11th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN), Nis, 3-6 June 2024

Axon Chess Engine Evaluation Function

Vladan Vučković
Computer Department
Faculty of Electronic Engineering, University of Niš
Niš, Serbia
vladanvuckovic24@gmail.com (https://orcid.org/0000-0002-7552-9596).

***Abstract*— This paper presents the details of the evaluation function, which is an integral part of the latest version of the author's chess application Axon (version 2024). The evaluation function is the most complex part of the chess program, developed and tested over a long period of time. The heuristics, which are explained and discussed in detail in the paper, represent an excellent basis for the development of chess program evaluators of grandmaster strength.**

Keywords*— Theory of Logic Games, Computer Chess, Chess Engines, Evaluation Function.

I. Introduction

The evaluation function is one of the most important procedures in a chess program [1],[2],[3],[4]. In the literature, this function is often called the heuristic evaluation function, because it defines the heuristic or expert chess knowledge that the program possesses [1]. The evaluation function is responsible for the game style of the program; whether it is a predominantly combinatorial, positional or combined style of play [6],[12].

By definition, an evaluation function is a mathematical additive function that summarizes the material and positional aspects of the current chess position created in the game tree and assign it a numerical value [5],[7],[12].

The position value can be defined in different formats [13],[15]. In a large number of programs, including the author's applications, the 16-bit evaluation format is adopted, so that the numerical values of the position range from -32768 to +32767. In doing so, extreme values (greater than 30000 or less than 30000) are used to evaluate mats [5],[7],[11].

When calculating the tree, the algorithm at some point reaches the maximum depth that was determined at the beginning of the search procedure. The positions that are evaluated at that depth are usually unstable, that is, in a large percentage of cases there are tactical threats from both sides. Attempting to evaluate tactically unstable positions is unproductive, as has been shown in numerous works by researchers. In all modern programs, at the level of terminal nodes, a special procedure (*quiescence search*) is started, which has the task of stabilizing the position through the processing of tactically active moves - exchange of pieces, some chess, promotions.. to the level where the evaluation function can be applied with the highest possible reliability [1],[8],[12]. Of course, depending on the quality of the quiescence procedure and the evaluation function, the end nodes are processed with a certain level of accuracy [16]. The mentioned two procedures are also responsible for minimizing the harmful impact of the so-called *horizon effect* [1],[11]. It is extremely important to set the optimality and relationship between the nodes that are processed at the level of the root tree and at the level of the quiescence procedure. In tactically complicated positions, the ratio of the number of nodes generated at the level of the quiescence procedure and the main part of the tree is 90:10, while in tactically calm positions the ratio is 50:50. It has been shown that using overly complex quiescence procedures is counterproductive [16]; programs cannot reach significant primary depth.

The *notion of granularity* of the evaluation function is also significant. The granularity or *resolution* of the evaluation determines the minimum difference that can occur between two positions. For the evaluation range shown, that difference is 1. Considering the functioning mechanism of the tree processing procedure, which ends with terminal nodes where the evaluation function is applied, a higher resolution is more suitable because it enables a greater number of ALPHA-BETA cuts in the tree. On the other hand, the resolution must be sufficient to perform the gradation of similar positions, which is very important in the strategic aspects of the chess game, especially when it comes to closed, non-tactical positions. There are no clear rules and conclusions in this part, most often the aspects of the evaluation function are adjusted afterwards, since the entire chess application has been developed. With leading programs, which have dozens of positional parameters, the evaluation setup can take years.

Unlike top master chess, where the knowledge of position evaluation is acquired over the years, and where this knowledge is subject to change and improvement, in computer chess the evaluation function is essentially static in nature. Once defined, it is applied in an immutable form in the thousands and millions of positions that can occur during the calculation of the tree. In some programs there are options for changing the "style" of the computer game - from an aggressive and attacking style to an extremely cautious and positional one. When examining parties that play the same program under the same conditions, it can almost be concluded that they are different programs. However, it is about changing only a few parameters in the evaluation function [9],[10].

