

# Large Scale Q-tables

## A Study using Bidding in the Game of Bridge

Kalle Prorok  
Department of Computing Science  
Umeå University  
SE-901 87 UMEÅ  
Sweden

October 17, 2001

### **Abstract**

This article studies the possibilities for Q-learning to learn the bidding in the card-game Bridge. The learnt Bridge bidding systems are quite useful but below the level of an average club-player. Some studies regarding the relation between the learning rate, the exploration rate and initial Q-values are also done.

## 1 Preface

As soon as computers entered people tried to make use of them when playing games ([1], [2]). In the Bridge area there have been some tries but the game is more complex than many other games. A lot of attempts to play Bridge has been done. There are also plenty of training “tools” for improving human play and bidding by doing statistical analyses and for practicing but not until recently (2000) a computer Bridge player [3] has become competitive with humans. Its card-play is almost always technically correct but there is still plenty of room for improvements in the bidding and tactical areas.

The term reinforcement learning(RL) seems to be coined by Minsky in the 1960’s and also turned up independently in control theory ([4] and [5] as cited in [6]) but only recently recognized by each other [7]. One of the advantages with applying RL to a game (like Bridge) compared to many other areas are the quick evaluation of the results compared to robotics (with slow hardware) and repeatability compared to many psychological or biological experiments.

## 2 Introduction and Background

The Bridge application has also some other interesting features like learning to cooperate with a learning partner and developing some kind of common language.

My hope is that these experiments with Bridge will be applicable on other domains like network management (information transfer), compression algorithms (finding descriptions of i.e. a face automatically), electronic commerce (how to perform an auction) and robotics (learning to interact). Another area might be language learning; how did we invent a language “in the beginning”.

Machine learning methods are advantageous when preprogramming is not possible as in the following situations:

- unknown environments;
- varying environments;
- too complex or much to program.

### 2.1 Why Bridge?

Blocks world [8] has been used as a small-scale model system for decades in the Artificial Intelligence community. Today’s problems are of a different kind; cooperation, stochastic and/or non-linear systems, emergent properties etc. My suggestion is to use the international card-game Bridge as a new model world with some interesting properties from the real world but still on a reasonably small scale. Some of the things that can be studied using the Bridge platform are:

- Collaboration/Cooperation within pairs and teams;

- Competition between pairs, teams and countries;
- Communication - information transfer with limited bandwidth;
- Insecure domain - not all information is visible for all parts;
- Stochastic domains - there is a random factor (how the cards were dealt);
- The role of concentration; mental training;
- Planning (of bidding and playing);
- Emergent properties; single suit versus multi-suit plays, end-plays;
- Democratic(descriptive) or Captain (enquiry) based methods;
- Tactics, Strategy, Risk taking, “the only chance” or safety play “guarding”;
- Constructive versus destructive ways of doing things;
- Psychological aspects - to be unpredictable, when to fool partner or opponent;
- Philosophical aspects - try to find out what partner or opponents think;
- Memory; a complex bidding systems versus an easy to remember, understand and use bidding system;
- Getting information - what and when;
- Reducing amount of information to opponents who may take advantage;
- Learning from mistakes and successes.

The game is described in 2.1.1 and can be learnt from [9] with the standard bidding in Sweden presented in [10]. Bridge is fairly international with similar rules all over the world and there are at least one million players in the world. Now it also is an Olympic game. In my opinion bridge is more complex compared to i.e. Chess which is an open game (no hidden information or stochasticity) and with no partnership to bother about, now performed well by computers using more or less brute-force search methods (IBM Deep Thought II) and databases for openings and end-plays. There are bridge programs performing well and it is also a matter of time before the computer beats most human bridge players. The main reason for this is the technical advantage a computer has (remembering all cards during the play and the bidding system, possibilities of statistical simulations and calculations making it possible to play well and invent conventions “on-the-fly” etc) compensating for the lack of intuition and other psychological aspects by making fewer mistakes. Today’s situation (September 2001) is that the play with the cards is very good but bidding lacks behind. The reason for this is that entering human judgement as rules in a bid-system database is very boring and tedious. To reduce the size, some compact descriptions are used but are prone to errors (Fig. 4).


	North/Deal	
	♠ -	
	♥ T9632	
	♦ T72	
	♣ QT873	
West		East
♠ Q963		♠ AJT842
♥ 85		♥ AK7
♦ AJ64		♦ 5
♣ J86		♣ AQ4
	South	
	♠ K75	
	♥ QJ4	
	♦ K92	
	♣ K953	

Figure 1: A typical deal in schematic form. It is dealt by player North who then starts the bidding with East next in turn. North is void in (have no) spades but has a weak hand and passes. East has a strong hand and normally opens the bidding by giving a bid, maybe 1 ♠. South has an average hand but not enough to enter the bidding with a bid other than Pass.

### 2.1.1 The rules of Bridge

Bridge is a card game played with a deck of 52 cards. The deck is composed of 4 suits (spades ♠, hearts ♥, diamonds ♦, clubs ♣). Each suit contains 13 cards with the Ace as the highest value and then the King, Queen, Jack, Ten, 9, . . . , 2. The first five are often abbreviated to A, K, Q, J, T. The game begins with some shuffling of the deck (not in tournament play) and then the cards are dealt to the four players, often denoted North, South, East and West (N, S, E, W). N and S play together in a team against E and W. A typical deal is presented in Fig. 1. Before the card play begins an auction (bidding) takes place. One of the teams<sup>1</sup> wins a *contract* to make at least a certain number of *tricks*.

**Definition 1** *A card from each player, played clockwise is called a trick. The winner of the trick is the one, having the card with the highest rank in the suit*

---

<sup>1</sup>With the highest bid.

N	E	S	W
P	1♠	P	2♠
P	4♠	P	P
P			

Figure 2: The schema of the bidding with the hands in Fig 1. Three passes ends the auction.

*played by the first hand (the leader). In triumph contracts the highest card in triumph, if played, wins the trick.*

To prevent opponents from taking a lot of tricks in a long suit of theirs, a trump-suit is looked for in the bidding. If no common trump suit is found, a game without trumps can maybe be played; No Trump (NT), which is ranked higher in the bidding than the suits which are ranked ♣ (lowest), ♦, ♥, ♠ (highest).

During these two phases, the bidding and the play, there is cooperation in pairs playing versus each other with four players at each table. The pairs communicate via bidding where the bids are limited to a few (15) “words”, but many sequences “sentences/dialogs” are possible. Opponents are kept informed<sup>2</sup> about the *meaning* (semantics) of the bid or sequence. This can be seen as a mini-language with a few words like 4, 1, Spades, Double, Pass etc. and these words can be combined via simple grammatical rules into short “sentences” like 3 Spades. Some sentences are put in a sequence “conversation” called the bidding. It can look (within a pair) like E:1 Spade - W:2 Spades, E:4 Spades - W:Pass with semantic meanings like “I have a hand with spades as the longest suit and above average number of Aces, Kings and Queens” - “I have support for your spades but not so good hand”, “I have some extra strength” - “I have nothing to add”. The first non-pass bid is denoted *opening*. A full bidding schema is presented in Fig. 2. The normal goal in the bidding is to agree on a suit with at least eight cards in common and at an appropriate level. Sometimes a *common suit* with seven cards will or have to do or a NT contract without a trump-suit.

