TITLE: A Principle-Based Chess Playing Program

AUTHOR: Santos Gerardo Lázzeri Menéndez[1]

[1] Universidad de las Américas, Puebla,
Departamento de Ingeniería en Sistemas Computacionales
Ex-hacienda de Santa Catarina Mártir s/n
San Andrés Cholula, Puebla 72820
México
TEL: (52)-222-2292625, (52)-222-2292029
glazzeri@mail.udlap.mx
http://www.udlap.mx/~glazzeri

Summary: Most chess playing programs use search-based algorithms that do not use the kind of high level strategies used by human chess players. This has been the main approach since the early days of computer chess, when it was discovered that it was easier to increase the speed of the hardware and basic searching algorithms, than to encode chess knowledge correctly. However, it is possible to develop a chess playing program using only strategic principles such as those used by human players. This paper outlines the foundations for the development of such an algorithm, which relies heavily on artificial intelligence techniques such as case-based reasoning. Besides outlining a general solution for each phase of the game, the paper focuses on the implementation and evaluation of a case-based approach for playing the middle game in chess. Furthermore, possible extensions and generalizations of the ideas presented in the paper are discussed, as well as other potential application domains.

Keywords: Case-Based Reasoning, Computer Chess, Fuzzy Logic

Topics: Reasoning Models, Case-Based Reasoning.

Section: Paper Track

A Principle-Based Chess Playing Program

Santos Gerardo Lázzeri Menéndez[1]

[1] Universidad de las Américas, Puebla,
Departamento de Ingeniería en Sistemas Computacionales
Ex-hacienda de Santa Catarina Mártir s/n
San Andrés Cholula, Puebla 72820
México
glazzeri@mail.udlap.mx
http://www.udlap.mx/~glazzeri

Abstract. Most chess playing programs use search-based algorithms that do not use the kind of high level strategies used by human chess players. This has been the main approach since the early days of computer chess, when it was discovered that it was easier to increase the speed of the hardware and basic searching algorithms, than to encode chess knowledge correctly. However, it is possible to develop a chess playing program using only strategic principles such as those used by human players. This paper outlines the foundations for the development of such an algorithm, which relies heavily on artificial intelligence techniques such as case-based reasoning. Besides outlining a general solution for each phase of the game, the paper focuses on the implementation and evaluation of a case-based approach for playing the middle game in chess.

1 Introduction

Ever since Shannon [1] published his proposal for a chess playing program, most programs have followed the brute force approach to chess, which relies on searching a large number of possible chess positions in order to produce a move that is appropriate for a given chess position.

Some systems have used a knowledge-based approach to deal with chess positions. Unfortunately, these programs have been able to deal only with very limited subsets of the game. For example PARADISE [2], and Pitrat's program [3] use planning techniques to deal with certain middle game positions for which a solution is known to exist within a short horizon, while Michie's advice language [4] applies artificial intelligence (i.e. AI) techniques to solve very simple endgames. More recently, systems such as ICONCHESS [5], and Kerner's system [6], have tried CBR approaches to deal with chess middle games.

This paper presents an alternative to computer chess playing by approaching the game in a way that is closer to the way human players play the game than search-based chess playing programs.

Section 2 presents some of the well known chess playing principles for each phase of the game, and describes how they can be implemented in a chess playing program.

Section 3 focuses on the middle game, which is the most difficult part of the game in terms of the application of general principles, and outlines an implementation of a case based approach to play this phase of the game. Section 4 presents experimentation that was used to evaluate the implementation described in Section 3. Section 5 describes some possible generalizations and extensions to this approach for chess middle game playing. Section 6 presents potential applications of this approach to other domains, conclusions and future work.

2 Principle Based Chess Playing

A game of chess can be roughly divided into three different phases, the opening, the middle game, and the end game. For each of these phases different general principles apply. The remainder of this section describes some of the principles used in each phase of the game and their implementation in a chess playing program.