Based on all of the above, it is completely clear the importance of evaluation and how much attention must be paid to the study and implementation of this function.

This paper was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia [grant number 451-03-65/2024-03/200102].

II. Symmetric Evaluation Function

The term *symmetric evaluation function* refers to the complete symmetry of the calculation of the material and positional aspects of pieces regardless of the color, the side that is on the move or the state of the accompanying features of the position. Also, the evaluation must be horizontally positional symmetrical. Let's assume the evaluation is +2.00 for white and then change the color of all the pieces so that the black pieces change to white and vice versa. The resulting evaluation must be -2.00. If we change the position of the pieces on the board using horizontal symmetry so that the pieces from the fields A1 and H1, A2 and H2 ... swap places, the evaluation must be identical to the initial value. In practice, it has been shown that the use of a symmetrical evaluation function leads to a maximally stable game in strategic and tactical terms. The use of asymmetric evaluation functions, which may be the result of errors in the programming of the function itself, leads to instability in the calculation and reduces the quality of the game.

A. Material Evaluation

The basis of the evaluation of each position is the material evaluation. In most cases, in the positions generated during the tree calculation, the material parameters are sufficient for the final assessment of the position. If this detail is considered in connection with the quiescence procedure, it is shown that the material calculation is almost dominant in the realization of a high-quality and reliable procedure for the evaluation of the position.

If the sixteen-bit evaluation range is adopted, when constructing the evaluation function, it is necessary to determine the absolute material values of individual pieces as well as their mutual relations. The following table I shows the material values of all pieces, assuming that the piece with the least weight, the pawn, is evaluated with a value of +100.

TABLE I
The table shows certain evaluation values for some applications.

Piece	Chess 4.5 engine	Modern chess engine	Axon 2024
Pawn	100	100	100
Knight	325	300	300
Bishop	350	325	300
Rook	500	500	500
Queen	900	900	900

Since each side has only one king, their values are not calculated. As shown in Table 4.8, the evaluation values have not changed significantly since the construction of the first chess programs. Actually, the only changes are in the knight and bishop values. The relationship between the pieces that is shown is the result of expert knowledge (heuristics) created by many years of theoretical and practical work of generations of chess masters. This ratio of the material value of the pieces is also confirmed in numerous chess programs. In the author's application, the fact that these values are in the mathematical ratio 1:3:5:9 is used. For more precise corrections of the evaluation of individual pieces, the positional part of the evaluation function is in charge.

If a positive sign is adopted for the white pieces, the black pieces will have identical values, but with a negative sign. The simple formula, which calculates the material balance of the entire position, can be represented by the following mathematical function:

$$\mathbf{M_{Eval} = SUM\ (W_i\text{-}B_i)\ V_i} \quad \text{(i is the index of all pieces),}$$

therefore, it is a simple sum of tabular values of white and black pieces. Black has negative sign.

In the initial chess position, when the number of pieces is the same for white and black, the material evaluation is 0. If the evaluation has a positive value, white has a material advantage, and if the evaluation is negative, black has an advantage. The initial material value of the pieces on one side is 8*100+2*300+2*300+2*500+900 = 3900, so approximately 40 pawns. The presented basic material evaluation is often supplemented by an evaluation of advantages.

Specifically, if one side has a material advantage over the other, the rational strategy is for that side to exchange pieces and not pawns, in order to simplify the realization of the advantage as much as possible. It is known from chess theory that the realization of a material advantage is done most simply if there are only pawns on the board. On the other hand, if a side is weaker, in material deficit, a good strategy is to exchange pawns and not pieces. In this way, with few pawns on the board, realizing the advantage of the stronger side is increasingly difficult, so that the game approaches a draw. One side can be in advantage for the whole piece (+3.00), but if there are no pawns it is theoretically a draw, e.g. king and bishop or knight against the king. In this sense, depending on the program, the expansion of the existing evaluation function is introduced.