When the bidding has finished, a pair handles the final contract (4 spades) and the play with the cards can begin by taking tricks. After the play, the pairs are given points according to the *result*. In tournament play the points are compared to the other pairs playing a *duplicate*<sup>3</sup> version of the deal, almost compensating the factor of luck with “good cards”.

<sup>2</sup>If they ask.

<sup>3</sup>Or replayed later by storing the deal in a wallet.

In our work we have studied undisturbed bidding which means the pair is free to bid on their own without disturbing interventions from the opponents. There is no need for preemptive bids and sacrificing or to keep in mind the importance of reducing the amount of information transferred to the opponents. By having these limitations we are able to learn better in shorter time and requiring less memory.

### 2.1.2 Scoring

Modern, tournament, scoring in bridge is done the following way; The suits are classified into two groups, minors ( $\clubsuit$  and  $\diamondsuit$ ) and majors ( $\heartsuit$  and  $\spadesuit$ ). For each trick above 6 taken in a minor contract you earn 20, for each in a major it will be 30. In No Trump (NT) there will be 40 for the first and 30 for the next one(s). You are supposed to take at least the number of trick given by the contract, otherwise the opponents get 50 for each undertrick. If you have bid a contract where the sum of the points is at least 100 (i.e.  $5\clubsuit$ ,  $4\spadesuit$ , 3 NT or higher) you have contracted for a *game* and if you make it you will get 300 extra as a bonus, lower contracts give a part-score bonus of 50. Some examples:  $4\spadesuit$  bid and made gives  $4 * 30 + 300 = 420$  but  $3\spadesuit$  with an overtrick gives only  $4 * 30 + 50 = 170$ ,  $4\spadesuit$  down one gives 50 to the opponents but  $3\spadesuit$  exactly made gives  $3 * 30 + 50 = 140$ . IRB (In real bridge) there is actually a variation added; zones; when vulnerable the game bonus will be 500 and penalties 100 for each undertrick.

### 2.1.3 Available software

There are some computer programs around (reviewed in [11]) able to play Bridge but most only at a low (beginners) level. One candidate is AI-researcher professor Matthew L. Ginsberg's GIB which he thinks will be better than humans by the year of 2003. In August 2000 GIBson was released and it is able to do declarer play on a very high level. This improvement also makes the bidding better because it simulates the play to find the best or most probable contract (See Algorithm 2; Borel simulations).

**Algorithm 2** ([12], Borel simulation) *To select a bid from a candidate set  $B$ , given a database  $Z$  that suggest bids in various situations:*

1. *Construct a set  $D$  of deals consistent with the bidding thus far.*
2. *For each bid  $b \in B$  and each deal  $d \in D$ , use the database  $Z$  to project how the auction will continue if the bid  $b$  is made. (If no bid is suggested by the database, the player in question is assumed to pass.) Compute the double-dummy result of the eventual contract, denoting it  $s(b, d)$ .*
3. *Return that  $b$  for which  $\sum_d s(b, d)$  is maximal.*

There are dozens of other programs for playing, generating deals[13] etc. Ian Frank has made an end-play analyzer/planner FINESSE which seems to work well but too slow for practical use (written in interpreted Prolog) and he has also written an excellent doctoral's thesis "Search and Planning Under Incomplete

		N-S bid	
		1♣ 4♣ 5♣	
NS-Score	+70	-150	-200
		E-W bid	
	P	2♠ 4♠ 5♠	
NS-Score	?	-170	-420 +50

Figure 3: The partnership N-S are able to select a bid first (in clubs) and then E-W decides. On this hand it is assumed N-S are able to take 7 tricks with clubs as trumps and E-W 10 with spades.

Information” [11]. PYTHON, [14] as cited in [15] is an end-play analyser for finding the best play but not on how to reach these positions.

The former World Champion Zia Mahmood has offered one million pounds to the designer of a computer system capable of defeating him but has withdrawn his offer after the last match against GIB which was a narrow win for him.

**Definition 3** *In a dynamic game[16], the players move in a fixed sequence. The players moving later in a game do so knowing the moves others have made before them. Those who move earlier must take this into account in devising their optimal strategy.*

One example of a dynamic game is Bridge. If the game in Fig. 3 has perfect information, the optimal bid will be 5♣ by N-S and P by E-W (solved by doing a game tree) but in a **imperfect information** game N-S can force E-W into a guess (by bidding 4♣ or otherwise) whether they can take 10 tricks (as denoted in the table) or 9 or even 11 tricks.

A game in which players meet in strategic interactions repeatedly are referred to as **repeated** games. **Zero-sum** games are games where one earns the same amount as the other loses (i.e Bridge).

## 2.2 Computer Bidders

Most programs basically use a rule-based approach. The first one was by Carley[17] with four (!) bidding rules with a performance like “The ability level is about that of a person who has played dozen or so hands of bridge and has little interest in the game”. Wasserman (cited from [11]) divides the rules

```

1*#08115 " {5{ [CDHS]: " !!
%0% %.% <0x1083> *2153*#08116 " 5[+ #b] " !DOPI!
%6% %:H[^:] *~8&[^:] *~8&% *2160*#08117 " 5[++ #b] "
!DOPI! %7%
%:H[^:] *~8&[^:] *~8&[^:] *~8&% *2159*#08118 " 5[+++ #b] "
!DOPI! %8% %:H~8&~8&~8&~8&%
*2156*#08119 " 4N:$7 " !!
%0% %.% *1501*#08120 " (5N| [67] [CDHSN]): " !!
%0% %.% <0x1083> *2153*#08121 " X" !DOPI!
%3% %.% *2154*#08122 " P" !DOPI!
%4% %:H[^:] *~8&% *2155*#08123 " X" !DOPI!
%5% %:H[^:] *~8&[^:] *~8&% *2154*#08124 " P"
!DOPI! %6%
%:H[^:] *~8&[^:] *~8&[^:] *~8&% *2155*#00126 "
.*{{{{{{[CDHS] <#j=3><#k=6>:P:4N " !!
%0% %.% *2037*#00127 " : (P|X): " !!
%0% %.% #00128 " @ACE_BLACKWOOD~" !BLACKWOOD!
%1% %.% *1501*#00129 " @ACE_RKCB~" !RKCB!
%1% %.% #00130 " [123] {{{{{{[CDHS] <#j=3><#k=6>:P:5N$12 " !GSF!
%0% %.% *2229*#00131 " :,: " !!
%0% %.% #00132 " 7#a " !!
%5% %:H[^:] *~9>#a% *2034*#00133 " 7#a " !!
%5% %:H[^:] *~6=#a% *2034*#00134 " 6#a " !!

```

Figure 4: A small part of the GIB-database, showing how the Ace-asking bid Blackwood(4 NT) should be used.

into collection of classes that handle different types of bid, such as opening bids, responding bids or conventional bids. The classes are organized by procedures into sequences. “Slightly more skillful than the average duplicate Bridge player at competitive bidding”. MacLeod [18](cited from [11]) tries to build a picture of cards a player holds but cannot represent disjunction, when a bid have more than one possible interpretation. Staniers [19](cited from [11]) introduces look-ahead search as an element of Bayesian planning and Lindelöf’s COBRA [20] uses quantitative adjustments. Lindelöf claims that COBRA bidding is of world expert standard. Ginsberg has invented a relatively short but cumbersome(Fig. 4) way of representing bidding systems.

