Evaluation of a chess position using computing with words

M.Satyajit
K.B.N.Vidyuth

1 M.Satyajit, F-46, Gitam boys hostel
Gitam, Rushikonda , Visakapatnam 530045
malugu.satyajit@yahoo.com

2 K.B.N.Vidyuth, F-51, Gitam boys hostel
Gitam, Rushikonda , Visakapatnam 530045
vidyuth_koniki@yahoo.co.in

Abstract

Though, chess has been considered as an Art by many experts, all the programming effort till now has been concentrated in the strict scientific perspective of bi-valent logic. The objective of this paper is to point out the fallacies of such a scheme by giving an account of the contradicting, intuitional aspects in chess. Emphasis is given to the positional elements, their classical implementation and how computing with words can be utilized to achieve a more promising solution. An account of perceptions in chess and how they can be implemented through computational theory of perceptions (CTP) is also given. Finally the advantages and the scope of this scheme are outlined.

Key words: Chess programming, Computing with words, Soft Computing

1. Introduction

Chess has been considered as the *Drosophila Melanogaster* of AI as it is to genetics because it was utilized as a testing tool for many break throughs. The greatest challenge for the AI researchers is to defeat human players in Chess- which can be regarded as the best demonstration of human mental faculty -thinking, reasoning and rationalizing. With deep blue defeating Kasparov in 1997, the issue has been considered to be conquered. Yet, still computer's positional perception and technique is far from perfect. This may be the reason for its occasional defeats with human players. Grand masters consider that a computer can never reach the positional understanding of a human expert. This may be true as the bivalent logic used in chess programming is a poor imitation of the human brain, which can evaluate and integrate very complex data easily than a computer. Hence i propose to use soft computing techniques- computing with words, computational theory of perceptions in addition to the classical approaches in chess to achieve more human like thinking.
What I propose, here is not the total abandon of the numerical approach but utilizing computing with words and CTP in allegiance with the traditional approaches. A chess player though a great one never goes by intuition alone. He needs to calculate at least some variations perfunctorily to get a sound feel about the position. So, tactical possibilities are to be done in numbers and positional elements in words. This is a more human like approach to the most humane game.

2. Chess Programming

Broadly dividing, the game of chess can be divided into three elements
- Tactical- calculational
- Positional- perceptional
- Psychological- dynamic

2.1 Tactical elements

Tactical elements are forced series of moves that brings a favorable position to the initiating player. Examples are fork, pin, double attack and sequence of captures. Tactical advantages are like being a piece ahead or two pawns ahead or a forced check mate on the next move.

Every chess program contains an evaluation engine which calculates the best move in the given position. The algorithms used for it are mini-max search, alpha - beta pruning and several other improvements on it. Every chess piece is assigned some value like queen-9, rook -5, bishop and knight 3 and pawn 1(king has 4 points in end game). The searching is done using game trees. Each of the leaf nodes has a numerical value attached to it. For example if a node says +1.5 it means that white is one and a half ahead. The search engine chooses the best amongst them. The algorithms used for finding the best move are mini-max search, alpha - beta pruning and several other improvements on alpha - beta like quiescent search.

Tactical possibilities in a position are calculated in terms of numbers. This is not a bad practice in fact chess players do think in terms of piece values. In this particular field we can say that computers are far superior to men because the depth, accuracy and speed with which a computer does mathematical like calculations is no match for the human players.
Fig 1: Evaluation function in chess program Fritz8, note the variations and the corresponding values.

2.2 Positional elements

Positional elements are generally more volatile and abstruse. In chess programming jargon they are called patterns and there are more than some 50,000 such patterns. (Some of them are listed at a later place). Positional elements are very much related to perceptions but the weird thing is that they are implemented in numbers. For each identified pattern a value is associated like a knight is center has 0.5 points more than a dull knight.

This is an inferior way of doing things. A human player percepts his knight is stronger because it is placed in center. He does not even in his wildest dreams think that it is 0.5 more powerful than opponent's. This gives him more flexibility, generality and efficiency. Humans have the remarkable capability of identifying the most feasible variations by the first glance itself. Unlike computer he will not brood over useless silly variations. He has the capability to know which are silly; computer lacks that seemingly basic faculty. This ability is called positional understanding; computers try to and are implementing this but with crude numbers. This human like perception can be included in
chess programs by computing with words. This opens up completely new doors which are previously thought almost impossible.

