Human-like Chess Playing Program

Petro Gordiievych

What is the different between computer and human way to play chess?

“While current computers search for millions of positions a second, people hardly ever generate more than a hundred. Nonetheless, the best human chess players are still as good as the best computer programs. Although this model operates excellently in computer programs, it has very little realism where human thinking is concerned. It is probabilistic and in most task environments the generation of all possibilities even to the depth of one „move” is unrealistic. In making an investment decision, for example, one cannot normally generate all imaginable ways to invest and heuristically select the best: there simply exist too many ways to make the decision. This is why heuristic search models are too coarse to be realistic models of the mind. Much more sophisticated analysis is required in order to explain human problem-solving behaviour” (Saariluoma 1998).

Chunking

We consider term „chunking" as process whereby chess pieces are combined into groups. A „chunk" is simply a group of some of the chess pieces that appear on a chessboard and the action of „chunking" is the grouping together of chess pieces.

Chunking

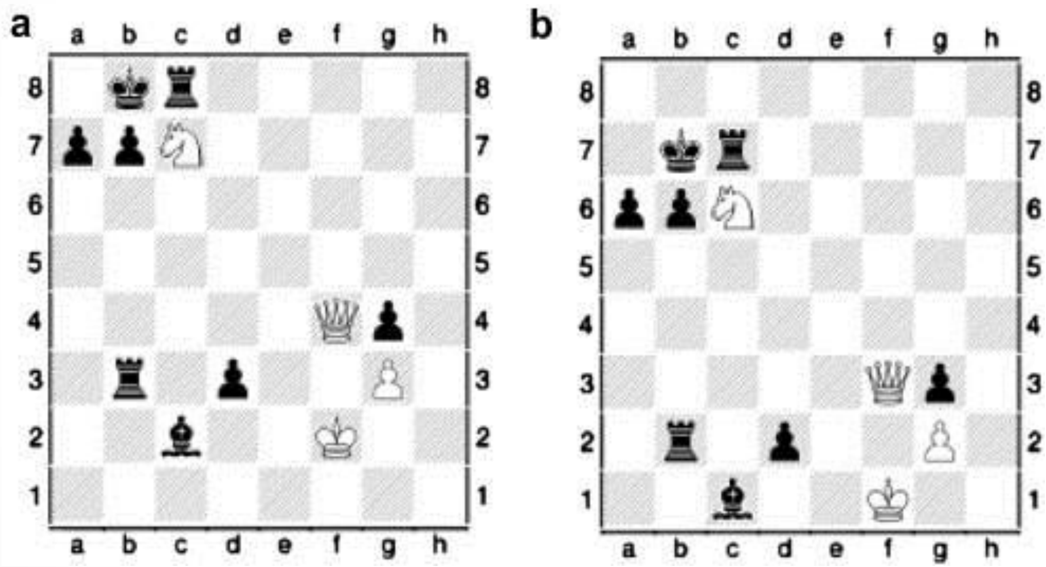
Much of the evidence for chunking in chess is taken from psychological experiments such as de Groot's memory test on expert and novice players. In this well-known experiment de Groot tested three classes of chess player: Grandmaster plus Master, Expert and Class „A" player, (a „Class A player" is a good chess player, but below expert level), by showing them a chessboard configuration from an unfamiliar game with twenty-two pieces on average, for a few seconds (de Groot 1978). The subjects were then asked to reconstruct the configurations, either verbally or on another board. The experiment was repeated by Chase and Simon but included a novice group. The results showed Masters scoring 81% correct, Class „A" players 49% and the novices 33%. But when the positions were randomised each group only recalled only three or four pieces correctly. This dramatic result implies that advanced chess players remember pieces in structured positions, and that pieces are remembered as groups or chunks rather than the individual pieces themselves.

DEFINING A CHUNK

- Chunks are learnt constellations
- Chunks are frequently occurring configurations
- Chunks contain elements that are related to each other
- Pieces are related by proximity
- Pieces are related by attacking/defending relationships
- Experts have larger chunk knowledge than the novice