GIB uses Terrorist’s Moscito (Major-Oriented Strong Club with Intrepid two Openers) because it is difficult to find a good defense. It is complicated to remember for a human person but easy for a computer because the conditions are well-defined. Actually Moscito seems to be (temporary?) abandoned now (September 2001) due to redesign of the database language.

## 2.3 Representation levels of a hand

The view a human sees is seen in Fig.. 5 but this is of course unnecessarily advanced to use as a computer representation. A shorter representation with



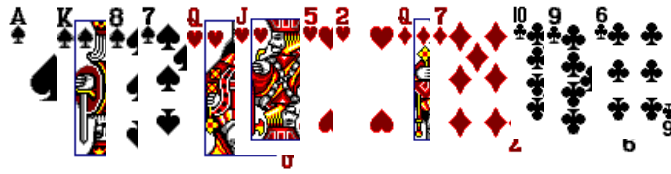


Figure 5: A typical hand

almost the same meaning (losing the graphic “personality” of the cards) is:

♠AK87  
♥QJ52  
♦Q7  
♣T96

(standard way of presenting in books and newspapers) or, simply, AK87, QJ52, Q7, T96 with the order of suits understood. The total number of possible hands are  $\binom{52}{13} \approx 6 * 10^{11}$ . There are 8192 different suit holdings (AK92, AK82, etc), and only 39 hand patterns (5431, 5422, etc disregarding suit-order) [13]. There is a scale invented by Milton Works (table 1) to estimate the strength in a hand, the high card point (h.c.p.)

Card	Strength Value
Ace	4
King	3
Queen	2
Jack	1

Table. The high card points.

and the hand can be represented as a 4432 (4 spades, 4 hearts, 3 diamonds and 2 clubs) with the strength 12 h.c.p.. A hand with AKQJ, AKQ, AKQ, AKQ contains 37 points which is the maximum. The number of possible hands in this representation is  $560 * 38 = 21180$ . The total number of h.c.p. in a deck is 40 and the average strength is exactly 10. Of course this scale is just to make an rough estimate of a hand and every bridgeplayer adjust (reestimate) the value of the hand, maybe dynamically during the bidding due to fit etc. One add-on invented by Charles Goren is the distributional strength

Suit	Distributional value
no card (renons)	3
one (singleton)	2
two (doubleton)	1

Table . Distributional points.

where the total value of our hand is  $12 + 1$  (doubleton ♦) = 13 points. These distributional points are normally not added during a search for a no trump

Table 1: The average number of tricks with a given strength.

(NT) contract. Several other approaches are available as Ely Culbertsons long-suit points and loosing trick count (LTC). Another approach is “The Law of the total number of tricks” [21] which says that the total available tricks by both sides is the same as the number of trumps in common among the both pairs of players. This is not always true but can be of some use during competitive bidding. Thomas Andrews [13] have made some analysis showing that the value of the Ace is underestimated in the 4321-scale.

These strengths are used to define a bidding system in terms of distribution and strength intervals. In our approach we reduce it further by just keeping the distribution of the cards, the shape, and the range of strength:

$$4432, \text{medium}$$

where medium might show 12-14 (Goren)points.

Robin Hillyard [22] has calculated the expected number of tricks at no trump, presented in table 1. My guess is that this table slightly overestimate the number of tricks achieved in real bridge because the analysis is based on playing optimally with open cards which is never the case in real life.

HCP	20	21	22	23	24	25	26	27	28	29	30
Tricks	6.1	6.7	7.2	7.6	8.2	8.7	9.1	9.7	10.1	10.6	11.1

### 2.3.1 Allowed systems

In most real tournaments the possible systems are limited by assigning penalty points (dots) to unnatural opening bids (Svenska Bridgeförbundet(SBF) and World Bridge Federation (WBF)). This is so because it is difficult for a human opponent to find a reasonable defence within a few minutes to a highly artificial bidding system.

**Reinforcement signal** What value of the reward  $r$  should the evaluator give the learners?

a) Calculate the difference between the actual score to the score achievable on this combination of hands

$$r = \text{score} - \text{achievable} \quad (1)$$

It works bad due to high exploration. Assuming a start value of Q with zero “optimistic start”, almost all tried bids will start with negative scores (strange contracts) making them worse than hitherto untested alternatives. This will lead to a huge, time consuming exploration phase of all alternatives.

b) Difference as above but added 200 points.

$$r = \text{score} - \text{achievable} + 200 \quad (2)$$

A contract not worse than -200 is denoted better than the untested cases. The Pass always system evolved guaranteeing a score of 200. Maybe a long run will improve this policy?

c) Absolute score. Simply reinforce the learner by the actual score.

$$r = score * 10 \quad (3)$$

It worked best, at least on our short (one day or two) runs. The score was scaled with 10 to avoid integer round-off errors when multiplied by the learning rates.

d) Heuristic reinforcement.

$$r' = r - 100 * isToHighBid \quad (4)$$

To avoid high-level biddings with bad cards there was an extra negative reinforcement on those bids that were above the optimal contract.

**Discount/Decay** How far-sighted should the bidders be? By selecting different values for  $\gamma$  (Eq. ??), different behaviours was obtained. A low value make the learners prefer a short bidding sequence and resulted in guessing into 3NT more often. Maybe a value higher than the normal limit of one will encourage a more informative, long sequence? Eligibility traces was not used.

### 3 Learning bidding systems by using large-scale Q-tables

Some of the ideas above were implemented in a program called “Reese”, after the famous Terence Reese who died 1996. He was known for his excellent books and simple and logical bidding style. Regarding our bidding program there was hope for finding a super-natural bidding system but so was not the case. This program requires a quite powerful computer and at least 348 MB of memory with the described configuration. The experiments can be seen as what can be achieved with these methods in a complex two-learner task and also gives some hints of how to chose learning parameters.

#### 3.1 Introduction

By using the huge database of GIB-results when playing against itself for 717.102 deals, we found a reasonable evaluative function. The database was generated using an early version of GIB but both the declarer and the opponents were not so strong so maybe the result is quite reasonable anyhow. We have used the 128 most common shapes which includes all hands with at most a six-card suit except for the rare 5440, 5530, 6511, 6430 and 6520 shapes. This covers about 91% of the hands and, neglecting the inter-hand shape influence,  $0.91 * 0.91 = 83\%$  of the hand-pairs. The strength was classified into the following 16 intervals (although this is easy to modify):

Class	0	1	2	3	4	5	6	7
hcp	0-3	4-6	7-8	9	10	11	12	13
Class	8	9	10	11	12	13	14	15
hcp	14	15	16-17	18-19	20-21	22-24	25-27	28-37