2.1 The Opening

Most intermediate and advanced chess players memorize a significant number of opening lines in order to play this phase of the game quickly. However, before beginning chess players start memorizing hundreds of moves in different well known opening variations, they should learn the general principles of the opening. Some of these principles are, for example: Control the center of the board, Develop your pieces, Move your king to a safe place, Do not move the same piece twice unless absolutely necessary, etc. Of course, there are some situations that require breaking one or more of these rules, but most of the times the correct application of these principles yields a good position.

Nowadays most tournament players prepare opening repertoires, or opening libraries in chess playing-programs, but there is always the possibility of being taken "out of book" (e.g. to a position that is not present in our opening library/repertoire). When this happens, the player can usually find acceptable moves by applying opening principles. If you ignore these principles, and rely purely on search algorithms, you risk not finding the most adequate move due to possible horizon problems (e.g. you cannot see far enough in your search to properly evaluate the position reached).

A different way to play chess openings is to encode principles such as those mentioned above so that it is easy to identify those moves that follow most of them. The idea is to take the results from a move generator and apply a filter to these results, so that only those moves that apply the opening principles are considered as candidate moves. This approach has yet to be tested in a computer program, but a successful implementation would be more economical in space than current approaches, since opening libraries would not be necessary, and in time, since it would not need to spend time searching thousands of positions. Ideally, this kind of chess playing program would start with a small number of general principles and then it will learn new ones by experimentation.

2.2 The Middle Game

The middle game is the most complex part of the game, since this is the phase of the game where we find the largest number of possible moves. Given that the pieces have been developed, and not many of them have been removed from the board, there is usually a very large number of moves to analyze for each player. Unfortunately, in this phase of the game there are no general principles, but depending on the type of position it is possible to determine specific goals for which a player must strive.

In order to determine the principles or specific objectives that apply for a given position, it is possible to use a case based approach, such as the one implemented in ICONCHESS [5]. The idea is to have a case base of middle game chess positions indexed by relevant features, both syntactic and semantic, and a description of the main objectives for each of those positions along with a description of the recommended actions for each player. When a new position is reached, it is indexed according to the same features used in the case base, and the most similar positions are retrieved from the case base. Finally, the program determines what recommendations from the cases retrieved can be applied to the new position. Sections 3 and 4 describe these ideas in further detail.

2.3 The End Game

As pieces are removed from the board, the game simplifies and the number of possibilities reduces. The end game arises when a reasonably small number of pieces (usually 2 or 3) and pawns (usually 5 or less) are left on the board. At this point in the game, the reduced strength of the players makes it difficult to develop direct attacks on the king, so the king becomes most of the times a very active and frequently decisive piece. So the principles for the end game may include: Centralize your king, create passed pawns (e.g. pawns that have a good chance to be promoted).

When the total number of pieces and pawns is small enough, it is even possible to make use of end game tables that completely solve a given type of end game and their use guarantees perfect play. Before reaching that point it is also possible to use principles to select the best line of play in a very similar way to the approach described earlier for playing the opening.

3 Case Based Approach for Playing the Middle Game in Chess

In ICONCHESS [5], a case based approach to the generation of high-level advice was implemented and tested in terms of the usefulness of the advice generated for different users. In order to generate this advice, ICONCHESS relies on a case base of middle game chess positions. Each case consists of a particular position indexed by its relevant features, a list of recommended actions for each player, and the preconditions that were present in the position that suggested the recommended actions. Additionally the case may contain some moves that show how the indexed position was reached, how the game proceeded after the indexed position, or the entire game in which the position was reached.

The indexing scheme is the basis for a similarity metric used to compare new positions with those available in the case base. The indexes were built in terms of four different metrics: Material, Pawn Structure, King Protection, and Influence. These metrics, as used in ICONCHESS, are briefly described in the following subsections, a more comprehensive description can be found in [5].

3.1 Material

The material metric is divided into:

1) Total Material, computed by using a table of values for each piece and pawn, such as the one proposed by Shannon [1].

2) The number of fast pieces refers to the total number of pieces for each player, disregarding pawns and kings.

Table 1. Format of a Fuzzy Table.

