Cooperative Fuzzy Agents Combining Human Entered Data with a Database, Fuzzy Logic and Q-learning

Kalle Prorok
Department of Computing Science
Umeå university
SE-901 87 Umeå
Sweden
Email: kalle.prorok@umu.se

***Abstract*—We go through a way of using Fuzzy-rules combined with human interference and Q-learning to get a suggested structure for managing real world problems, specially those related to communication between people and agreements or auctions in an not fully informed stochastic situation. We are using the bidding system for the cardgame Bridge as an simplified but useful model world, smart people have struggled since 1925 to create a good bidding system without full success. A survey of alternative methods is also given and Fuzzy rules looks like the most promising representation due to their human readable and understandable form making them more thrustworthly and adjustable. The result is on the level of good Bridge-players and suggestions of how to probably achieve super-human performances are given.**

I. Introduction

Many games including Backgammon[], Chess[], Go[] and Poker[] have been more or less successfully solved by computers/AI as well as the card-playing part in Bridge. After many attempts to model, learn and encode the preceding bidding before the play in the card-game of Bridge it now seems we have found a start for a successful approach. The bidding is an auction in which someone will get the highest contract and can decide what suit (if any) is supposed to be trumps. During this bidding the players also can give special meanings(semantics/encoding) to the bids to be able to transfer information about the cards they are given (their *hand*) to reach the pair's best *contract* (level; number of tricks to take, and trump-suit). The opponent-pair also tries to find their best, maybe higher, contract as well as trying to destroy the bidding for the first pair; this makes this part of the game a communication channel with restricted band-width as well as destructive adversial elements in a stochastic environment. There are also a lot of hidden information which makes it interesting and more related to real life situations.

A comparison/analogy can be made if i.e one of the player is a medical doctor and the other a psychologist. They study/measure their patient (hands) in their own way and end up with parameters to describe the situation (blood pressure, overweight and problems with temper, divorce, family problems respectively). Given those parameter values they can discuss with each other to find the best treatment 'contract'(pills, prison, hospitalizing etc), maybe based on statistics of the results from previous treatments in similar situations/patients. Here we also have limited amount of time/money spending for finding parameter values as well as limited time for diagnosing and discussing and finding a common language between the experts and their areas so the modelled Bridge-framework example can be useful in other applications as well, more examples might be similar; the structures can be useful for contractors and agreements, politicians, community-personel with different competences finding the best solutions; i.e should society go for more police officers or take better care of the youths to avoid them to get criminal(s)?

II. Methods

We will try to classify important features in the hands; finding some values who might be important to describe for our partner and for us to make good final decisions. In our first case we restrict ourselves to just one situation which then can be muliplied into many others. We assume an average strength bridge hand with a spade-suit combined with a weaker hand without support in spades. This is one of the scenarios who are difficult for humans to handle; is it worth to continue the bidding for a remote chance to get a score of +600 instead of +120 but with a risk of getting -200 sometimes instead? This depends on several things; a) most players will have difficulties to estimate the percentage levels and calculate the expected value b) we need to have efficient bidding agreements to be able to stop low if the hands doesn't fit each other well and bridge-people might not have the ability or will to remember complicated agreements about seldomly used bids so computers have good chances to be super-human in those situations. The collected meaning of each bid, *the bidding system*, is quite free to be agreed upon but has to be publicly available in advance for the opponents to handle.

The general plan in this work is to reduce search space by using rules, classifying hands into types, use human knowledge where possible and add AI/learning were it is needed; the computer doesn't have to reinvent the wheel; by this the results will probably be meaningful, understandable, useful and not too difficult to achieve.

A. Previous attempts

Even if this looks a bit trivial the author have spent several years on trying different approaches. As being a professional programmer for 20+ years as well as a Bridge player on the national level this has become some kind of lifework; "being your own expert" the solutions are supposed to be reasonably good and realistic. Here is a brief summary of some of the latest attempts. They are not ranked of how bad they performed in tue A*-spirit; we will do that when we have really good methods.

1) Rule-based: By using Lisp or matching Prolog-clauses or other relevant descriptive languages there are still to many (10,000+) rules to be practical to enter. it might work in amini-situations but scales poorly into real bridge.

2) Q-learning: 3 years and a Lic.Thesis was done trying to learn bidding sequences by using 6-dimensional arrays in a Reinforcement learning setting but one of the problem was that if some of the earlier rules were changed the things already learnt became obsolete due to destroying the Markov property; the environment changes while learning and there is no longer any promise of convergence. Maybe a multi-step Q-learning will work? Today's computers are also more powerful compared to 15 years ago when those experiments were made.