DEFINING A CHUNK

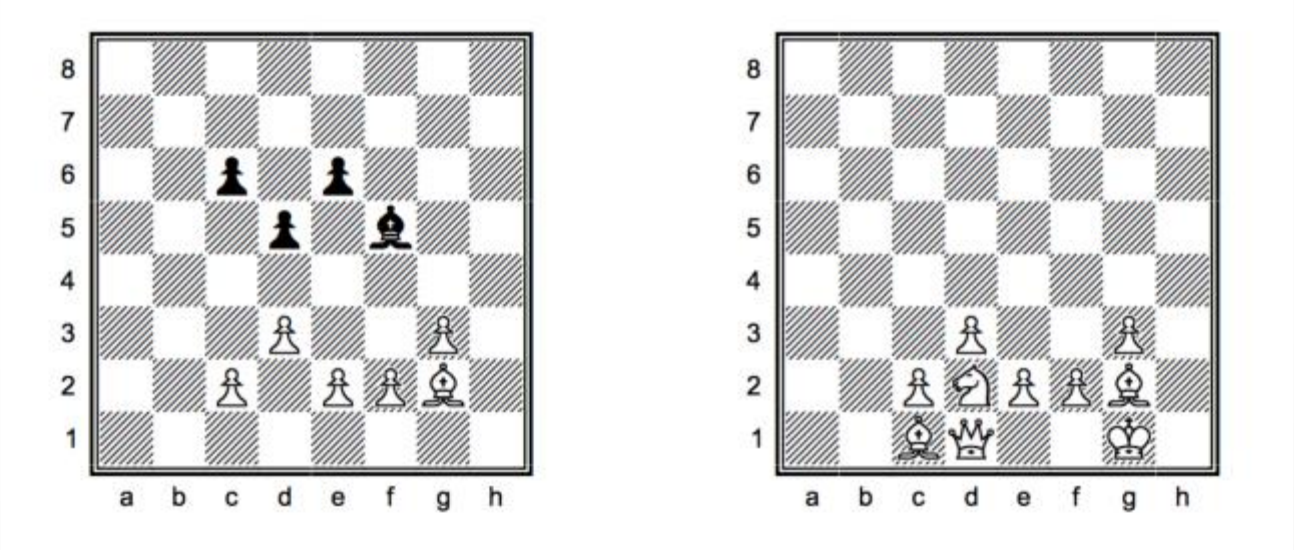
Chunks are absolutely positioned



Shifting pieces from 'a' to 'b' changes winning position from White to Black (from Lane and Gobet, 2010)

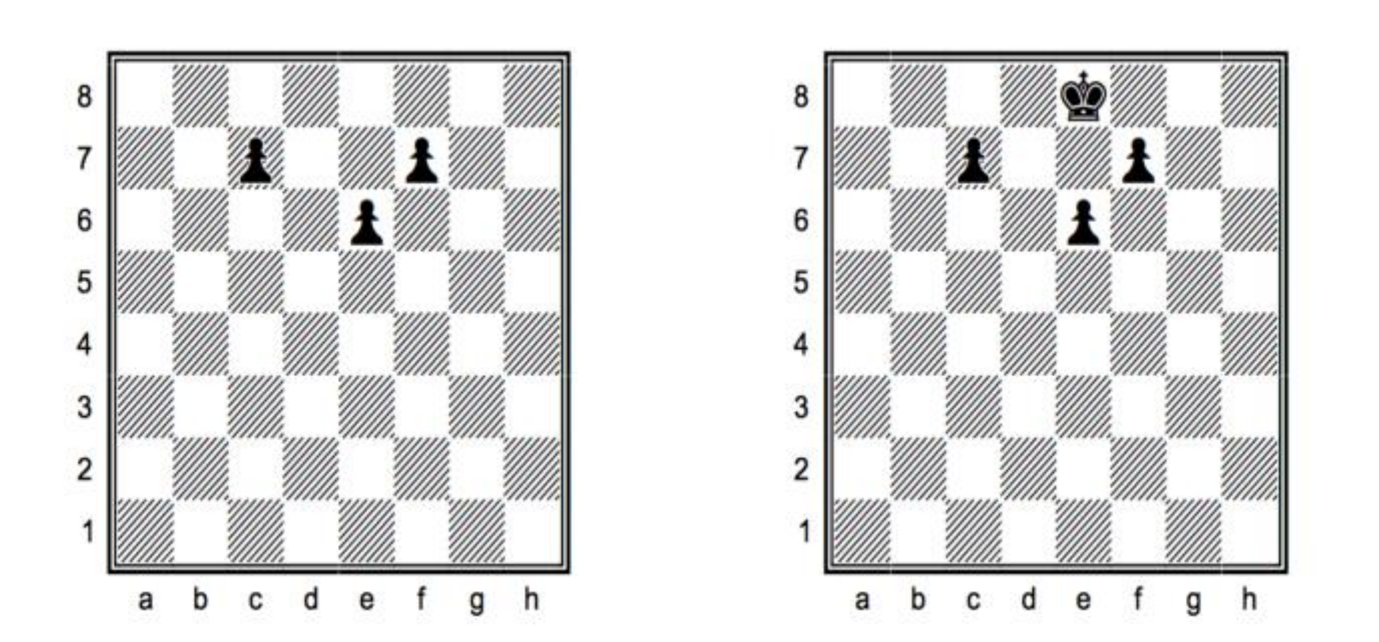
What a chunk looks like

Chunks may be composed of either or both color pieces:



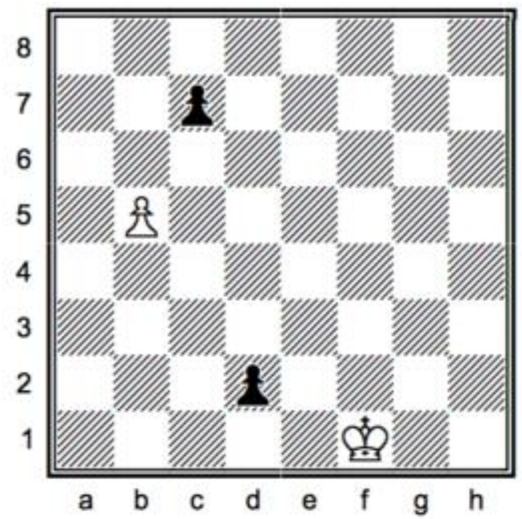
What a chunk looks like

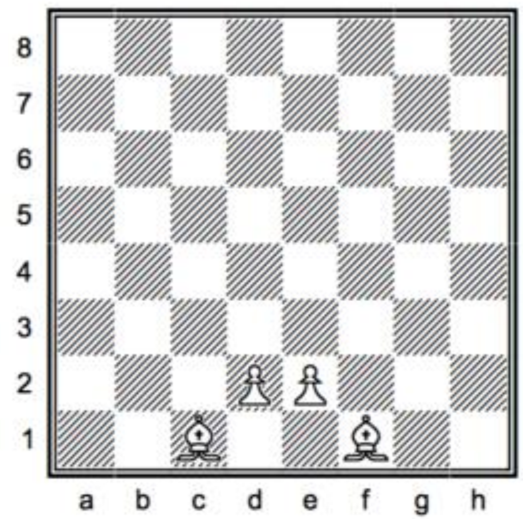
A chunk may be built from smaller chunks:



What a chunk looks like

Pieces in the chunk may be unrelated (below left). A chunk may be part of the initial board layout (below right)



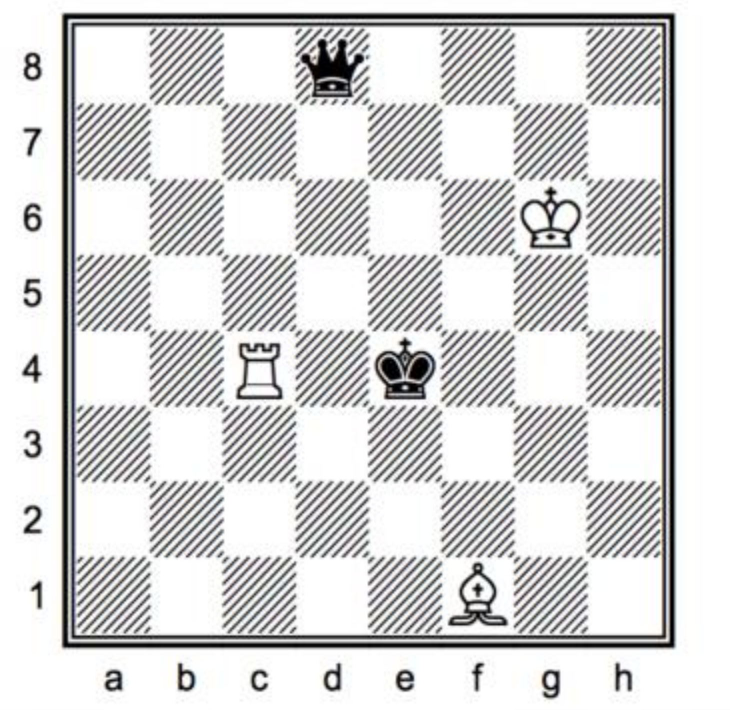


Chunk statistics

From an analysis of chessboards it is clear that there are many patterns or constellations of pieces that occur frequently. The repeated constellations or chunks exist due to the properties of the chess pieces and the rules of the game. It is easy to extract chunks from chess games; the difficulty is finding meaning associated with the chunks. The existence of chunks in itself is not a measure of the player's skill as chunks are found across the whole range of skill sets. Therefore attempts to correlate the player's skill with chunks used are futile, but rather, it is the player's skill that recognizes chunks to assist his chess play.

How many chunks on the board

To illustrate how to count chunks, consider the chessboard figure above, with just five pieces. The chessboard shows five pieces on squares as follows: qd8, ke4, Kg6, Rc4, Bf1. The pieces combine to produce chunks as follows:



How many chunks on the board

A chunk is shown within chevrons and pieces separated by commas. This notation is used de Groot and Gobet (1996). The piece is denoted by the piece name (R=Rook, B=Bishop, N=Knight, Q=Queen, K=King, P=Pawn), followed by the square location on the board. If the piece name is lowercase then the piece color is black, otherwise it is white.

The pieces combine giving a number of chunks increasing as a piece is added. With each piece added, the number of resulting chunks follow the series,

1,3,7,15, 31... This series can be expressed as a formula:

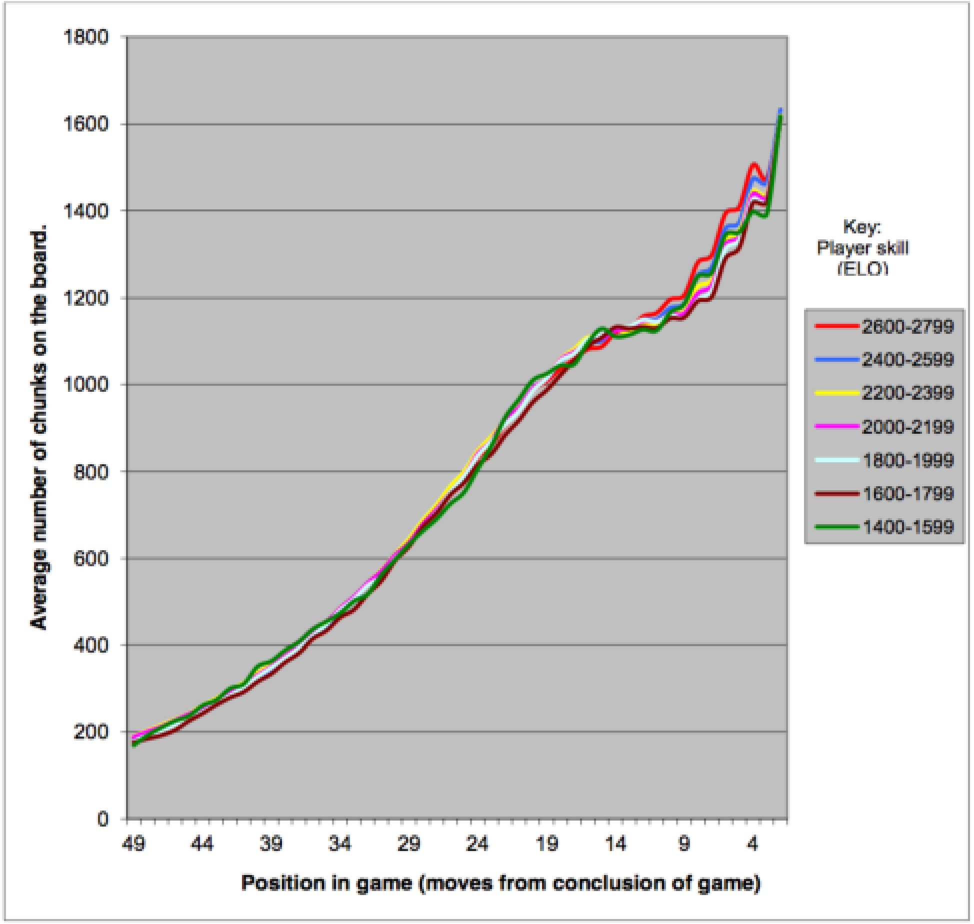
Combinations = $(2^{n}) - 1$ Where „n" is the number of pieces on the chessboard.

How many chunks on the board

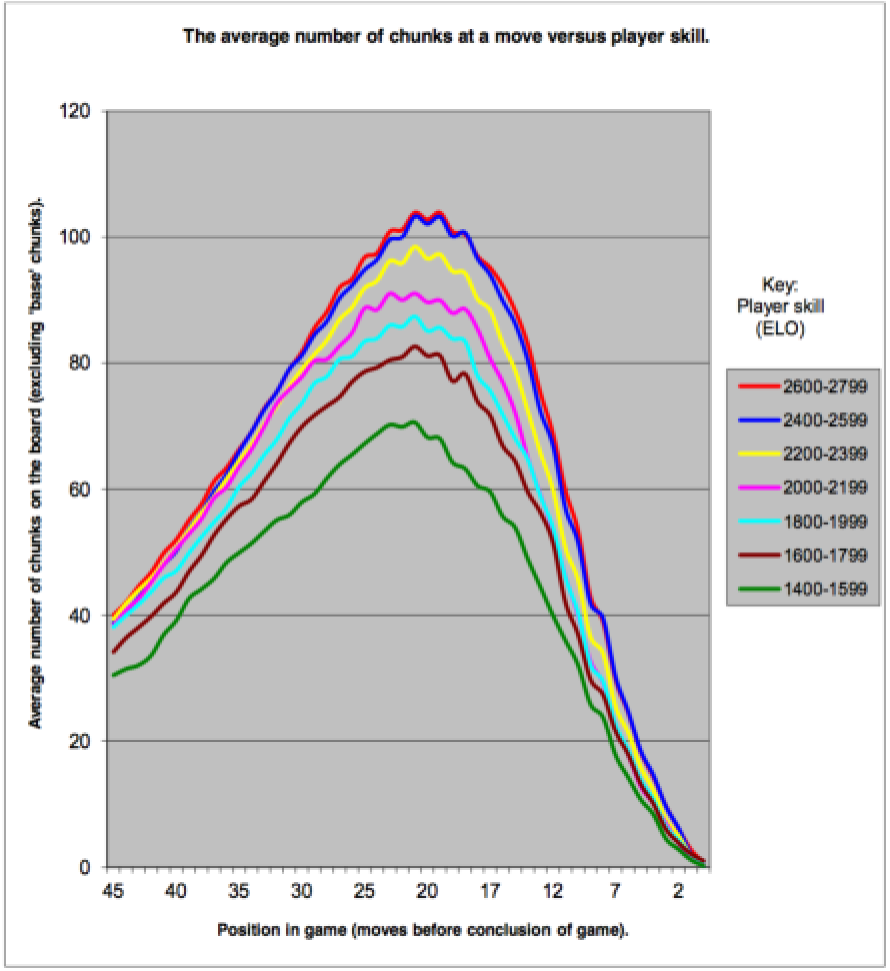
<qd8>
<ke4>
<ke4, qd8>
<Kg6>
<Kg6 ,qd8>
<Kg6, ke4>
<Kg6, ke4, qd8>
<Rc4>
<Rc4, qd8>
<Rc4, ke4>
<Rc4, ke4, qd8>
<Rc4, Kg6>
<Rc4, Kg6 ,qd8>
<Rc4, Kg6, ke4>
<Rc4, Kg6, ke4, qd8>
<Bf1>
<Bf1, qd8>
<Bf1, ke4>
<Bf1, ke4, qd8>
<Bf1, Kg6>
<Bf1, Kg6 ,qd8>
<Bf1, Kg6, ke4>
<Bf1, Kg6, ke4, qd8>
<Bf1, Rc4>
<Bf1, Rc4, qd8>
<Bf1, Rc4, ke4>
<Bf1, Rc4, <ke4, qd8>
<Bf1, Rc4, Kg6>
<Bf1, Rc4, Kg6 ,qd8>
<Bf1, Rc4, Kg6, ke4>
<Bf1, Rc4, Kg6, ke4, qd8>