B. Material Evaluation in the Endgame

There are positions in the final where the material evaluation has no significance. In the event that one side has a material advantage, but the piece is not checkmate (knight or bishop), the position is not evaluated for the given material, but is assigned a value of 0 - a draw. The following table shows characteristic combinations with 2, 3 or 4 pieces that must be immediately evaluated to 0, wherever they appear in the tree, because they are theoretical draw positions.

This principle of pre-defined draw is later considerably expanded and generalized by using ending bases (*Nalimov, Syzygy*).

TABLE II
THE TABLE SHOWS ALL THE SITUATIONS THAT ARE EVALUATED AT 0, BECAUSE THEY REPRESENT A THEORETICAL DRAW.

White	**Black**
King	King
King+Knight	King
King+Knight	King+Knight
King+Knight	King+Bishop
King+Knight	King+Knight+Knight
King+Bishop	King
King+Bishop	King+Knight
King+Bishop	King+Bishop
King+Bishop	King+Knight+Knight
King+Knight+Knight	King
King+Knight+Knight	King+Knight
King+Knight+Knight	King+Bishop
King+Knight+Knight	King+Knight+Knight

The consideration also applies to symmetrical structures, when the colors of the pieces change. In positions with 2 or three pieces, a draw can be immediately evaluated if the layouts presented in the table appear, and in more complex structures, it is necessary to examine whether the side is in check before the actual evaluation. If even one pawn is present on that board, a draw must not be evaluated due to the possibility of promoting the pawn to checkmate. The situations shown are simply solved programmatically with a series of conditional jump commands. More complex situations with a small number of pieces are solved by using endgame bases.

C. Contemt Factor

Contemt factor is introduced when it is necessary for the program to avoid repeating moves or draw variants, most often when it is necessary to play to win. Contemt factor means that a draw that occurs by repeating a position through eternal chess is not evaluated with 0.00 but with some small negative value. In this way, the program avoids playing draw unless the evaluation is negative and less than the contempt factor. For example, if the contempt factor is set to -1, the program will play to win even when there are fewer pawns. Otherwise, the program would try to achieve a draw in case it judges its position to be weaker.

III. POSITION EVALUATION

Positional evaluation is a major part of the evaluation function. Its task is to summarize all non-material positional aspects of the current position. In order to maintain the rationality of the game and prevent the program from unnecessarily sacrificing pieces for a checkmate attack that the opponent can defend against, a limitation of the evaluation function is introduced. Depending on the program, these limits are usually set at +- two pawns. This means that if the material is equal, one side can have the biggest advantage +2.00 in a situation where its positional parameters dominate in relation to the opposing side.

The positional parameters that are evaluated can be very different and they depend a lot on the chess knowledge of the programmer. Also, the programs differ greatly not only in the number of evaluation parameters that are introduced, but also in the difficulty of certain aspects of the static analysis. It is important to mention that it must be taken into account that each element of expert knowledge that is introduced via the evaluation function must be defined using a suitable series of conditional machine commands, which are by definition slow. Therefore, it is necessary to find the optimum in terms of the speed of work of the evaluator and the amount of expert knowledge that is programmed. Programs with a minimalist approach *(Fruit engine)* that have a low level of expert knowledge built into the evaluation function can be very effective in terms of achieved chess strength.

In the following, we will present some segments of static evaluation, which are standard for implementation in chess programs. *Weights* - values of individual parameters will not be displayed because they are fine-tuned only afterwards and in function of other segments of the chess program as well as based on results and games against real opponents. The elements of static evaluation can be divided into two groups: *affirmative* ones that reward positive positional aspects that are recognized in the current position and *restrictive* ones that punish positional deficiencies that are registered. Affirmative parameters add specific values to the static evaluation while restrictive moves subtract values.