3) Heavily encoded: Bridge-playing software often have some kind of smart encoded rules. The representation is often very complex/cryptic and almost only the author can make changes and the number of systems and conventions are limited and not easily extendable. They usually fail in some uncommon, untested situation and make totally crazy bids. Expert bridgeplayers can also make strange (psychic) bids fooling those softwares.

4) Genetic algorithms: Has been tested and might work. It is important to find some nice encoding/representation. Tried some but the very large search-space demands a lot of (hundreds of years?) patience, I didn't/don't have that.

5) Random systems: An attempt was made with the Fuzzy-rules in this paper but the system to test was randomly generated, assuming computer raw power rules over smartness; 10,000 combinatorical versions of rebids and then 10,000 responses in combination which makes 100 miljon bidding systems, at least some of them should be ok? Evaluated them on 10,000 examples each in this very limited 1 ♠ - 1 NT situation. These trillions took 4 hours on an i4790 3.6 GHz i7 with 16 GB RAM and the four cores running 100%. All randomly generated systems was garbage so even the best of them was not even good in practise; the problem was that the systems were not careful enough with bidding space - by bidding to high a negative income was guaranteed. There was a try to keep bidding levels low by randomizing lower bids more but still to bad, it might work with many more attempts but it has to be better, smarter, ways of doing it.

6) Artificial Neural Nets[ANN]: Some tries were made but ANNs doesn't feel optimal for problems based on symbolic data. We used See5 [] to translate example-runs through the net and made rules based on the tried examples to study the black-box approach, to try to understand what the ANN have learnt and outputs with some typical inputs. The rules were very messy even if we tried to prune the results so they were not useful for humans to learn from.

B. Playing database

As a help there is an experience double-dummy play database with almost a million deals computer played by GIB[] in all contracts so if you have a bidding system, this database, even if not perfect, can be used to learn/improve from or for evaluating the performance. The very same hand-pair can get different number of tricks depending on how the opponents cards were dealt, i.e if the Ace is positioned before the King or after the King and able to kill it so the results are stochastic and not ground truth. It is also played by studying all hands (cheating) and this doesn't happen in bridge were the hands are hidden to each player but given that the opponents also gets some extra information this might balance itself - the more modern GIB can play hands without cheating but there are no database as far as I know but it can be produced.

Without this pre-calculated database the deals have to be analysed on-line during learning, taking much more CPU-time but possible. The entries in the database looks like this (slightly modified and skipped last hand):

$KQJ97.4.A72.QJ72$
$A4.KQJT86.KT3.K6$
$65.A753.QJ95.A83$
$9797A9A96666BABAAAAA$

and were converted to shape (distribution of cards in the hands with ♠ (5) first, then ♡ (1), ♢ (3) and ♣ (4) and honour strength (Ace = 4, King 3, Queen 2 and Jack one point), typical contracts and tricks:

5134(13) 2443(11) 1NT 2♠ 9 6 10 10 7

The 9797.. sequence says how many (in hexadecimal) tricks that can be taken depending on the contract and who is starting. In our approach the first hand always starts the bidding (with 1♠) and the third hand makes the response (1 NT) and then the first hand bids again and if that bid is not pass the responder will be given a chance to bid another bid. The bid is formally a promise to try to take 6 + *the level* tricks out of the thirteen possible. Bids always have to be increasing

1♣,1♢,1♡,1♠,1NT, 2♣, 2♢, 2♡, 2♠,2NT, 3♣,..7NT

so there is a risk of bidding to high, failing on your promise and give the opponents score instead. In our calculations we assume the scoring according to 'the vulnerable zone'; more bonuspoints are given if bidding 3 NT, 4♡ or 4♠ but also double the penalty if failing. The hand between $(A4.KQ..)$, one of the opponents, is supposed to keep silent (bid Pass) during the bidding but that is sometimes not the case in real Bridge; in this situation the bid would have been (2♡), then we have opponents who bids; disturbed adversial bidding and will be handled in a later paper.

The fuzzy logic translates this into the best Fuzzy matching rule (first if equal match) for both hands and the examples of contracts and number of tricks are used for evaluation and improvement of the bidding system:

Example 1. Rule 4 and Rule 6.