Chunks statistic

The average number of chunks found on the chessboard



Chunks statistic

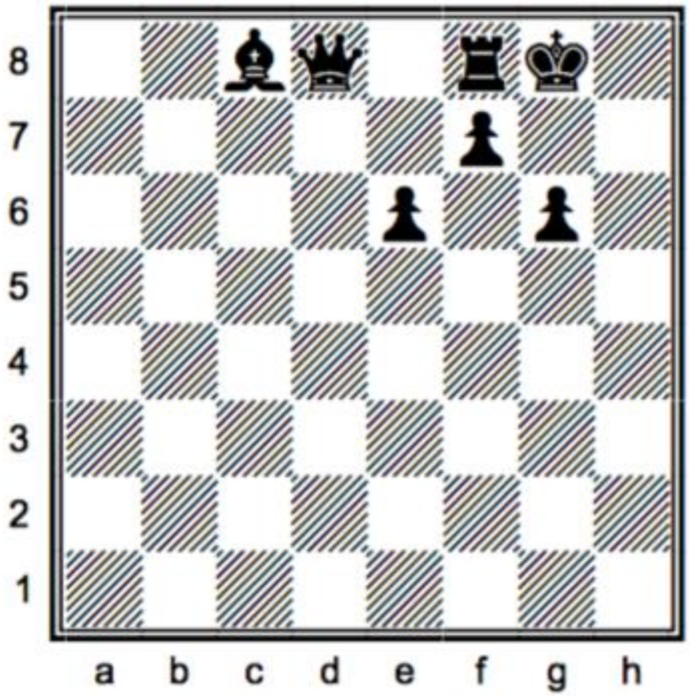


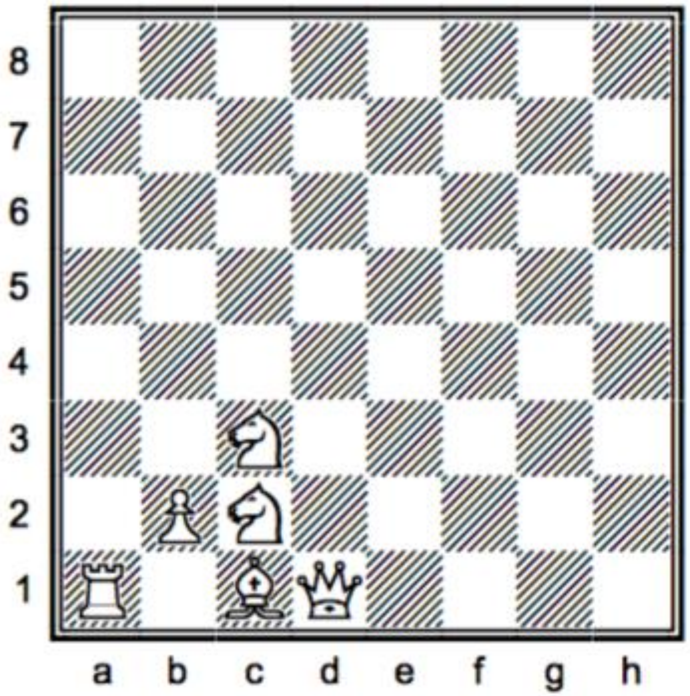
The average number of chunks found on the chessboard excluding 'base chunks'