To reduce memory requirements, learning time and simplify the restriction of learning into allowed bidding systems, the opening bids was defined by the user as a small two page table. Part of the definition of the opening bids in bidsyst.txt is:

```

Shape 0  4  7  9 10 11 12 13 14 15 16 18 20 22 25 28
6421 P      1S      2S
6412 P      1S      2S
6331 P 3S 2D      1S      2S
6322 P 3S 2D      1S      2S
6313 P 3S 2D      1S      2S
6241 P      1S      2S
6232 P 3S 2D      1S      2S
6223 P 3S 2D      1S      2S
6214 P      1S      2S
6142 P      1S      2S
6133 P 3S 2D      1S      2S
6124 P      1S      2S
5521 P      1S      2S
5512 P      1S      2S
5431 P      1S      2S
5422 P      1S      2S
5413 P      1S      2S
5341 P      1S      2S
5332 P      1S      1N      1S 2C 2N 2S

```

It can be represented as a decision tree or ruleset but this introduces errors (due to pruning; about 2%), is larger (five and ten pages respectively) and considerably more complex to read. The bidsyst.txt can be rearranged (preferably with a program) from this matrix representation into an input file with examples which look like this (points, number of Spades, Hearts, Diamonds, Clubs, Opening-bid):

```

0,6,4,2,1,pass
1,6,4,2,1,pass
2,6,4,2,1,pass
3,6,4,2,1,pass
4,6,4,2,1,pass
...
```

Each strength 0.37 is used to generate an example, resulting in  $128 * 38 = 4864$  examples.

The See5, inductive-logic, software (available via <http://www.rulequest.com>) generates the following decision tree:

See5 [Release 1.13]      Wed Mar 14 15:57:11 2001

```
Options:
  Generating rules
  Fuzzy thresholds
  Pruning confidence level 50%
```

Read 4864 cases (5 attributes) from openings.data

Decision tree:

```
pts in [20-38]:
:...s in [5-6]:
:   ...pts in [25-38]: 2S (390)
:   :   pts in [0-24]:
:   :   :   ...pts in [0-21]:
:   :   :   :   ...s in [1-5]:
:   :   :   :   :   ...d in [5-6]: 2S (4)
:   :   :   :   :   :   d in [1-4]:
:   :   :   :   :   :   :   ...h in [5-6]: 2S (4)
:   :   :   :   :   :   :   h in [1-4]:
:   :   :   :   :   :   :   :   ...h in [1-2]:
:   :   :   :   :   :   :   :   :   ...d in [1-2]:
:   :   :   :   :   :   :   :   :   :   ...d = 1: 2S (2)
:   :   :   :   :   :   :   :   :   :   :   d in [2-6]:
:   :   :   :   :   :   :   :   :   :   :   :   ...h = 1: 2S (2)
:   :   :   :   :   :   :   :   :   :   :   :   h in [2-6]: 1S (2)
:   :   :   :   :   :   :   :   :   :   :   :   d in [3-6]:
:   :   :   :   :   :   :   :   :   :   :   :   :   ...h = 1: 1S (4)
:   :   :   :   :   :   :   :   :   :   :   :   :   h in [2-6]:
:   :   :   :   :   :   :   :   :   :   :   :   :   :   ...d in [1-3]: 2C (2)
:   :   :   :   :   :   :   :   :   :   :   :   :   :   :   d in [4-6]: 1S (2)
:   :   :   :   :   :   :   :   :   :   :   :   :   :   :   h in [3-6]:
:   :   :   :   :   :   :   :   :   :   :   :   :   :   :   :   ...h in [4-6]: 1S (6)
... (4 more pages)
```

And the following set of rules (my comments on right of \*):

```
Rule 1: (512, lift 3.9)
      pts in [0-3]
      -> class pass [0.998]
```

```

* Pass with a very weak hand

Rule 2: (132/2, lift 3.8)
    pts in [0-11]
    s in [1-3]
    h in [1-4]
    d in [5-6]
    d in [1-5]
    -> class pass [0.978]
* Pass even up to average and 5 diamonds

Rule 3: (39, lift 3.8)
    pts in [0-12]
    s in [4-6]
    s in [1-4]
    h in [2-6]
    h in [1-2]
    d in [1-3]
    -> class pass [0.976]
* Pass with up to average and 4 spades

...

Rule 47: (20, lift 15.6)
    pts in [10-38]
    pts in [0-19]
    s in [5-6]
    d in [5-6]
    -> class 1S [0.955]
* open 1S with 10-19 hcp and 5-6 spades and 5-6 diamonds

```

The program uses the original bidsyst.txt and it is doubtful if the tree or the rules can be useful to a human reader.

No opponents were taken into account except for the specified opening bids which included preemptive bids like 2 diamonds (showing a strong hand with diamonds or a weak hand with a six-card major) and 3 in a suit showing a weak hand with a six-card suit.

The database contains a vector with the actual number of tricks taken in any suit or no triumph. We have assumed East to be declarer in all cases due to performance reasons. The maximum number of bids in a row was set to four (a pass was automatically added as a fifth bid if applicable) and the number of possible bids limited to sixteen.

Learning rates was encoded as integers and shifted right ten steps, effectively scaling them down with a factor of 1024. Exploration rates was integer coded in promille. A gamma-value of one was used and the initial values of the Q-tables for east and west respectively was a command line argument. The examples were separated in a training and a test set and training examples were selected in random order to avoid biasing effects. The desired number of examples were selected in order from the database except for unrepresentable uncommon shapes which were simply skipped. Exploration was not used during test-evaluations but used during training evaluations.

The score was calculated from the number of tricks taken and final contract. The vulnerability was “none” and dealer was always east. Undoubled part-scores, games, small-slams and grand slams was calculated and contracts going more than two down was assumed to be doubled.

The Q-learning reinforcement rule

$$q_{shape,strength,situation,bid} := q_{shape,strength,situation,bid} + lr * (score - q_{shape,strength,situation,bid}) \quad (5)$$

was used for all four bidding situations (were applicable).  $q$  is the average resulting score having this hand and following the policy,  $lr$  is the learning rate and  $score$  is the resulting score if playing the final contract. The bid with the highest  $q$  was selected in a particular situation unless exploration was actual on which any bid was selected with equal probability. If several bids had the same  $q$ -value one of them were selected randomly.

In practice this meant one out of 2048 q-opener were combined with another of 2048 q-responders and they had to learn together with their partners repeated multiple times depending on the database examples; thence multi-agent reinforcement learning.

### 3.2 Experiments

We tried to find the best possible bidding system by doing three experiments; first a survey run with different parameters, a second survey with some new, better, parameter values and then a longer run with the parameters estimated to be best by the human observer. The parameters to be selected was the learning rate (common for both hands), exploration rate (also common) and initial values for the Q-tables.

In the first run we used learning rates 5, 10 and 20, exploration rate 5, 10 and 20 and Q-values -100 for east and -200, 0 and +200 for west. Each run was repeated three times with different random number seeds (13579, 5799 and 8642 respectively). This resulted in a total of  $3*3*3*3 = 81$  cases. Ten million training examples were selected in each epoch and 200 epochs was run in each case. Each case took about 41 minutes and the total CPU-time was about 55 hours on a 1.5 GHz Intel Pentium IV running Windows 2000 and equipped with 512 MByte memory. Each training occasion required about one microsecond including evaluation.