Some of the basic components of static evaluation are detailed and analyzed below:

A. Piece Mobility

Piece mobility is often calculated by the number of squares a piece controls, assuming they are not simultaneously controlled by an opposing piece. The greater mobility of the pieces allows for a greater number of combinations, which makes it more likely to find a suitable move in defense and attack. Also, the high mobility of one's own pieces means a reduced mobility of the opponent's pieces, which indirectly affects even more the strengthening of one's own position. In this part, it is possible to introduce a restriction for pieces that are blocked and whose mobility is limited by own or opponent's pieces. Evaluation of long-range pieces is crucial in terms of mobility. In particular, it is necessary to pay attention to the evaluation of the hunting pair, which comes to a great expression in open positions, which are inevitably generated after a series of exchange of pieces, in the final stages of the game. Also, in this part it is necessary to detect "masked" pieces. In next position (diagram in Fig. 1.) white knight is excellently placed, but its position must not be rewarded but punished because of the double mask it is in. The mobility of the knight in the shown position is actually reduced to zero and it represents a significant positional minus for the white side.

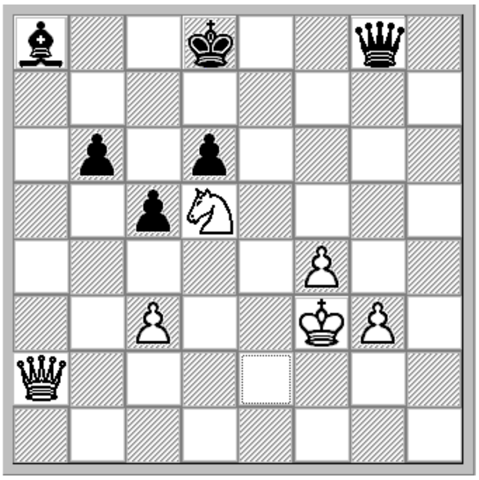

Fig.1. Illustration of evaluation details. The white knight in center is double "masked". White knight has negative positional weight.

B. King Centralization and King Safety

Evaluating the position of the king is a very complex task due to the dual role that the king plays in the center and endgame. In the center, when there are a maximum number of pieces on the board, the king often becomes the target of attacks, so it is strategically correct to cast the king to one of the sides as early as possible in the game. In this way, the king moves away from the center towards the edge of the board, where it is significantly less likely to be attacked by the opponent's pieces, especially long-range pieces. On the other hand, it is easy to form a defensive infantry structure. The evaluation of the security of the king in the center involves rewarding the castling of the king as well as restriction in cases of broken castling - when the king remained in the center of the board or when it was casted and the pawn structure protecting it has serious weaknesses. The environment around the king, the number of the opponent's attacking and own defensive pieces is also significant. If the opponent has a queen and in addition the king does not have adequate protection in the form of pawns or defensive pieces in his vicinity, the value of that position must be reduced. In many checkered endings, the opening of one side's king can be decisive in losing the game.

On the other hand, in the phase of transition from the center to the endgame as well as in the endgame, the centralization and good position of the king is extremely important. Contrary to the situation in the center, the king in the endgame becomes an attacking piece, so the principle of evaluation is completely different. This duality in the evaluation is most simply solved by introducing a material limit above which the evaluation in the center is calculated. Otherwise, the king in the endgame is evaluated. In many programs, that limit is set at +12.00 to +15.00. The strength of the centralized king must be calculated in function of the other present pieces of the opponent. In purely pawn endings, the value of the king centralization parameter is maximum.

C. Piece Position and Centralization

The centralization of pieces is a very important segment of the overall evaluation in the middle and end. Centralized pieces very easily act on all segments of the board, control the largest number of squares and develop their maximum strength. That's why centralized pieces are rewarded with a corresponding bonus regardless of the position of other opponent's or own pieces. Pieces that are poorly positioned and located on the edges of the board reduce the evaluation. The technique of determining the bonus for the centralization of pieces is most often implemented through the corresponding centralization matrices [12],[13],[15].