Defensive Chunks

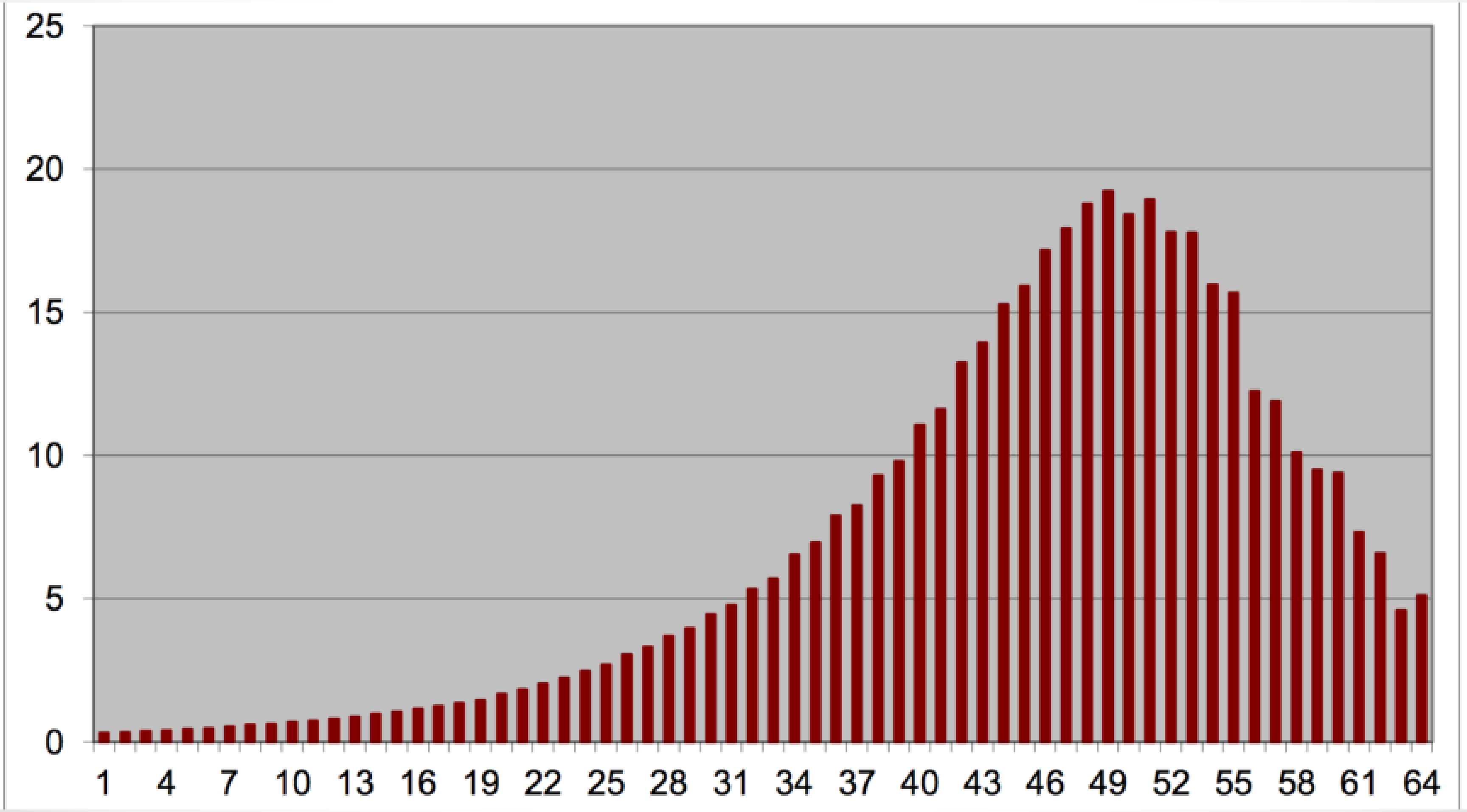
Using the chunk list generated by CHREST (Chunk Hierarchy and REtrieval STructures) program and removing all of the pieces, except those pieces that defend each other within the chunk, the number of chunks reduces to 2,504. These chunks („defensive chunks”) have each piece protecting another piece within the same chunk and in this way the group of pieces making up the chunk have an intrinsic value. Chunks in this case are therefore only composed of pieces of the same colour, yielding 972 white and 1,532 black chunks. These patterns are only chunks where the pieces defend each other.

Examples of 'defensive' chunks





The occurrence of defensive chunks throughout a game



The persistence of defensive chunks with player skill

Skill Elo Range	Average Chunks throughout the game	Games Played	Number of Boards evaluated
1200 – 1399	0.323787	55	22836
1400 – 1599	0.284048	191	88890
1600 – 1799	0.295222	343	150978
1800 – 1999	0.276351	1134	539136
2000 – 2199	0.278841	8714	4113450
2200 – 2399	0.276492	18402	8827488
2400 – 2599	0.279601	18704	8795946
2600 – 2799	0.274532	4689	2290836

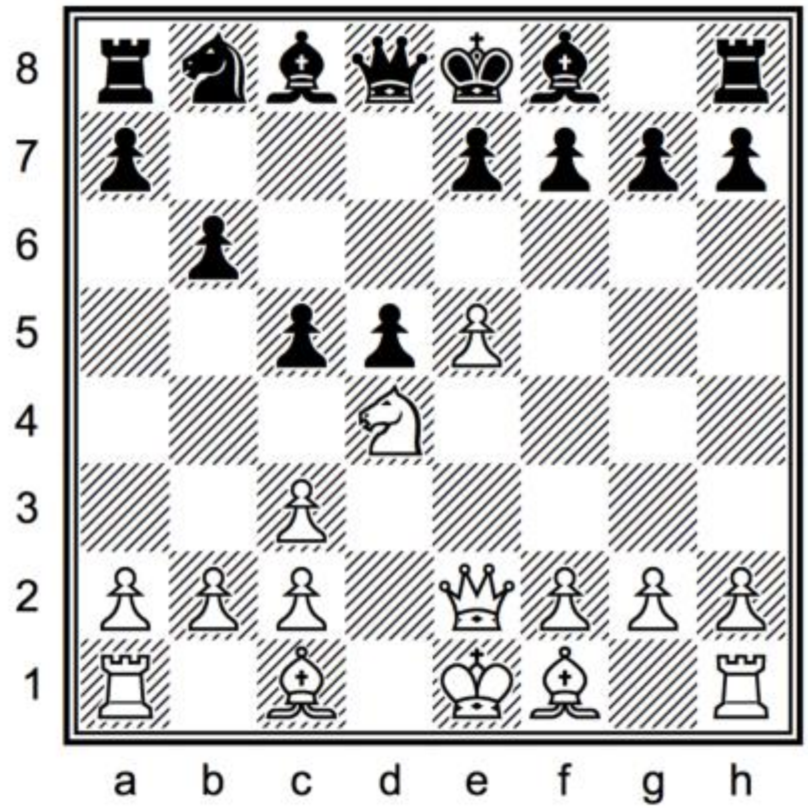
The results of the analysis comparing the persistence of defensive chunks shows no significant differences between skill groups. It was therefore not considered necessary to perform any further statistical analysis on these data.