	$\epsilon = 0.005$	$\epsilon = 0.010$	$\epsilon = 0.020$
<b>Lr = 5/1024</b>	$27.2 \pm 8.7$	$27.6 \pm 0.2$	$30.7 \pm 9.9$
<b>Lr = 10/1024</b>	$24.9 \pm 1.8$	$40.4 \pm 6.1$	$41.7 \pm 3.7$
<b>Lr = 20/1024</b>	$33.8 \pm 3.6$	$50.1 \pm 10.0$	$63.6 \pm 3.1$

Table 2: The average of the last ten test-evaluations, averaged over three runs.  $q_0$  is zero

Every 99th epoch, a bidding system was emitted into two files; responses.txt (115 kB) and rebid.txt (525 kB), the second response was not stored. The generation of the rebid-file takes the opening restrictions into consideration to reduce the size. A log-file (6 kB) with the time, average training-sum and average test-sum was generated and the standard output was redirected into a file (1.4 MB) with the actual biddings from all the test-cases for future reference. About  $2MB * 81 = 162MB$  disk space was used.

After this, a second run was done with 200 epochs with fixed learning rate and exploration but three different initial strategies and finally two long runs with 1000 epochs each making it possible to evaluate the resulting bidding system.

550.000 training examples and 200 test examples (positioned from 710.000 and forward) were used. 109.940 (19.81%) were skipped, reasonably close to the assumption above.

### 3.3 Results

Some test were run with different combination of parameters.

#### 3.3.1 Selecting learning rate, exploration and initial Q-value

The first run was made with a  $q_0$  of 0. The results are plotted in Fig 6 and in Table 2. It seems as the learning rate does not have a huge importance but the exploration rate has. The highest used value for both of them resulted in the best performance; a average value of  $63.6 \pm 3.1$  with an learning rate of 0.019 and an exploration of 0.020, denoting a 20% risk of selecting a random bid. The training and test-curves seems to follow each other but at a different level. The test-curve is always below the training curve although no exploration is used. The test-curve is also more smooth due to the large number of examples (10 million). The experiments were rerun with a  $q_0$  of +200, meaning a un-tested alternative is worth 200 points. Because most bridge-results are in the -100..140 region this is the optimistic approach resulting in an (more) exhaustive search. The behaviour in this case is similar to the behaviour in the  $q_0$  zero-case.

A third run with  $q_0 = -200$  where done with better results. This value (-200) results in less search of alternatives because a reasonable result like -100 is preferred to un-tested alternatives. This probably leads (guides) the learning through a kind of needles eye into reasonable bidding systems and contracts. The best result ( $68.7 \pm 4.9$ ) were obtained with the highest parameter values;



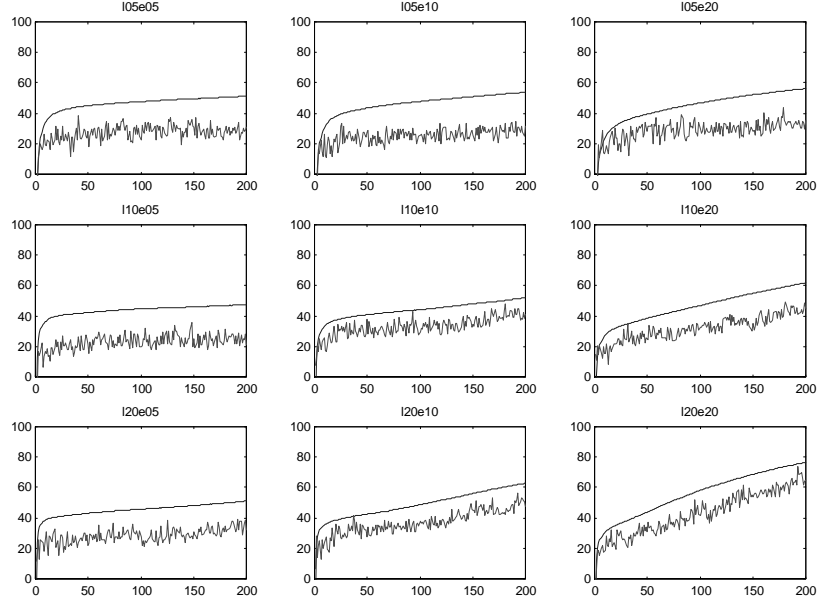


Figure 6: With  $q_0 = 0$  ( $q_0 = +200$  is similar). The smooth curve is the evaluation of the training set and the rugged curve the test set. 200 epochs, each 10 million tries were run, repeated three times and averaged into these curves.

	$\epsilon = 0.005$	$\epsilon = 0.010$	$\epsilon = 0.020$
<b>Lr = 5/1024</b>	$29.3 \pm 5.1$	$36.7 \pm 13.6$	$51.4 \pm 6.9$
<b>Lr = 10/1024</b>	$28.4 \pm 5.6$	$40.9 \pm 6.2$	$54.6 \pm 3.2$
<b>Lr = 20/1024</b>	$27.2 \pm 2.4$	$48.3 \pm 8.3$	$68.7 \pm 4.9$

Table 3: The average of the last ten test-evaluations, averaged over three runs.  $q_0$  is -200

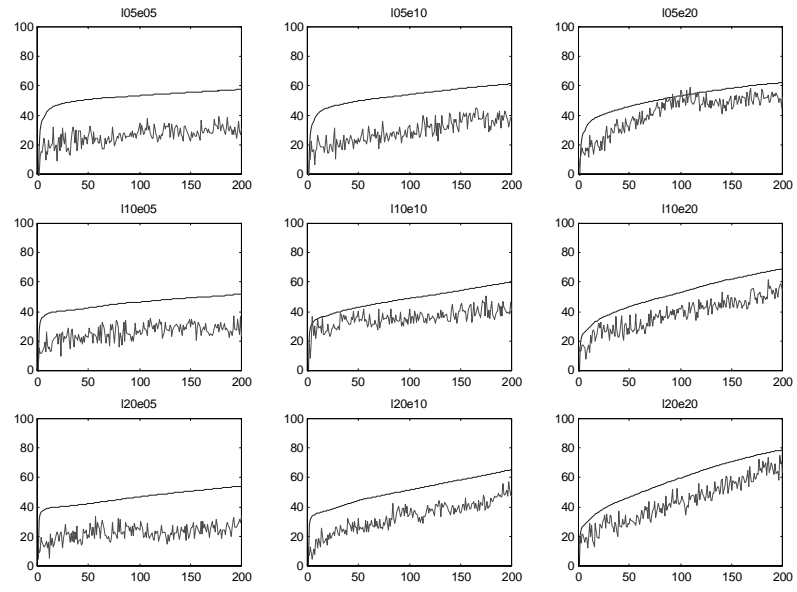


Figure 7: Runs with  $Q_0 = -200$ . The smooth curve is for the training set and the rugged curve the test-set.