Another example from the database, making 3 NT: $AT964.A4.K4.K854$
$5.KQJ2.J652.QT93$ 9999888877777777AAAA
5224(14) 1444(9) 3N 2S 8 7 7 10 9

Example 2. Rule 9 and Rule 0

We hope these rules catches the essence of the values so it will be able to achieve god results while still generalisable into similar, unseen, situations.

C. Fuzzy rules

Why are these so good? Because the representation resembles the traditional human way of representing the system including the Fuzziness; i.e if you have a bidding system where you either open with your best 5+card suit or 1 NT (No trump) if you have at least 2 cards in each suit but no 5-card suit, a *balanced* hand. With the rare 1-4-4-4 distribution you have no opening bid and have to improvise into what is the best compromise; with 4 good heart-cards like AKJ7 you might open 1 ♡ even if you only have 4. You can also change the bidding system to allow/promise only four card suits but this is a greater sacrifice for this rare situation. As the bidding situations becomes more complex more membership functions may be added; i.e number of Aces, stoppers in a suit etc to be able to distinguish/separate otherwise similar situations. This information can be achieved from the database.

We were using the multiplicative Fuzzy And-rule,

$$Fuzzyand(\alpha, \beta, \gamma, \delta, \epsilon) = \alpha \times \beta \times \gamma \times \delta \times \epsilon,$$

because the traditional and-min is too coarse, not separating well by simply using the lowest value independent of the other factors.

1) Membership functions for shape/suit lengts: We have only used 2 membership functions as seen in Figure 1; one(-) for sligthly shorter than promised and one(+) for slightly longer length in respective suit.

2) Membership function for strength: Also here only two membership functions; minimum strength(-) or extra values (+) simliar to the shape-functions but with the values $-2, 1, 4$.

3) Prototype hands: For each bidding situation a typical minimum hand is given. The rules then descibes improvements compared to the minimum hand; more cards in some suit, more high-cards. This make it possible to use the same shape- and strength-membership functions even if the orginal shape/strength differs. In our case we have used 5333(12), not a Bridge hand (14 cards) but we didn't want to make any suit more significant than any other i.e. by using 5323 premiering diamonds. The responder's prototype hand to compare with was 1444(9).

D. Filtering

We limited ourselves to 1 ♠ - 1 NT (No trump), one of thousands of possible situations. The honour strength was measured by assigning points to high cards;. The spade-hand was limited to 11-16 points and the 1 NT-responder to 8-12. About 12,000 hand-pairs out of 700,000 examples were fullfilling these requirements and were filtered out, saving about 2,000 examples for evaluation/test.

E. Selecting appropriate rebids

These were manually set to reasonable values, not to high making the further bidding worthless.

F. Learning

These definitions are usually made by humans anyhow but further bidding assumes logic thinking and there are veruy many situations. We let the computer use Q-learning for the last bid, making the environment Markov by keeping the bidding definitions static. [Will include illustrative general learning overview graphic Figure here]

G. Estimating results

The results are given from the database. Even if 3 NT might be the best contract the bid 4 ♠ can be playable and managed ok by chance and even get higher results (+620 instead of +600) but is maybe more risky.

III. Results

A. Fuzzy rules

Small rule-set (11) and larger set (22). Extra length in suits and/or extra strength via honors. The larger set handles extra length in two suits. Same set of rules for opener and responder. The hands are compared to a prototype hand.

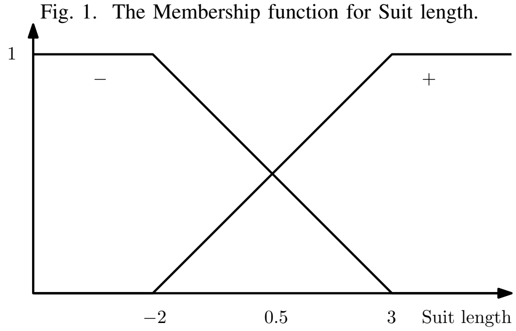
TABLE I
THE RULES, MEMBERSHIPS AND BY HUMAN RECOMMENDED BID.