CLAMP (Chunk Learning and Move Prompting)

CLAMP program (created by Andrew Cook) analyses Master and Grandmaster games, building a library of frequently occurring patterns (chunks) that precede moves of chess pieces. When presented with a new board configuration CLAMP uses its ‘chunk knowledge’ to suggest which pieces are most likely to be selected to play the best move. CLAMP does not use forward searching or minimax techniques but from pattern matching alone, in the mid-game, can appropriately select the most likely pieces to move with a statistically high probability.

The Bratko/Kopec tests

Test 4 (best move: pawn lever)



Results of Bratko/Kopec test

From	To	Score	F	From	To	Score	F	From	To	Score	F
Bc1	Bg5	474395		Pg2	Pg3	264308		Qe2	Qg4	152290	
Bc1	Bd2	434346		Qe2	Qd2	254985		Pb2	Pb4	147631	
Pa2	Pa3	398470		Qe2	Qh5	247290	5	Ra1	Rb1	146494	
Bc1	Be3	369474		Ke1	Kd1	244264		Pf2	Pf4	145460	
Bc1	Bf4	367386		Bc1	Bh6	212982		Ph2	Ph4	121797	
Nd4	Nf3	364672	2	Nd4	Nc6	205980		Qe2	Qd1	117279	
Ph2	Ph3	309563		Pf2	Pf3	205363		Rh1	Rg1	104693	
Pb2	Pb3	304146		Nd4	Nf5	204345		Qe2	Qa6	101879	
Pa2	Pa4	298771		Qe2	Qd3	201949		Qe2	Qe3	96046	
Pc3	Pc4	297073		Qe2	Qb5	195325		Ke1	Kd2	78482	
Nd4	Nb3	291897	3	Qe2	Qc4	177933		Pg2	Pg4	76638	
Nd4	Nb5	284993	4	Nd4	Ne6	164475		Pe5	Pe6	70692	<
Qe2	Qf3	277261		Qe2	Qe4	164306					

The top five CLAMP scores

Shows the top five CLAMP scores with the position of the move in order of preference from an analysis by the Fritz chess engine, with "1" being the best move and "38" being the worst move.

From	*To*	**Fritz move preference**
Bc1	Bg5	8
Bc1	Bd2	13
Pa2	Pa3	19
Bc1	Be3	11
Bc1	Bf4	12

Basic chess combinations

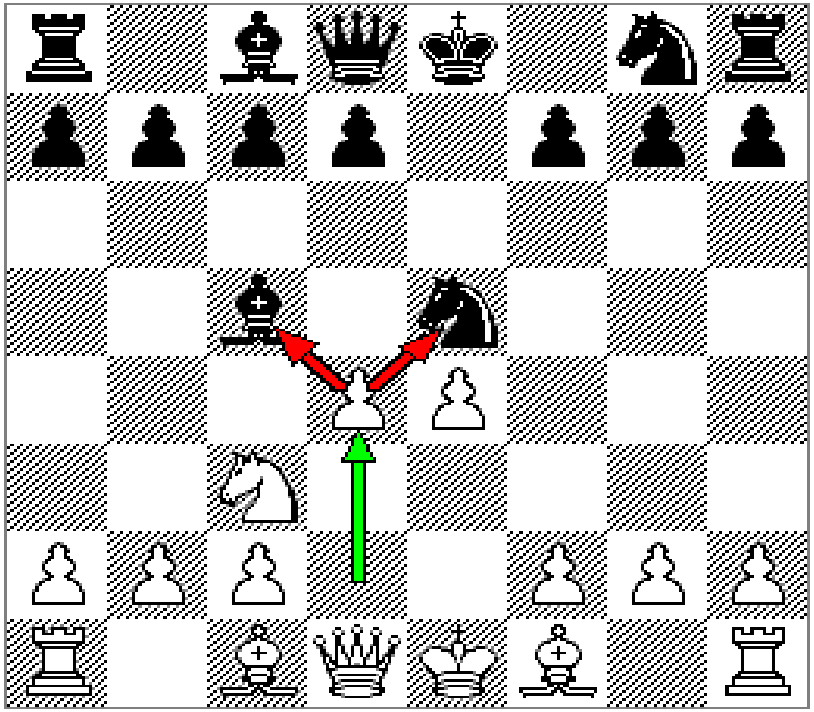
In my research I go from different side and develop statistic knowledge about basic combinations which people learn from the begging when they start play chess:

- fork;
- double attack;
- skewers (in this work I'll not make difference between absolute and relative, pin and skewer).

Its important to know how often players use these small and simple chunks for building their game strategy and complicated combinations. The goal is to find out how players recognize opportunities to add right combination in the game situation, on which sides and directions they focus attention.

