

v(hand)=vp(pattern(hand)) +
  vh(spades(hand)) + vh(hearts(hand)) +
  vh(diamonds(hand)) + vh(clubs(hand))

So we assign a value vp() for each pattern from "13-0-0-0" to "4-3-3-3."

We also assign values vh() to each suit holding, from void to AKQJT to 13-card suits.

If v is of this form - what I called in my previous article a *shape-adjusted holding evaluators* - then we get that the Tr() function also satisfies the rule:

```
Tr(hand)=Trp(pattern(hand)) +
    Trh(spades(hand)) + Trh(hearts(hand)) +
    Trh(diamonds(hand)) + Trh(clubs(hand))
```

where

```
Trp(pattern(hand))= vp(pattern(hand))+Average(vp(pattern(pard)))

Trh(holding) = vh(holding) + Average(vh(pardHolding))
```

Determining v from Tr

The data from the original article is just these Trp and Trh values - that is, we have already values for these functions. Can we then work backwards and construct vp() and vh()?

Yes, we can.

Think of the values of vp() and Trp() as vectors:

```
VP = [ vp(4-3-3-3), vp(4-4-3-2), vp(4-4-4-1), ... , vp(13-0-0-0)]
TRP = [ Trp(4-3-3-3), Trp(4-4-3-2), Trp(4-4-4-1), ... , Trp(13-0-0-0)]
```

Then the Avg(vp(partner)) can be expressed as:

```
VP * A
```

where A is a relatively simple-to-determine 39x39 probability array (since there are 39 hand patterns.)

Then TRP = VP + VP*A = VP * (I+A) and determining VP amounts to inverting (I+A). If we invert this matrix, we get:

```
VP = TRP * (I+A)^(-1).
```

For holdings, we can do a similar computation. If we have 8192 holdings, we have to invert an 8192x8192 matrix, which we don't want to do. But if, instead, we treat cards smaller than 9 as "spots," then we get vectors of size 512, and the Perl package I used to invert matrices was able to invert the 512x512 matrix in a few hours.

The resulting values for v(), for offense, defense, notrump and suit, can be found in: patterns.txt (patterns.pdf) and holdings.txt (holdings.pdf).

Notes on the values

I had to "fudge" some of the data in the TRP vectors, since I didn't have much data for suits of length 9-13.

You'll notice that the suit offense value given for the 13-0-0-0 shape is only about 7.68, although you are certain of 13 tricks. Why is that? Because partner always has a void in your suit, so his (shape) value is going to be 5.116, minimum. So the 7.68 automatically takes into account partner's known void, avoiding "double-counting."

Similarly for any long suits, this evaluator takes into account that partner might be holding shortness in the suit.

Someone emailed me a pair of hands and asked for an evaluation of the notrump prospects of the two hands:

KJT Axx KQx AJTx

A9xx Kxx Axx K9x

My evaluator comes up with slightly more than 13(!) But, since the data driving this evaluator is double-dummy data, that's not too surprising, because double-dummy, I can finesse both black queens, so I can take four spades, four clubs, three diamonds and two hearts. That's the risk of using double-dummy data. Even without the nice nines and tens, my evaluator says these two hands are worth 12.5 tricks. That seems a bit rich - how often can I make 12 tricks, even double dummy?

Still, Binky Points are surprisingly accurate at guessing the appropriate level for a deal.
Suit Values: Losing Tricks, Sort Of

In suit contracts, the maximum and minimum values for any suit length give us a sense of the total value of cards:

Worst	Best	Difference
x	A	1.36 Tricks
xx	AK	2.42 Tricks
xxx	AKQ	3.08 Tricks
xxxx	AKQJ	3.31 Tricks
xxxxx	AKQJT	3.41 Tricks
xxxxxx	AKQJT9	3.26 Tricks

This table is interesting, but I'm not sure what to make of it. It looks somewhat like losers, but the difference in tricks is a bit more than you'd expect.