Each square on the chessboard is assigned a corresponding weight. Since each of the pieces has its own specificities, a centralization matrix is defined separately for all types of pieces. For pieces of the opposite color, vertically symmetric centralization matrices are generated. An example of one such matrix for a white knight is given in the following table:

TABLE III
TABLE OF THE CORRESPONDING MATRIX FOR THE WHITE KNIGHT.

-16	-8	-4	0	0	-4	-8	-16
8	16	32	32	32	32	16	8
24	40	56	62	62	56	40	24
32	48	60	64	64	60	48	32
16	32	56	60	60	56	32	16
-8	8	24	20	20	24	8	-8
-20	-16	-8	16	16	-8	-16	-20
-48	-32	-24	-16	-16	-24	-32	-48

Bonuses or restrictions for individual fields are obtained by direct access to a certain element of this matrix. *The central fields in the matrix have the highest values, while the edge fields have negative ones values.* The average values of the matrix elements increase when the figure is in the opponent's part of the board, thus forcing the program to strive for an active - attacking game. The highest positive value is when the knight is on fields D5 or E5 (+64) and the smallest when it is in the corner fields on its half of the board: fields A1 and H1 (-48).

Of course, depending on the basic values in the main part of the evaluation function, these local values in the matrix can be normalized. When the corresponding matrices are constructed, following the aforementioned principle of symmetry of the evaluation function, the corresponding matrix for the opposite color is simply generated.

During the computation of the evaluation, the values for black are negated. Symmetric matrices of this type enable quick and simple calculation of several positional aspects, primarily the static activity or inactivity of the figure as well as the development and influence on other segments of the board.

D. Piece Development

In the starting position on the board, all pieces are in their basic positions. In chess theory, there are certain heuristic rules that must be followed in the opening to ensure a harmonious and harmonious development of the pieces. The parts of the evaluation function that handle this segment of the chess game are based on the use of a certain system of rules. The basic rule is that pieces that are behind on the baseline reduce the evaluation. In this part, special attention is paid to the development of light pieces, knights and bishops. This simple mechanism makes it possible to force the development of the pieces in the early stages of the chess game, which indirectly opens the fields for the castling of the king. The un-casted king, which is left behind in the center of the board, represents a big positional minus and becomes the object of attack, so positive points in the evaluation must be generated for the castling of the king. A good evaluation of this segment follows the basic opening plan:

- Development of centralization pawns,
- Development of light pieces (bishops and knights),
- The castle of the king,
- Placement of rooks on strategically important verticals,
- Queen development.

With a good combination of positional evaluation parameters, this strategic plan is achieved in most professional chess programs. Well-developed pieces in the context of a certain opening system provide opportunities for a good continuation of the game in the center. The Axon 2024 program has shown in practice the decisive influence of the evaluation function on the quality of the opening play.

IV. Pawn Structure Evaluation

The evaluation of the pawn structure is extremely important for a precise and high-quality evaluation of the position. The importance of the pawn structure increases as the number of pieces on the board decreases. As is known from the chess rules, a pawn has the possibility to be promoted to a queen, rook, bishop or knight by reaching the last row, which provides an opportunity for a reversal in the material evaluation and assessment of the position. The pawn structure is often evaluated in conjunction with some positional aspects that are adopted based on the expert knowledge and experience of chess masters. The elements related to this part of the evaluation are:

Promotional pawns – Pawns whose progress to the starting square cannot be prevented or blocked by the opponent's pawns are called promotion pawns and their presence is a positional plus for the side that owns them. The power of promotional pawns grows exponentially as the distance from the starting field decreases. If there are promotion pawns on adjacent columns (connected promotion pawns) their strength is exceptional, in some cases even the opponent's rook is not able to stop the promotion of one of them, provided of course that the opponent's king is far enough away. As an illustration of these statements, we will present an example of the evaluation of connected pawns on the sixth row. The position given in Fig. 2., where Black is on the move, is obtained for White, which is easy to show by analysis at a depth greater than 8-9 half moves:

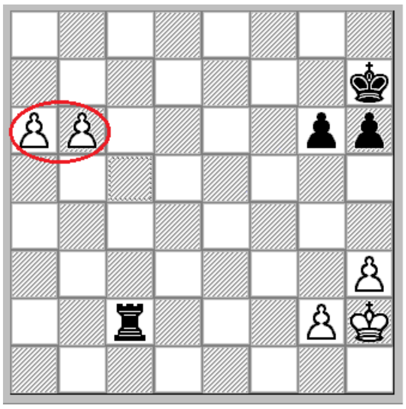

Fig.2. Illustration of the power of linked promotional pawns. It is winning position for white.