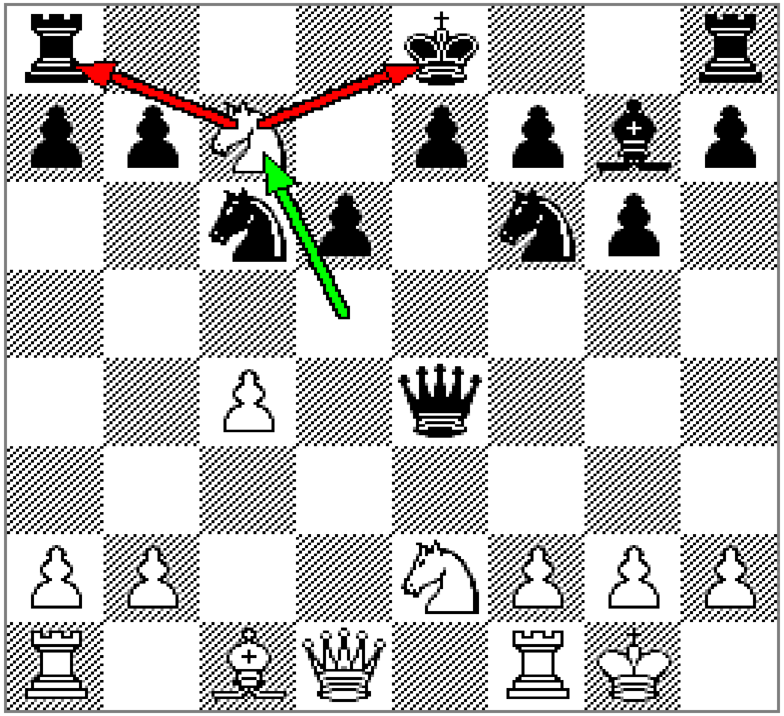
Fork

This combination appear when pawn attacks two of opponent’s pieces at the same time.



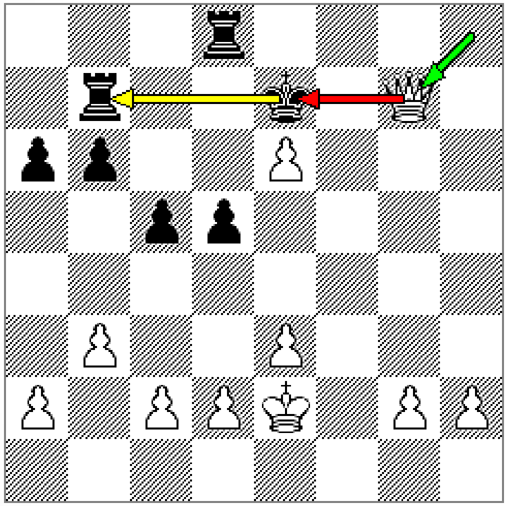
Double attack

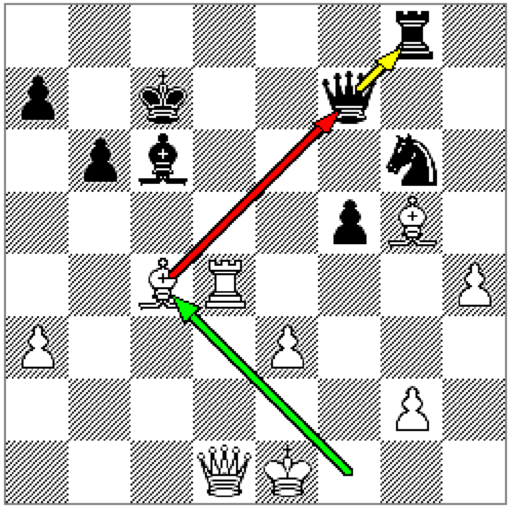
Similar to Fork, just instead pawn here we have other kind of pieces:



Skewers

A skewer is a move which attacks two pieces in a line.





As data source I used database of chess games from FICS (Free Internet Server) for 2008 year. During analysis were processed 300000 games within moves between 20 from the start of game and 20 before end (suppose its middle game part). Results shown in next table:

N games	N moves	Winner	White F	Black F	White DA	Black DA	W Skewers	B Skewers
148714	4615841	White	45647	40088	2327912	2274124	902777	826671
139028	4555252	Black	42954	40848	2258209	2291212	857291	852570
12258	860841	Draw	7636	6708	444955	442262	227369	221103

In percent

N games	N moves	Winner	% White F	% Black F	% White DA	% Black DA	% White S	% Black S
148714	4615841	White	0.989	0.868	50.433	49.268	19.558	17.909
139028	4555252	Black	0.943	0.897	49.574	50.298	18.820	18.716
12258	860841	Draw	0.887	0.779	51.688	51.376	26.412	25.685

Where

F – fork
DA – double attack
S – skewer

From provided results we can see that Whites statistically won more often then Blacks (49.57% to 46.34%) and just in 4.09% of cases there are draws. Percentage of forks appearing doesn't show significant information just tell that Whites can make it more often (possible more simple) then Blacks. Skewers as well as forks don't show so much correlation with won side. Most interested is Double Attack combination, its appear in 50.433% Whites to 49.268% of Blacks moves when Whites won. In the games where Blacks won Blacks have higher percentage of Double attacks compare with Whites. This can lead us to simple conclusion – side which will have double attacks in most of the moves has more chance to win, it has sense in case if we merge this with chosen strategy.

For future research I am planning to split middle game by number of pieces (I guess it has correlation with percentage of chunks), add new combinations (split skewers to absolute, relative and distinct with pin combinations) and make tests to check efficient of suggestions, which can make based on this analysis, compare with Fritz chess engine.