Column	0	1	2	3	>3
a	w1	.	.	.	.
b	w2		w3		
c	.				
d	.			w4	
e	.				
f	.	w5			
g	.				w6
h	w8				

Fig. 1. Influence Regions

3.2 Pawn Structure

The pawn structure is useful to determine to some extent the type of opening that was used to reach a position. This is the first step to determine the possible strategic objectives for both players. ICONCHESS relies on a set of **Fuzzy Tables** which are predefined patterns that represent some of the most common pawn structures encountered in the middle game (i.e. **Canonical Structures**).

Each Fuzzy Table represents one particular Canonical Structure. A Fuzzy Table allows the system to classify the pawn structure that appears in a given position in terms of how close it is to the Canonical Structure represented by this particular Fuzzy Table.

For each Canonical Structure, the following information is kept:

Name of Pawn Structure (e.g. Benoni formation)

Relevant Characteristics, such as strong pieces, main objectives, weaknesses, etc. **Short Name** (e.g. Benoni)

A **Fuzzy Table** that contains the weights assigned for pawns located in each possible square.

A Fuzzy Table's format is illustrated in Table 1. In that table the value w1 in the entry [a,0] stands for the weight given to a pawn located in column a, which has not moved from its original position, w3 stands for a pawn in column b and has moved 2 squares from its original position, and so on.

3.3 King Protection

Three different factors compose a **king's protection** evaluation as understood by ICONCHESS: the actual location of the king, which allows the program to guess if it has castled or not, the pawn cover, or the number and location of pawns that protect it, and the comparative strength of defensive and attacking pieces that have a direct impact of the king's position. These factors are used to index a table that contains information about specific weaknesses of the specific configuration.

3.4 Influence

Influence is a metric introduced in ICONCHESS as an attempt to determine the impact of the **fast pieces** (e.g. queens, rooks, bishops, and knights) on the different parts of the board. This is useful to get a general idea of the balance of power throughout the board. Influence is related to the well known concept of mobility in chess, and it is defined as the number of possible moves of the fast pieces of a color in 6 different sectors of the board as defined in figure 1. These sectors are similar to those used in chess literature and are identified for both players as follows:

UQS = Upper Queen Side	UC = Upper Center	UKS = Upper King Side
LQS = Lower Queen Side	LC = Lower Center	LKS = Lower King Side

The influence of each player in each of these sectors is computed by counting the number of possible moves of each piece that is neither a king nor a pawn in each of those sectors. Additionally, the influence of each player on each king's position (e.g.all the squares surrounding the king) is also computed, and it is the one amount preceded by a BK (Black King) or WK (White King) in the influence area of the evaluation window.

3.5 The Similarity Metric

ICONCHESS's similarity metric is based on the evaluation factors described above: material, pawn structure, king protection, and influence. As the evaluation factors are computed, a specific weight is assigned to each of them according to the particularities of the position. These weights are used to combine the different evaluation factors through a weighted average in order to generate a similarity metric that can be used to retrieve relevant cases from the case base. The weights are an estimate given by ICONCHESS as to how important each factor is to classify the position.

The first factor that is consulted is the pawn structure and its closeness value. If the closeness value is high, it is assumed that the position is close to the opening, and thus the pawn structure is given a high value, then, the weight for the influence is calculated in reverse proportion to the weight given to the pawn structure. The weights assigned to the material and king protection rely mainly on their discriminative value. Furthermore, the user is given the opportunity to select alternative pawn structures and to modify the weights that are involved in the computation of the similarity metric before making the actual search for relevant cases.

4 Experimentation and Results

Several positions were used to test the ability of the system to retrieve relevant positions from the case base for a new position, with successful results. This section presents a detailed expert analysis[1] of two relevant positions and the similar positions encountered by the similarity metric.

4.1 A case retrieved with direct similarity

Figure 2 presents a case in which the position retrieved from the case base is very similar in terms of strategic/semantic features. Both of the positions were taken from local Mexican chess tournaments in 1984 and 1985 respectively. Even though the positions may not look very similar to the casual observer, an experienced player will easily find the resemblance in terms of the strategic objectives that may be followed by each player. ICONCHESS discovers the similarity between these two positions by considering the different components of the similarity metric As we can see in the southwest window in Fig. 2, both Pawn Structures have a great percentage of similarity (98) with canonical pawn structures known to ICONCHESS. This fact leads ICONCHESS to use a high weight (10) for this feature of the similarity metric, so that it is used as the dominant factor for the comparison.