2.3 Psychological elements

Psychological elements in chess are really complex; they are largely dependent on how an opponent thinks and his basic style of play. Great understanding of human nature is required for the mastery in this-- out of 64 squares art. We can safely say that till no system till date has this capability. (So even computers master the positional play, yet chess is not fully conquered!).

3. Computing with words

In a way, we can say that all through the evolution of computers, the greatest target has been to emulate the human brain. Though, in some domains like calculation and information storage they have far surpassed the human beings but in perception, creativity, imagination etc... they lag far behind. Seemingly simple tasks like summarization, parking a car are onerous tasks for computers. This may be not due to the lack of processing speed but the bivalent, Aristotelian logic used in all the computations is unable to cope up with the real world dynamic problems. Soft computing and its sub fields are a great step towards mimicking the human brain. Soft computing is aimed at an accommodation with the pervasive imprecision of the real world, exploiting the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution cost, and better rapport with reality.

A drastic change which is bound to revolutionize the concept of computing is introduced in computing with words, in which the objects of computing -- words and perceptions are drawn from the natural language. Computing with words is inspired by the remarkable human ability to perform a wide variety of physical and mental tasks without any measurements and any computation. Manipulation of perceptions plays a key role in human recognition, decision and execution processes. As a methodology, computing with words provides a foundation for a computational theory of perceptions a theory which may have an important bearing on how humans make and machines might make rational decisions in an environment of
imprecision, uncertainty and partial truth.

4. Positional elements in chess

The following is the list of important positional elements (patterns) in chess and their approximate value in the evaluation function. These values change from programmer to programmer and its strength. The given values are only arbitrary, they are not absolutes.

Table 1: Description of the pattern and its value

<table>
<thead>
<tr>
<th>Name of the pattern</th>
<th>Brief description</th>
<th>Value in evaluation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong center</td>
<td>Control on the square e4,d4,e5,d5</td>
<td>+0.35</td>
</tr>
<tr>
<td>Rapid piece development</td>
<td>More number of pieces are in active play</td>
<td>+0.20 for each piece</td>
</tr>
<tr>
<td>Piece co-ordination</td>
<td>All the pieces are in concentrated together</td>
<td>+0.30</td>
</tr>
<tr>
<td>Mobility of pieces</td>
<td>The more number of squares a piece can move</td>
<td>Depends on the number of squares</td>
</tr>
<tr>
<td>Castle Position (available or not; if available done or not)</td>
<td>King is considered to be more safe if it is castled</td>
<td>-0.50 if castling option lost</td>
</tr>
<tr>
<td>Concentration of pieces on the opponent’s king (also called tropism)</td>
<td>Mating chances- if more number of pieces are clustered around the opponents king</td>
<td>More than 4 pieces then +1.0</td>
</tr>
<tr>
<td>Weakness in the opponent’s castled position</td>
<td>Moves like h6, g6 or vanishing of the pawn make king vulnerable</td>
<td>+0.25 for each</td>
</tr>
<tr>
<td>Pair bishops in a open position</td>
<td>Bishops power escalates</td>
<td>+0.50</td>
</tr>
<tr>
<td>Pair knights in a closed position</td>
<td>If the position is closed knights have more scope</td>
<td>+0.35</td>
</tr>
<tr>
<td>Double rooks on a open file</td>
<td>Open file control gives initiative</td>
<td>+0.75</td>
</tr>
<tr>
<td>Double rooks on 7th rank</td>
<td>7th rank is the base for all the pawns and king stays in 8th rank, so having control on that gives a clear edge</td>
<td>+1.0</td>
</tr>
<tr>
<td>Opposite color bishops in ending</td>
<td>They have more chances of draw, even in case of a pawn down</td>
<td>Neutralizes opponent’s advantage, if no other advantage is present.</td>
</tr>
<tr>
<td>Strong squares (which cannot be threatened by a pawn)</td>
<td>Pieces posted on advanced strong square have a menacing affect</td>
<td>Depends on the rank and proximity to the centre. Center squares have +0.75</td>
</tr>
<tr>
<td>Pawn structure</td>
<td>They hinder each others movement and are easy targets for attack</td>
<td>-0.5 for doubled pawns and -0.75 for tripled pawns</td>
</tr>
<tr>
<td>a) Doubled or triple (pawns on same file)</td>
<td>Since pieces have to support them, they are a hindrance</td>
<td>-0.25</td>
</tr>
<tr>
<td>b) Isolated pawns (pawn without a support of a pawn)</td>
<td>They are always a threat for potential queening</td>
<td>+0.25, +0.5 in endgame</td>
</tr>
<tr>
<td>c) Passed pawns (unobstructed by opponent’s pawns)</td>
<td>More number of clusters demand more protection</td>
<td>-0.25 for each island</td>
</tr>
<tr>
<td>d) Pawn islands (cluster of pawns)</td>
<td>They promote space advantage</td>
<td>+0.35</td>
</tr>
<tr>
<td>e) Mobile pawns (chance to advance)</td>
<td>The four squares in front of the pawn are controlled</td>
<td>+0.30</td>
</tr>
</tbody>
</table>