Another table for suit contracts:

Worst	Ace	Difference
x	A	1.36 Tricks
xx	Ax	1.52 Tricks
xxx	Axx	1.60 Tricks
xxxx	Axxx	1.61 Tricks
xxxxx	Axxxx	1.50 Tricks
xxxxxx	Axxxxx	1.53 Tricks

Ace with other honors:

Lower	Higher	Difference
Qx	AQ	1.64 Tricks
Qxx	AQx	1.69 Tricks
Qxxx	AQxx	1.70 Tricks
Qxxxx	AQxxx	1.68 Tricks
Qxxxxx	AQxxxx	1.59 Tricks

Lower	Higher	Difference
Kx	AK	1.49 Tricks
Kxx	AKx	1.56 Tricks
Kxxx	AKxx	1.58 Tricks
Kxxxx	AKxxx	1.59 Tricks
Kxxxxx	AKxxxx	1.59 Tricks

Lower	Higher	Difference
KQx	AKQ	1.41 Tricks
KQxx	AKQx	1.49 Tricks
KQxxx	AKQxx	1.51 Tricks
KQxxxx	AKQxxx	1.52 Tricks

Lower	Higher	Difference
KJx	AKJ	1.50 Tricks
KJxx	AKJx	1.51 Tricks
KJxxx	AKJxx	1.57 Tricks
KJxxxx	AKJxxx	1.57 Tricks

Lower	Higher	Difference
QJx	AQJ	1.65 Tricks
QJxx	AQJx	1.61 Tricks
QJxxx	AQJxx	1.70 Tricks
QJxxxx	AQJxxx	1.58 Tricks

Lower	Higher	Difference
QTx	AQT	1.56 Tricks
QTxx	AQTx	1.54 Tricks
QTxxx	AQTxx	1.56 Tricks
QTxxxx	AQTxxx	1.55 Tricks

So, we see the value of an ace is highly variable, anywhere from 1.36 (in the case of a stiff) to 1.7 tricks.

Also realize that the 'x' spots in the above calculations are always less than nine.

In general, for a holding, we could define the 'top-down' value of the honors by first finding the value of the highest honor relative to all spots, and going down from there. For example:

Lower	Higher	Difference
xxxx	Axxx	1.61 Tricks
Axxx	AQxx	0.66 Tricks
AQxx	AQJx	0.28 Tricks
AQJx	AQJ9	0.01 Tricks

So in the holding AQJ9, the ace is worth 1.61 tricks, the queen is worth 0.66 tricks, the jack is worth 0.28 tricks, and the nine is worth practically nothing. A bottom-up approach might give you something like:

Lower	Higher	Difference
xxxx	9xxx	0.05 Tricks
9xxx	J9xx	0.31 Tricks
J9xx	QJ9x	0.55 Tricks
QJ9x	AQJ9	1.65 Tricks

So a bottom-up valuation gives the ace as worth 1.65 tricks, the queen 0.55 tricks, the jack as 0.31 tricks and the nine as 0.05 tricks. We could take all orders and then average. For the example of AQJ9, there are possible sequences of the honors. We can compute the average ace value:

Lower	Higher	Difference
xxxx -> Axxx	(6 times)	1.61 Tricks
9xxx -> A9xx	(2 times)	1.64 Tricks
Jxxx -> AJxx	(2 times)	1.67 Tricks
Qxxx -> AQxx	(2 times)	1.70 Tricks
J9xx -> AJ9x	(2 times)	1.71 Tricks
Q9xx -> AQ9x	(2 times)	1.67 Tricks
QJxx -> AQJx	(2 times)	1.67 Tricks
QJ9x -> AQJ9	(6 times)	1.65 Tricks

So the weighted average value of the ace in AQJ9 is 1.65 tricks. Similarly, we can evaluate the weighted average for each card, and we get the value of the queen is 0.55, the value of the jack is 0.29, and the value of the nine is 0.07.

Thomas Andrews (bridge@thomasoandrews.com)