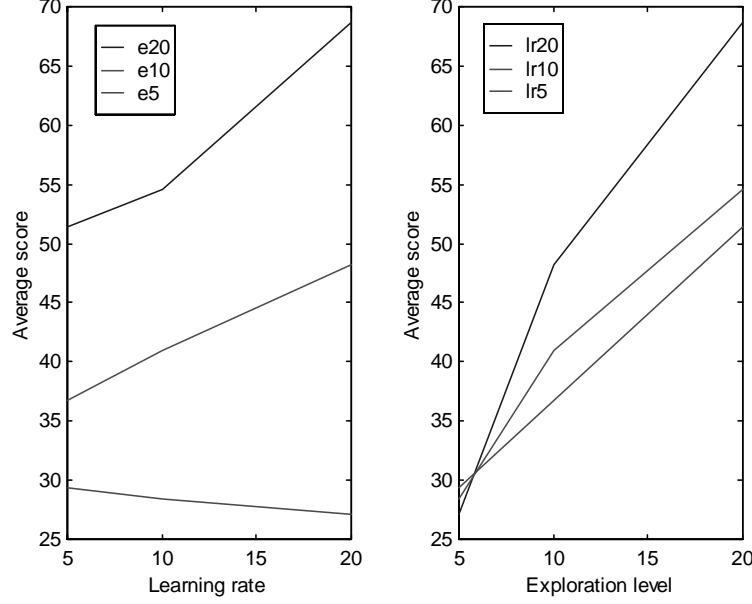


Figure 8: A high value of the exploration and the learning rate improves fast learning. High exploration seems to be most important.  $q_0$  is -200.

$Learning\_rate = 20/1024 \approx 0.019$  and  $\varepsilon = 0.020$  as usual. The table is also graphically illustrated in Figure 8.

A rerun with a higher learning rate was done and shown in Figure 9. The plot shows the average of three runs. Now the best results were instead obtained with  $q_0 = 200$ . Obviously there is a combination-effect on these parameters with high learning combines best with high  $q_0$  and vice versa.

A long, single run with lower learning rate but high exploration is shown in

Figure 10. The levels seem to more or less stabilize around 100.

The single learner Q-learning theory says that convergence is assured if the learning rate is reduced slowly enough. An attempt to improve the average score is done by reducing the learning rate, with a decay factor. In our cases, the learning seems to drop off at about epoch 150. By setting the learning rate to  $20/1024$  at this point two experiments were run; one with slow decay (Figure 11) and one with high decay-rate (Figure ??). There are jumps in the performance and if viewed in detail those jumps are positioned according to table 4 in the slow-decay case. When the learning rate steps down to small values, the required difference between score and q-value has to be at least one when scaled with the learning rate according to Equation 5. The first noticeable effects appear

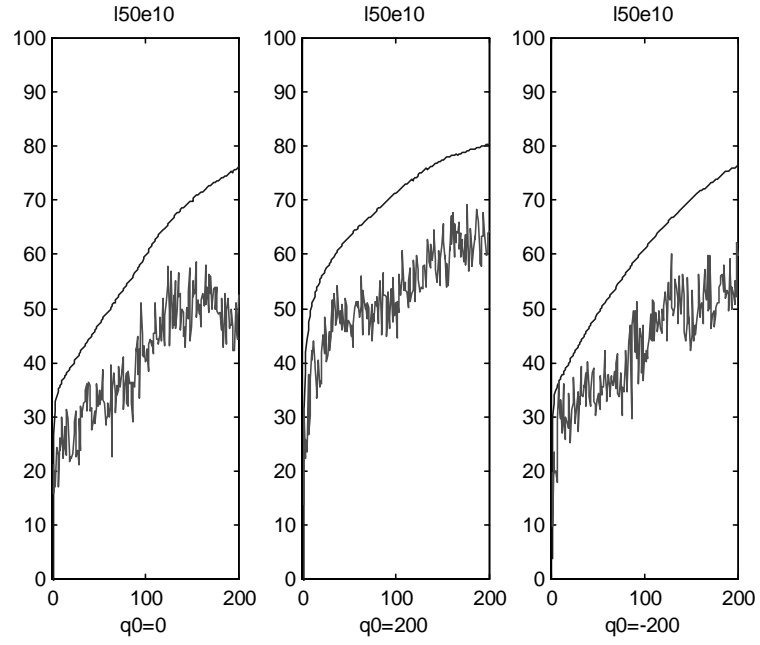


Figure 9: A rerun with a higher learning rate.

Epoch	Training-level	Lr	Difference
849	84.7	6/1024	170.7
951	83.4	5/1024	204.8
1074	79.1	4/1024	256
1221	67.3	3/1024	341.3
1413	45.7	2/1024	512
1683	35.6	1/1024	1024

Table 4: Integer round-off error

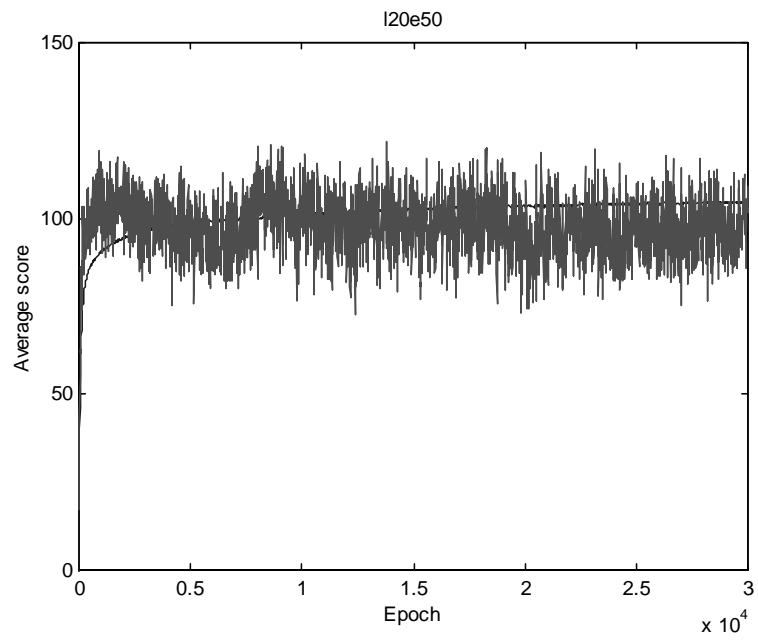


Figure 10: A 5-day CPU run of learning. Lr is  $20/1024$  ,  $\varepsilon = 0.050$  and  $q_0 = -200$ . The maximum training case has a score of 104.89, max test-score is 121.7 and average on the last ten is 94.65.