The precise valuation of promotional pawns is of extreme importance for the correct evaluation of the position, because their presence in some cases completely changes the value of the position. In the previous example (Fig. 2.) the material evaluation shows that Black is ahead with +3.00 but as the dynamic analysis in Table 4.12 showed, Black is actually losing with -4.00. The difference between material and dynamic, i.e. positional evaluation in this case amounts to 7 pawns. Problems of a similar type occur when treating other aspects of the evaluation function, so solving them with a combination of static and dynamic evaluation is of utmost importance for building a top-class chess program.

Evaluation of duplicated and tripled pawns – In the event that two pawns of the same side are located on the same vertical, a defect in the pawn structure is created that reduces the value of the evaluation.

Isolated pawns - By definition, isolated pawns are those pawns that do not have pawns of their color on adjacent verticals.

Pawn Islands - Pawn Islands are groups of pawns separated by verticals that do not contain pawns of the same color. A larger number of pawn islands is a positional disadvantage for the side that owns them.

V. Evaluation Function in Searcher Algorithm

The following listing shows the basic procedure that implements the minimax principle (Pseudo Pascal). At the same time, this is the core from which every chess program is developed by gradually adding functionality. *Position of the evaluation function call inside this procedure is bolded:*

```
procedure NegaMax (Position, Depth): integer;
var …
 { Position – Current position }
 { Depth – Search tree depth }

begin
   if (depth=0) then
     begin
        Evaluate(Position);    { << Evaluation function call }
        Exit;
    end;
   best: = -INFINITY;            { Initial value }
   succ: = Successors(Position);   { Find successor }
   while not Empty (succ) do       { All legal moves loop }
   begin
        Position: = RemoveOne(succ);        {Remove one move }
        Value: = -NegaMax(Position, depth-1);   { Recursive call }
        if  (value > best) then best: = value;       { New best value }
   end;
NegaMax:=best;   { Return value that is minimum or maximum }
end;
```

The details of the specific application are based on one variant that is often called *NegaMax*. The central point in this function is call of *Evaluator function*.

The input arguments for the function call represent the position and depth to which the calculation needs to be performed. The *Successors* procedure is tasked with generating at each level a list of moves that represent the outputs from a particular node in the branch. The recursive call, which is marked separately, generates the next level with the depth reduced by 1 and the new position, which is obtained by playing a move in the corresponding branch. In this way, the procedure recursively generates levels of decreasing depth until the process eventually ends at depth 0, when the evaluation process is performed. In this way, the problem of calculating the value of the position at the depth d is reduced to solving a series of simpler problems that have a depth of d-1. The number of positions that are recursively defined in this way grows exponentially, but since all branches are closed with terminal nodes, the calculation time is finite, regardless of the initial position and depth [14]. The evaluation function is called in each terminal node, and it closes each calculation branch in the decision tree.

VI. Conclusion

The evaluation function is a mathematical additive function consisting of a material and a positional component.

The *material component* summarizes the values of white and black pieces, whereby white pieces are counted with a positive sign and black pieces with a negative sign. The ratio of the values of the pieces in Axon 2024 engine is 1:3:3:5:9.

The *positional component* calculates intangible evaluation parameters based on heuristics developed and accepted by chess experts. The number and weight of individual segments is usually determined by the programmer himself, based on tests and longer adjustments during work with the program.