As we can see, the white pawn structure in the position retrieved from the case base is very similar to the one in the original position, and the black pawn structure is

[1] Since I was a part in all of the games included in the case base, I am using myself as the expert analyst.

identical, so it is not difficult to see why this position was selected from the case base. Additionally, if we consider the influence in both positions, we can see that there is some dominance of the white pieces in the entire king side of the board, so this fact is a reinforcement for choosing this similar case.

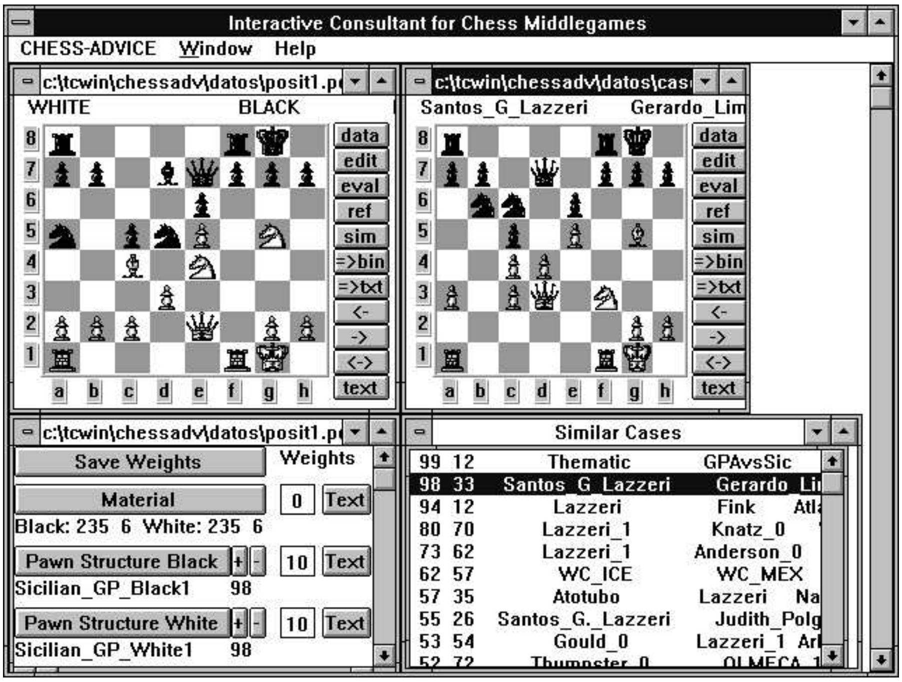

Fig. 2. Direct Similarity Example

From an expert point of view, the result is validated since in both positions white is in a situation where it can take advantage by starting a king side attack by means of a sacrifice. In the first position, a possibly deciding move is Knight takes pawn on h7, and in the retrieved position a starting move could be Bishop to f6.

4.2 A Case Retrieved with Reverse Similarity

Fig. 3 shows the retrieval of a position using the concept of reverse similarity. As in the previous case, the positions are not nearly identical, but there are important strategic similarities that allow us to take advice from one position into the other.

In this example, both position were reached in actual tournament games, Shambry-Lazzeri was played at the World Open in Philadelphia 1993, while Lazzeri-Polgar was played at the New York Open 1988.

Since the similarity in terms of pawn structure is minimal, the concept of influence is the most important one. At this point, ICONCHESS realizes there is a very strong reverse similarity, particularly between black's position in Shambry-Lazzeri and white's position in Lazzeri-Polgar.