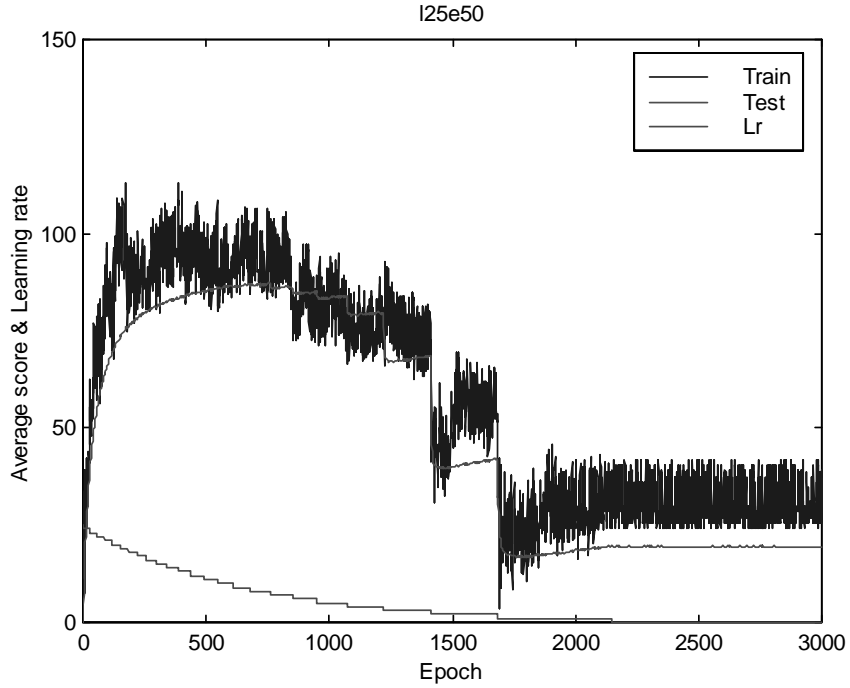


Figure 11: The scores are actually decreasing step-wise when the learning rate reduces.

when the learning rate is  $6/1024 \approx 0.006$  and the required difference is then  $1024/6 \approx 170.7$ . In bridge terms this means that the learner is not able to see the difference between part-scores and concentrates on the large contracts like games and slams. When learning rate gets below  $3/1024$ , the game-contract abilities are also lost. This problem with learning could be avoided by using floating-point representation but this consumes more memory (4 times in this case;  $384 * 4 = 1536$  MB) and requires more computing time.

Some of the interesting examples (many removed) from the bidding are presented in table 5.

At a later position (1000) under the the same training conditions, a bidding system was emitted. Only a very small part of it (the response when partner has started with Pass) is presented in table 6. It is illustrative because it indicates what suit to open with and which strengts are associated with a particular shape. Many cases are innovative and maybe unknown to most human bridge-bidders. Some examples are: opening with only seven h.c.p. when having six spades, two

Deal	East	Hp	West	Hp	E	W	E	W	Score	My comment
2.	2614	16	2353	15	1H	3N	4N	7H	1510	(Good but strange
5.	2434	9	5323	14	P	1S	2C	2H	110	how 7H was found)
6.	4522	10	1255	13	P	1C	1S	2C	110	
7.	3145	7	2434	15	P	1H	1N	P	90	
12.	2542	4	4252	13	P	1D	P		110	
13.	1543	9	5332	16	P	1S	2C	3N	400	
16.	3514	12	3253	14	1H	3N	P		430	
23.	4225	9	5323	11	P	P			0	(missed part-score in spades)
24.	3334	6	3343	14	P	1D	P		-50	
26.	5152	4	3613	10	P	1H	P		110	
27.	4243	2	2641	22	P	3N	P		-100	(4H better)
28.	2434	12	4243	12	1H	3N	P		-50	
29.	3613	13	3262	15	1H	3N	P		520	(missed a slam)
30.	3235	19	4432	6	1C	P			130	
31.	2614	4	5251	4	3H	P			-800	(opening bids defined by bidding system)
33.	5134	6	3622	11	P	1H	1N	2H	-50	(1N with 5 S?)
34.	3514	11	4135	13	P	1C	1N	P	150	
37.	2353	13	4324	9	1D	P			70	(P with 9 hcp?)
42.	3334	10	5512	13	P	1H	1S	4S	480	(smart bidding)
52.	2623	10	2236	7	2D	2H	P		110	(2D shows weak H/S or strong with D, 2H a good bid)
53.	2254	13	2632	10	1D	3N	P		400	(practical bridge, 4H probably safer)
54.	4342	2	3514	11	P	1C	1D	1H	-50	
56.	3163	12	5422	15	1D	3N	P		430	(4S safer?)
60.	4144	0	3325	17	P	1N	2H	2S	-100	(2H with 0 hcp and 1 h?)
64.	4441	9	5431	11	P	1S	1N	2S	110	
65.	3325	9	3244	16	P	1D	1N	3N	400	
69.	3433	14	4234	11	1H	P		140		
73.	4252	8	1633	10	P	1H	1S	2H	140	
76.	2551	7	4153	16	P	1D	2D	2S	110	
77.	4324	18	3352	10	1C	1N	P		180	(P with 18?)
78.	3154	13	4135	9	1D	P			130	(often P with avg. strength and support)
79.	3163	17	1525	12	1D	3N	P		490	(missed 6D)
80.	2542	13	6133	15	1H	3N	P		490	(often bids 3 NT with 5-card majors)
82.	1426	13	6241	9	1C	1N	P		90	
83.	2533	12	3433	12	1H	4H	P		420	
84.	3244	13	5332	13	1C	3N	P		460	
85.	2362	10	4432	15	3D	P			130	(found to P with 15)
87.	2344	7	3136	18	P	2S	2N	3N	430	
88.	2434	11	1642	12	P	1H	4C	4H	450	
89.	4252	10	1444	15	P	1H	1S	3N <sub>3</sub>	400	
90.	3253	12	4351	12	P	1D	1S	P	140	
91.	3433	20	4351	9	2C	3N	P		460	
93.	3433	4	3244	13	P	1D	P		-50	
95.	1363	13	3244	15	1D	3N	P		460	
97.	3352	16	2344	15	1N	3N	P		520	(to strong hand for 3N?)
99.	2524	9	5422	12	P	1H	P		170	

Table 5: Some typical examples of bidding from a reasonably good run with  $Lr=0.02$  and exploration = 0.05,  $q_0$  is set to -200. An average score of 87 on the training set and 100 on the test set is achieved after 562 epochs

# Response to P

Table 6: Some of the responses when partner has started with pass, showing a weak hand without extreme shape.

hearts and a four card minor (club or diamond) suit, opening with the lower of adjacent suit if weak (then pass) or strong (continue bidding) and higher suit if medium strength, pass if balanced and less than fourteen h.c.p, open one or three NT with strong hands and any shape, 2 hearts and spades(stronger) shows a good hand with eight+ cards in the majors, open with nine h.c.p and unbalanced with six-card major or diamonds or when having a five-five shape,...