These patterns are not absolutes like tactical possibilities. We cannot always say that some pattern has some rigid value. Some seemingly bad characteristic may be good in
some position due to a subtle difference. Hence programming these patterns is an arduous task and the strength of the program generally depends on these things rather than the search techniques and the knowledge of the program.

A programmer on his volition chooses values for the above patterns and for other patterns. For a specific position in the game or for a position arising in thinking, the tactical values and these positional values are summed up. (An example is given in the following section) The evaluation engine chooses the best possible value among the leaf nodes.

For an improvement in his program, a programmer arbitrarily assigns some new values and compares with his older program. So this is more a trail and play method rather than a principled approach.

5. Perceptions in chess

In chess, basically every thing is a matter a degree and perceptions. They are no rigid absolutes. The only absolute is the given position and most of the evaluations change from position to position. For example generally being a pawn ahead is considered as an advantage but the following position, though white is a pawn down he has the advantage.

Fig2: A game position between two humans M.Satyajit-IM A.Sastry

The musings of the players would typically be like these.
Human 1 (white): My pieces are more active, I have no weakness in my position, bishops are menacing. I should utilize my lead in development to bind the opponent’s position. I should castle quickly and bring my rooks into play to increase the hold on the position. The white bishop is more active than the black one.

Human 2 (black): Though I am pawn up I have several weaknesses, I should try to make pieces active as quickly as possible. Exchanging of pieces will relieve the pressure. Endgame is unfavorable for me as all pawns are weak. I should find a safe position for my king and to connect the rooks. I can never castle king side due to weakness (h7, f6, f7).

The way computer thinks is given below

Computer: Tactical --- white pawn down                  -1.0
Positional--- White:
   Open board two bishops   +0.50
   Lead in development     +0.20
   Black: (so add for white)
   Doubled pawns           +0.50
   More pawn islands       +0.25
Total---                  +0.45

The italicized words – active, weak, never denote the perceptions of a sensible human player. When we compare this with the computers evaluation the differences are glaring. Humans can percept his strong points and weaknesses and what actions are to be taken. He can formulate a plan depending on it. Whereas computer thinks only in numbers, it recognizes patterns and gives score accordingly. The program merely substituted assigned values for the patterns that are given by the programmer. It has no means to know what is happening on the board. So in a general position a computer plays or thinks move to move without a general plan. Moves are generated at random and are tried for a better score, it has no means to know what to think any why to think. Surely this is due to obsession with numbers.

A programmer follows the following scheme to make a computer choose the moves which gives it positional edge. He gives higher values to the patterns that are considered to be advantageous (according to his view of importance to the
established principles). So while considering moves a program chooses automatically higher valued positions. But this scheme has two disadvantages. First not always a position corresponds to the list of patterns (no matter how big it is). The problem in chess is that a slight variation in the position may change its evaluation drastically. Another problem the values assigned for a piece in a pattern may be less or more in the given position. As noted earlier the value of a piece changes according to the position. Finding all such position is nearly an impossible task. It is in line with solving the problem of move finding in chess with brute force search.

6. Disadvantages of the classical approach

1. Improving a program is more a random approach.(Regarding positional elements) A programmer can merely guess what could be wrong and then try some arbitrary values to improve that.

2. The number of patterns are above some 50,000 and it is a cumbersome task to make a program recognize each of these patterns and to assign values accordingly.

3. A computer most of the time thinks of the trash variations, more efficiency can be obtained if we are sure that some variations never lead to an advantage.

4. There is no general plan for a computer; it plays move to move. Any decent chess player agrees that doing is simple immaturity. More over it’s a waste of time.