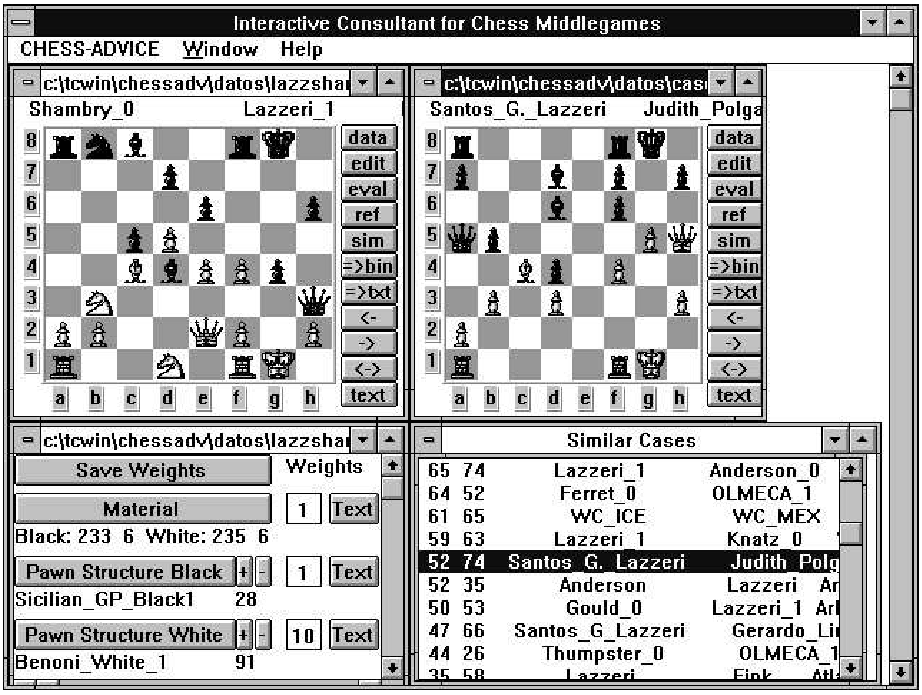

Fig. 3. Reverse Similarity Example

Fig. 4 shows in full detail the complete evaluation computed by ICONCHESS for each position. The evaluation on the left corresponds to Shambry – Lazzeri, while the one on the right corresponds to Lazzeri –Polgar. There we can see that in the first case the suggested weights for white's pawn structures (10) and influence (6) are considered the most important factors, the king protection for both players is somewhat important (4), and the other factors are almost irrelevant (1). These are the weights used by ICONCHESS to compute the similarity metric that it applies to determine which cases are most similar.

When making the comparison with the Lazzeri-Polgar position, it is evident that the reversed similarity is important, since the Benoni_White_1 has a 60% match with the pawn structure for Black, which is very similar although it is not as high a match as that for the pawn structure for white (91%). Also the classifications of the king protection show a very high reverse similarity.

Finally, the influence metric shows that Black has an important advantage in the king side, both upper and lower (7 8) versus (0 2) and on the kings positions (3 4) versus (0 0) in Shambry-Lazzeri, while the reverse is true in the Lazzeri-Polgar position, where white has an important influence advantage on the king side (5 5) versus (1 3) and still some advantage on the influence on the king's positions (3 1) versus (0 0).