Rule		Memberships					Rebid
Number	Mnemonic	♠	♡	♢	♣	Points	
0	-	-	-	-	-	-	Pass
1	♠	+	-	-	-	-	2♣
2	♡	-	+	-	-	-	2♢
3	♢	-	-	+	-	-	Pass
4	♣	-	-	-	+	-	Pass
5	+	-	-	-	-	+	2 NT
6	♠+	+	-	-	-	+	2♡
7	♡+	-	+	-	-	+	2♡
8	♢+	-	-	+	-	+	2♠
9	♣+	-	-	-	+	+	2♠
10	♠♡	+	+	-	-	+	Pass

B. Membership functions

Hand-tuned for god spread as can be seen in Table II, all rules are in use and no rule dominates. Regarding combination of rules all rule-pairs doesn't happen so often (like Rule 7, ♡+ for the opener combined with Rule 5 + for the responder, often making the contract 4♡ appears only twice) and some are very common (like the most common; Rule 1 for both; extra length in ♠, 305 times giving better statistics of possible contracts

Fig. 1. The Membership function for Suit length.



although low ones). Most combinations appear 10 to 100 times and the author have studied the list of the 3 most reasonable contracts in each pair. (11x44 table, to large to show).

TABLE II
THE RULES, HOW OFTEN THEY ARE APPLIED TO THE EXAMPLES.

Rule	Opener	Reponder
0	1348	202
1	1719	1852
2	988	883
3	1069	1286
4	1147	1240
5	684	109
6	1011	1333
7	604	642
8	548	720
9	547	641
10	335	1092

C. Estimating results

The bids were selected according to the tables and the final contract was then evaluated into a score based on the contents in the bidding database. It is possible to evaluate the contracts giving the highest scores for all examples and this will be the roof of much the score sum can be and the achieved performace percentage calculated in each case. In theory this sounds as a good idea but many of the high scores in the database is a result of luck, in reality luck seldom appears, rather, bad luck is more common so an evaluation of unused examples will be a better way. No attempt has yet been made because we plan to improve the sequencing of bids to further improve performance before doing a serious evaluation.

D. Implementation

The software was implemented in high-performance Parallel C++ with Visual Studio 2017 RC under Windows 10. It was important for the Randomized Bidding systems mentioned in [] but not necessary for the Fuzzy attempts; they were running within a few minutes with a single core solution although a Python based implementation might need an hour or so unless optimized.

E. Rebids

The openers rebids were chosen manually. Many sets kan be specified and then evaluated to find out which gets the best scores on average. Table III shows one set as an example and an extra attempt with natural bids (the bids are showing the suit bid; i.e 2♣ promise extra length in ♣s) was done. Maybe ten different combinations can be meaningful to compare but spending time for inventing better handling of sequences and disturbed bidding is prioritized as it is more realistic and useful in real Bridge.

TABLE III
THE REBIDS BASED ON GIVEN ARTIFICIAL MEANINGS.

−	♠	♡	♢	♣	+	♠+	♡+	♢+	♣+	♠♡
P	2♣	2♢	P	P	2NT	2♡	2♡	2♠	2♠	P

F. Second response

This is learnt via a simple Q-learning algorithm which tries different bids and selects the one which gives most points on average. The results are presented in Table IV.

TABLE IV
THE LEARNT 2ND REPLY-BIDS AND THE SCORE.

Rebid:	Pass	2♣	2♢	2♡	2♠	2NT
Rule	2nd reply to openers rebid					
0	P 24.2	2♠ 8.49	2♡ 18.3	2♠ 32.5	2N 6.69	P 13.1
1	P -53.9	2♠ 250	2♡ 60.6	2♠ 382	P 176	P 62.4
2	P -20.2	2♡ 7.19	2♡ 67.5	P 172	P 25.7	P 37.9
3	P -112	2♠ 82.6	P 28.8	2♠ 158	P 79.6	P 61.2
4	P -59	2♠ 96.9	2♡ -0.0125	2♠ 208	P 58.1	P 34
5	P 36.8	2♠ 32	2♡ 15.4	2♠ 14	P 15.1	P 12.6
6	P 499	2♠ 292	2♡ 109	2♠ 272	P 202	P 94.7
7	P 264	2♡ 138	2♡ 39.5	P 111	P 96.4	3♡ 42.2
8	P 205	2♢ 83.3	P 69.8	P 39.5	P 63.6	P 45.7
9	P 201	P 75.9	3N 80	2♠ 138	P 85.1	P 44.4
10	P -72.5	3♠ 26.9	2♡ 62.7	P 112	P 155	3♡ 32.3

Sum of performance scores : 5641.05 .

This bidding system is called conventional or artificial because the bids doesn't describe what you have in the bidden suit; i.e 2♣ shows extras in ♠s. As a comparison we made a trial with a natural system, bids showing extras in the bid suit, as specified in table V. Many players, particularly beginners and those without ambitions dont have any conventions and only bids natural hoping for the best - no complicated rules to remember wrongly leading to disasters. As can be seen in table VI the most common 2nd rebid is Pass. In the conventional system the Q-learning algorithm understood that the 2♣-bid showed spades and seldom passed but bid 2♠ mostly, having the chance to stop in the lower 2♡-contract with plenty of hearts and bad/few spades (Rule 2 and rule 7 in Table IV).