	0	4	7	9	10	11	12	13	14	15	16	18	20	22	25	28
3631 :	P	P	P	P	1H	1H	1H	1H	1H	1H	1H	2H	3N	3C	3N	?
3622 :	P	P	P	P	P	1H	1H	1H	1H	1H	1H	2C	3N	3N	3N	?
3613 :	P	P	P	P	1H	1H	1H	1H	1H	1H	1H	1H	2C	2D	?	?
3541 :	P	P	P	P	P	1H	1H	1H	1H	1H	1H	3D	2S	2H	2S	?
3532 :	P	P	P	P	P	P	P	P	1H	1H	1H	1N	3N	3N	2C	?
3523 :	P	P	P	P	P	P	P	1H	1H	1H	1H	1H	3N	2C	2H	3C
3514 :	P	P	P	P	P	1C	1H	1H	1C	1H	1H	1H	3N	3N	3C	?
3451 :	P	P	P	P	P	1D	1D	1D	1D	1D	1D	1N	2H	3D	?	?
3442 :	P	P	P	P	P	P	P	P	1D	1D	1H	1N	3N	3N	3N	?
3433 :	P	P	P	P	P	P	P	P	1C	1H	1H	1N	1N	3N	2H	3N
3424 :	P	P	P	P	P	P	P	1C	1C	1C	1H	1N	3N	3N	3N	?
3415 :	P	P	P	P	P	1C	1C	1C	1H	1C	1C	1H	1N	3N	?	?
3361 :	P	P	P	P	P	1D	1D	1D	1D	1D	1D	1N	1N	3N	3S	?
3352 :	P	P	P	P	P	P	1D	1D	1D	1D	1N	1D	3N	3N	2D	?
3343 :	P	P	P	P	P	P	P	P	1D	1D	1D	1N	3N	3N	3N	2S
3334 :	P	P	P	P	P	P	P	1C	1C	1C	1C	1N	3C	3N	3N	?
3325 :	P	P	P	P	P	P	1C	1C	1C	1C	1C	1N	3N	3N	3N	?
3316 :	P	P	P	P	1C	1C	1C	1C	1C	1N	1C	1C	3H	3N	?	?
3262 :	P	P	P	P	P	P	1D	1D	1D	1D	1N	3N	3N	3N	2N	3H
3253 :	P	P	P	P	P	P	P	1D	1D	1D	1D	1N	3N	2C	3N	?
3244 :	P	P	P	P	P	P	P	1C	1D	1C	1D	1N	3N	3N	3N	?
3235 :	P	P	P	P	P	P	P	1C	1C	1C	1C	1N	2C	3N	3D	?
3226 :	P	P	P	P	P	P	1C	1C	1C	1C	1N	1N	3N	3N	3H	?
3163 :	P	P	P	P	1D	1D	1D	1D	1D	1N	1D	2C	2N	3N	2D	?
3154 :	P	P	P	P	P	P	1D	1D	1C	1C	1C	1C	2C	3N	3N	?
3145 :	P	P	P	P	P	1C	1C	1C	1C	1C	1C	1N	3N	2C	2S	?
3136 :	P	P	P	P	P	1C	1C	1C	1C	1C	2C	2S	3N	3N	?	?



### 3.4 Conclusions

The runs show it is possible to find a reasonable bidding system in limited time. A relatively high learning rate combined with high exploration resulted in the best performance. The test-set evaluations are quite close to the training evaluations, indicating that no over-fitting is yet achieved yet. An initial  $q_0$  value set to a reasonable lower acceptable threshold seem to steer the learning into realistic areas. Good runs with decreasing learning rate was not possible to achieve due to the integer nature of this implementation.

The resulting bidding system, studied in detail is also interesting. The responder have learnt “understood”, that an opener shows hearts or spades or a strong hand with the two diamond opening bid and thereby never passes and do not bid two hearts with heart support. The no-trump bidding is also interesting; two-bid responses seems to be sign off, two NT is never used and three in a suit seems to be highly conventional and never passed by the NT-opener. Three NT is often the response with medium strength hands and not an extreme shape and four in suit is a transfer bid. Maybe an interesting NT-bidding part of the system can be developed if the learning were concentrated on this?

Anyhow; the learnt system is not practically useful until the defensive bidders are put into consideration.

### 3.5 Author information:

Kalle Prorok

Department of Computing Science

Umeå University

SE-901 87 UMEÅ

Sweden

Tel +46 90 786 50 18

Fax +46 90 786 61 26

Home +46 90 12 99 96, 070 - 3 33 35 37

E-mail: [kallep@cs.umu.se](mailto:kallep@cs.umu.se)

<http://www.cs.umu.se/~kallep>

### 3.6 Keywords:

Reinforcement learning, Q-learning, Multi-Agent systems, machine learning, bridge, bidding, auctions, ...

#### 3.6.1 Acknowledgments and Prologue

Thanks to Lars-Erik Janlert, Patrik Eklund, Per-Åke Wedin, Bo Kågström, Göran Broström, Stephen Hegner, Thomas Hellström, Lennart Edblom, Iradj Roozbeh, Andreas Hed, My family and Bridge partners.

## References

- [1] C. E. Shannon, “Programming a computer for playing chess,” *Philosophical Magazine*, vol. 41(4), pp. 256–275, 1950.
- [2] A. M. Turing, “Computing machinery and intelligence,” *Mind*, vol. 59, pp. 433–460, 1950.
- [3] M. L. Ginsberg, “How computers will play bridge,” *The Bridge World*, June 1996.
- [4] M. L. Minsky, “Steps towards artificial intelligence,” *Proc. Of the Institute of Radio Engineers*, vol. 49, pp. 8–30, 1961.
- [5] M. D. Waltz and K. S. Fu, “A heuristic approach to reinforcement learning control systems,” *IEEE Transactions on Automatic Control*, vol. 10, pp. 390–398, 1965.
- [6] T. Landelius, *Reinforcement Learning and Distributed Local Model Synthesis*. PhD thesis, Linköping, Sweden, 1997.
- [7] R. S. Sutton, A. G. Barto, and R. J. Williams, “Reinforcement learning is direct adaptive optimal control,” in *Proc. Of the American Control Conf.*, pp. 2143–2146, 1991.
- [8] P. H. Winston, *Artificial Intelligence*. Addison-Wesley, 1977.
- [9] S. Alfredsson and J. Malmström, *Trappan (in Swedish)*. Tolg, 360 40 Rottné: Svenska Bridgeförlaget, 1994.
- [10] M. Nilsson, *Modern Standard*. Stockholm: Sveriges Bridgeförbund, 1978.
- [11] I. Frank, *Search and Planning Under Incomplete Information*. London: Springer-Verlag, 1998.
- [12] M. L. Ginsberg, “GIB: Steps toward an expert-level bridge-playing program,” *Proceedings from Sixteenth International Joint Conference on Artificial Intelligence*, pp. 584–589, 1999.
- [13] T. Andrews, “Double dummy bridge evaluations,” 1999. <http://www.best.com/thomaso/bridge/valuations.html>.
- [14] L. Sterling and Y. Nygate, “Python: An expert squeezer,” *Journal of Logic Programming*, vol. 8, pp. 21–40, 1990.
- [15] M. R. Björn Gambäck and B. Pell, “Pragmatic reasoning in bridge,” Tech. Rep. 299, University of Cambridge, Computer Laboratory, Cambridge, England, 1993.
- [16] H. S. Bierman and L. Fernandez, *Game Theory with Economic Applications*. Addison-Wesley, 1998.

- [17] G. Carley, "A program to play contract bridge.," Master's thesis, Dept. of Electrical Engineering, Massachussets Institute of Technology, Cambridge, Massachussets, 1962.
- [18] J. MacLeod, "Microbridge - a computer developed approach to bidding.," *Heuristic Programming in AI - The First Computer Olympiad*, pp. 81–87, 1991.
- [19] A. Stanier, *Decision-Making with Imperfect Information*. PhD thesis, Essex University, 1977.
- [20] E. Lindelof, *The Computer Designed Bidding System - COBRA*. No. ISBN 0-575-02987-0, London: Victor Gollancz, 1983.
- [21] J.-R. Vernes, "The law of total tricks," *The Bridge World*, 1969.
- [22] R. Hillyard, "Extending the law of total tricks," <http://www.bridge.hotmail.ru/file/TotalTricks.html>, 2001.