With the addition of more rules in the additive function, the knowledge of the program increases, but the speed of the program decreases. *The optimal balance between the speed of work and the amount of knowledge is one of the key points in creating a good chess program and for Artificial Intelligence in general.* Finding an even better balance is a task for future research.

Acknowledgment

This study was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia.

References

[1] Slate D. J., and Atkin. L. R., “CHESS 4. 5 – The Northwestern University Chess Program”, *Chess Skill in Man and Machine* (ed. P. W. Frey),pp. 82-118. Springer-Verlag,New York,N. Y. 2nd ed. 1983. ISBN 0-387-90815-3. , 1977.

[2] Althöfer I., “An Additive Evaluation Function in Chess”, *ICCA Journal*, Vol. 14, No. 3, pp. 137-141. ISSN 0920-234X, 1991.

[3] Althöfer, I., “On Telescoping Linear Evaluation Functions”, *ICCA Journal*, Vol. 16, No. 2, pp. 91-94. ISSN 0920-234X, 1993.

[4] Althöfer I., and M. Buro, “Techniques for the Evaluation of Game Positions Using Examples”, *ICCA Journal*, Vol. 18, No. 1, p. 34. ISSN 0920-234X, 1995.

[5] Anantharaman T.S., “Evaluation Tuning for Computer Chess: Linear Discriminant Methods”, *ICCA Journal*, Vol. 20, No. 4, pp. 224-242. ISSN 0920-234X, 1997.

[6] Beal D.F,**“**Mixing Heuristic and Perfect Evaluations: Nested Minimaks”, *ICCA Journal*, Vol. 7, No. 1, pp. 10-15. ISSN 0920-234X, 1984.

[7] Christensen J., and Korf R. A, “Unified Theory of Heuristic Evaluation Functions and its Application to Learning”, *Proc. AAAI-86,* 1986.

[8] Frey P.W., “An Empirical Technique for Developing Evaluation Functions”, *ICCA Journal*, Vol. 8, No. 1, pp. 17-22. ISSN 0920-234X, 1985**.**

[9] Lee K.-F, and Mahajan S. A, “Pattern Classification Approach to Evaluation Function Learning”, *Artificial Intelligence*, Vol. 36, pp. 1-25. ISSN 0004-3702, 1988.

[10]Tunstall-Pedoe W., “Genetic Algorithms Optimizing Evaluation Functions”, *ICCA Journal*, Vol. 14, No. 3, pp. 119-128. ISSN 0920-234X, 1991.

[11]Vučković V., and Vidanović Đ., "An Algorithm for the Detection of Move Repetition Without the use of Hash-Keys",*Yugoslav Journal of Operations Research (YUJOR*),Volume 17,Number 2,pp. 257- 274. Belgrade,Serbia. ISSN 0354-0243. , 2007.

[12]Vučković V., “The Theoretical and Practical Application of the Advanced Chess Algorithms”,*PhD Theses*,The Faculty of Electronic Engineering,The University of Nis,Serbia, 2006.

[13]Vučković V., ,"The Compact Chessboard Representation",ICGA Journal,Volume31,Number 3,Tilburg,The Netherlands, ISSN 1389-6911. pp. 157- 164., 2008.

[14]Šolak R., and Vučković V., ,"Time Management During a Chess Game",*ICGA Journal*,Volume 32,Number 4,Tilburg, The Netherlands, ISSN 1389-6911. pp. 206- 220., 2010.

[15]Vladan Vučković, "An Alternative Efficient Chessboard Representation Based On 4-Bit Piece Coding", Yugoslav Journal of Operations Research (YUJOR), Volume 22, Number 2, pp. 257-274.,FON,Belgrade, Serbia, 2012, (DOI: 10.2298/YJOR081028011V) (ISSN 0354-0243).

[16]V. Vučković, “Modified Quiescence Procedure in Axon Chess Engine” Proc. of 50th Int. Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST 2015), Faculty of Telecommunications, Sofia, 24.06.2015-26.06.2015, pp.332-335, Bulgaria,2015. (ISSN 978-619-167-182-3)