TABLE V
THE REBIDS BASED ON A NATURAL SYSTEM.

−	♠	♡	♢	♣	+	♠+	♡+	♢+	♣+	♠♡
P	P	P	P	P	P	2♠	2♡	2♢	2♣	2NT

Sum of performance scores for natural system: 4982.7 . The result is slightly worse than the proposed artificial system but it remains to be seen how it performs on unseen test-hands.

TABLE VI

THE LEARNT 2ND REPLY-BIDS AND THE SCORE, WITH A NATURAL SYSTEM.

Rebid:	Pass	2$\clubsuit$	2$\diamondsuit$	2$\heartsuit$	2$\spadesuit$	2NT
Rule	2nd reply to openers rebid					
0	P 39.5	P 4.89	P 17.2	P 13.7	P 29	P 2
1	P -24	P 71.5	P 78.2	P 124	P 269	3$\spadesuit$ 18.2
2	P -1.6	P 58.9	2$\heartsuit$ 59.4	P 50	P 109	3$\heartsuit$ 13.3
3	P -72	P 65.8	P 32.7	P 77	P 108	3$\spadesuit$ -1
4	P -63.1	P 33.9	2$\spadesuit$ 39.4	P 56.1	P 150	3$\spadesuit$ -1
5	P 88.5	P 8.49	P 5.6	P 2.8	P 12.1	P 4.7
6	P 878	P 86.9	P 70.9	P 107	P 161	3$\spadesuit$ 57.8
7	P 404	P 30.4	P 41.6	P 25	P 91.8	3$\heartsuit$ 13.7
8	P 408	P 48.3	P 16.4	P 57.5	P 128	3$\spadesuit$ 12.6
9	P 387	P 15.9	P 58.4	P 43.6	P 96.6	4$\heartsuit$ 64.2
10	P -61.2	P 28.8	P 39	P 52.7	P 127	3$\heartsuit$ 12.5

IV. DISCUSSION

A. Sequences

In our case we have studied quite short bidding sequences like 1$\spadesuit$ - 1 NT, Pass or 1$\spadesuit$ - 1 NT, 2 in a suit, Pass or 1$\spadesuit$ - 1 NT, 2 in a suit and a 2nd response obligatory followed by an virtual Pass (to finish the bidding); i.e. 5 bids in a row. By allowing longer sequences there will be more possibilities for distinguishing hand-types without bidding to high; i.e. the sequence 1$\spadesuit$ - 1 NT, 2$\clubsuit$ - 2$\diamondsuit$, 2$\heartsuit$-2$\spadesuit$, 2 NT - Pass (8 bids). For the 1 NT opening those aspects are even more important so to be able to make it really god you have to handle this.

B. Full system

Our single situation can be redone in other situations; maybe there are about one thousand cases which should be reasonably ok to do semi-automatically. But we prefer to improve the performance (adding longer sequences and opponent bids) first to hopefully be super-human and then add more situations with the new approach.

C. Opponents bids

Another important aspect is to deal with disturbances; on many of our examples the opponents would have made a bid (if allowed) and the bids we have learnt wouldn't have been available; we can't bid 2$\diamondsuit$ if opponents already bid the higher 2$\heartsuit$. The opponents might have artificial disturbances and describe what the bids might contain for type of hands; i.e 5 spades and either 5 clubs or 5 diamonds (Micaels bid) and these are also to be handled by being able to enter information on-the-fly

D. Inventing bids

Computers have the possibility to simulate better during the bidding than humans have. Suppose a situation arises when there are no agreements - both players will know that and both know their own hand and what information they both have transfered through the bidding up till now. They can then both simulate separately arriving into the same result, a kind of thinking what the other is thinking, to agree withouth explicit communication.

V. CONCLUSION

We have managed to make a small bidding system on a subpart of the bidding in the cardgame of Bridge, bidding on a tournament player's level. By adding sequences we probably can get super-human performance.

VI. BLANKS

Temporary Empty space

ACKNOWLEDGMENT

Thank you Patrik Eklund and Frank Drewes for important information and inspiration before writing this paper. Thank you also Per-Åke Wedin for helping me achieving my Tech.Lic-thesis 2003.

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

(References will be added, if accepted, in the photo-ready version)