7. Why do we need computing with words?

- The available information is not enough to justify the usage of words- positional patterns and their value assigning is not enough.

- When precision carries a cost and low cost solution can be obtained- By using numbers only numbers we need to delve deep into all trash variations, by obtaining positional sense through CW we can achieve reduced cost (as humans do)

- When the expressive power of words is greater than the expressive power of numbers- here words like active can be more expressive than the reliance of numbers
8. Implementation through computing with words

A key idea in CWP is that of dealing with perceptions through their descriptions in a natural language. In this way, computing and reasoning with perceptions is reduced to operating on propositions drawn from a natural language. In computing with words and perceptions, the objects of computation are words, propositions, and perceptions described in a natural language. The following is the list of the words that can be used in chess for computing

Regarding pieces and pawns

Activity-- Very active, Active, Normal, Dull Dead
Strength – Very strong, Strong, Natural, Weak, Very weak
Centralization- Centralized piece, Controlling center piece, Not centralized piece
Attackability- Attacking piece, Neutral piece, Defending piece
Infiltration- Advanced, Neutral, Backward
Coordination- Well coordinated, normal, Disordered

Regarding pawn structures and position

Closed
Open
Centered game
Attacking
Positional
Endgame
Opening
Middle game

The perception of a player -- *My bishop is stronger than opponent’s*

In PNL, a proposition, $p$, drawn from a natural language, NL, is represented as a generalized constraint, with the language of generalized constraints, GCL, serving as a precisiation language for computation and reasoning. Here the prepositions would be --- strength, my bishop, opponent’s

GCL is generated by combination, qualification and propagation of generalized constraints
given by      X isr R  
     X=constrained variable  
    isr=copula  
     r=type identifier  
    R=constraining relation  

Here it would be  
X,Y isv R  
X= my bishop  
Y=opponent’s bishop  
v= veristic  
R= stronger

The protoform -- the abbreviation of the prototypical form
would for this perception

( A(B), A(C)) is R

Where,
A=strength  
B =my bishop  
C= opponent’s bishop  
R = much stronger

This is an example for a single perception. In a real game,
around 10 or so perceptions should be coped up and finally
the so called feel for the position can be deduced.

For the s/w simulation using CW the following steps could
be taken.
• Identify all the words
• Form general principles (and give relative values for each
of them) this forms the rule set.
• According to the position form prepositions, GCL and
protoforms. And give this value to the leaf nodes. Name
this as positional judgment.
• An evaluation function first must check the positional
judgment and then look for the leaf nodes evaluation. If a
variation is positionally very weak then we can avoid it in
the start itself.

9. Advantages of this approach

1. Positional playing strength increases because computer
   knows what it
   plays.
2. Can learn in terms of concrete principles and may be
3. This can also aid in forming a specific style of play. (For example the program Chess master 8000 has an option for playing in various styles by varying its choices in attack defense piece value. But it a bad example, now we can get a real style of play).
4. The kind of tactical possibilities that are to be calculated can also be reduced.

Circullam Vitae

The author of the paper M.Satyajit is a national chess player, with many laurels at the state and national levels. He has represented his state in Under-14, Under-15, Under-19 and Men. He is the current captain of his university. His international rating is 2142.

Both the authors are the students of 3rd year Computers Science and Engineering in Gandhi Institute of technology and management, Vishakapatnam, AP, India. The authors have started working on this as a project for their final semester.

References

Papers:

1 Knowledge engineering in chess programming- Evaluation of a chess position-criteria and procedure by M.Satyajit, K.B.N.Vidyuth at Tornado, Gitam, Visakapatnam

2 From computing with numbers to computing with words by M.Satyajit, K.B.N.Vidyuth at Nasmotics, Gitam, Visakapatnam

3 Artificial Intelligence procedures used in Chess programming by M.Satyajit, K.B.N.Vidyuth at head start’05 at CBIT, Hyderabad

4 Causality is Undefinable – Toward a Theory of Hierarchical Definability
   by Professor Lotfi A. Žadeh
5 Precisicated natural language an artical AI magazine by Professor Lotfi A. Zadeh

6 Test score semantics for natural languages by Professor Lotfi A. Zadeh

7 From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions by Professor Lotfi A. Zadeh

8 A New Direction in AI--Toward a Computational Theory of Perceptions by Professor Lotfi A. Zadeh

Websites:

1. www.chessbrain.net/beowulf THEORY.html

2. www.wikipedia.com