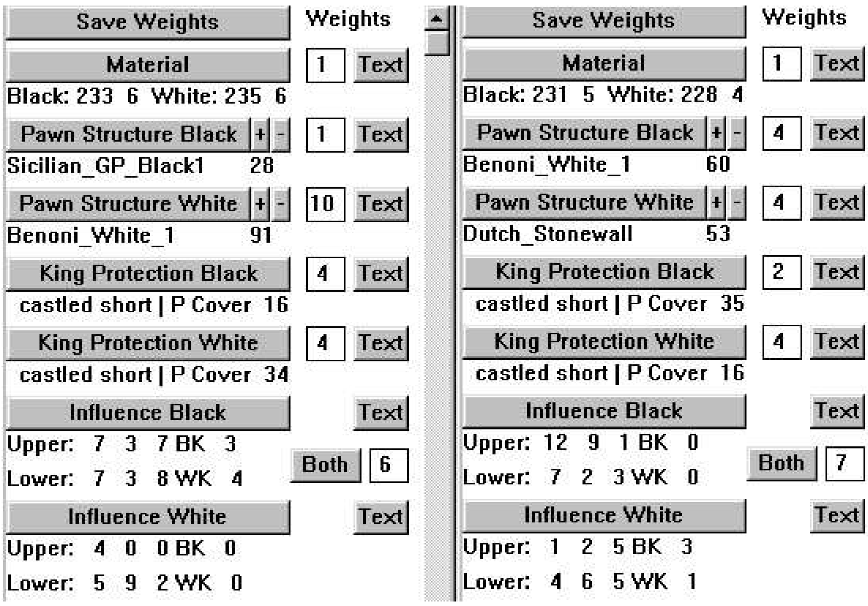

Fig. 4. Full Evaluation of the positions in Fig. 3.

5 Extensions and Generalization

From the analysis presented in Section 4, we can see that the technique applied by ICONCHESS to retrieve relevant positions using a similarity metric is successful. However, it is still possible to improve on the current implementation.

First, it is possible to make some extensions so that a complete game can be played by combining the proposed approaches for each phase of the game. In order to achieve this goal, we need to implement algorithms so that actual moves can be chosen for any position in a game. Another extension may involve the creation of larger case bases, both for the positions and for the canonical pawn structures.

Second, some of the concepts may be generalized so that the similarity metric may become more powerful. For example, the influence metric may be generalized so that regions of any shape and size may be considered, rather than just the predefined regions described in this paper. The implementation of the detection of this kind of influence areas, also known as Dynamic Influence Areas is currently under way and it may allow a more flexible way to find chess positions that can be useful in the analysis of other positions.

6 Applications to Other Domains, Conclusions and Future Work

This paper has presented the implementation of a case-based approach for retrieving relevant middle game chess positions in terms of their similarity to a target position. Chess has been chosen as the domain of implementation because of the characteristics of the game that make it easy to evaluate the correctness of the system.

However, there are other domains where these techniques may be useful. Such a domain may be for example disaster prevention. A particular case is volcano eruption in the context of the Popocatepetl volcano [7]. In this context it is important to determine danger and risk areas depending on weather conditions, intensity of eruptions, and other factors, such as importance of the potentially affected areas in terms of population and facilities threatened. For this problem we can see an important analogy with the generalization described in Section 5, regarding the Dynamic Influence Areas. The only difference is the way in which these areas are computed, since instead of using the rules of chess it is necessary to analyze the behavior of nature.

This paper described also the foundations for the development of a principle-based chess playing program. Furthermore, the paper analyzed the implementation of a particular case-based algorithm for finding relevant positions in the context of chess middle games, and its effectiveness was shown by means of a detailed analysis of relevant examples.

Possible extensions and generalizations were also presented, as well as potential applications to other domains. The implementation of Dynamic Influence Areas as described above is currently under development for the domain of chess. Other near term developments are the implementation of a complete chess playing program based on principles, and the application of some of these techniques to different domains such as the disaster prevention domain mentioned earlier.

References

1. Shannon, Claude: Programming a Digital Computer for Playing Chess. Philosophical Magazine Vol. 41. NO 7. (1950) 356-375
2. Wilkins, D.E.: Using Patterns and Plans in Chess. Artificial Intelligence. Vol 14. (1980) 165-203
3. Pitrat, J.: A Chess Combination Program which Uses Plans. Artificial Intelligence. Vol 8. (1977) 275-321
4. Michie, D.: An Advice-taking System for Computer Chess. Computer Bulletin 10. (1976) 12-14
5. Lazzeri, Santos G., Heller, Rachelle: Application of Fuzzy Logic and Case Based Reasoning to the Generation of High-Level Advice in Chess. Advances in Computer Chess 8. Universiteit Maastricht, The Netherlands (1996) 251-267
6. Kerner, Y.: Learning Strategies for Explanation Patterns: Basic Game Patterns with Applications to Chess. Proceedings of the 1st International Conference on Case Based Reasoning Research and Development. Sesimbra, Portugal, (1995) 491-500
7. Posada, Nidia: CBR y Toma de Decisiones en el Contexto de un GIS: Caso del Volcán Popocatépetl, Tesis de Maestría, Fundación Universidad de las Américas, Puebla (2001)