A Hybrid Approach to Representing Chessboard using Bitboard and Compact Chessboard Representation

Sigit Kariagil Bimonugroho, Nur Ulfa Maulidevi
School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia
E-mail: sigitkariagil@gmail.com, ulfa@informatika.org

Abstract. A chess engine is a computer program that can perform position analysis in chess and take which are the most optimal move. The main process in a chess engine can be divided into three: move generation, searching, and evaluation. Each of these processes are interrelated and require efficient board representation methods. Some methods of representation from previous research are efficient in some processes but less efficient in others. This study proposes an efficient representation method with a hybrid approach, which combines Compact Chessboard Representation (C.C.R) and bitboard representation. The C.C.R method is used in move generation and searching processes to reach conciseness, and bitboard representation method is used in the evaluation process to make it faster. The result of the movement validity test for the hybrid representation method shows the move generation is valid, and a hybrid-based chess engine also can do searching and evaluation faster than a pure C.C.R based chess engine. Keywords: data representation, chessboard, hybrid, compact chessboard representation, bitboard

1. Introduction
Since the mid-1960s, researchers in computer science have called chess as a drosophila of artificial intelligence (AI) [1]. Drosophila is a genus of small fruit flies used by Thomas Hunt Morgan to study the basic mechanism of inheritance in biological science. The phrase drosophila of AI is used to show that chess, like fruit flies, is an experimental technology that is easily accessible, general, and relatively simple but can be used productively to produce valid knowledge about other more complex systems.

There are three main processes in chess engine: move generation, searching, and evaluation. Move generation is the process of generating all candidate move. Searching is the process of finding optimal move that can be taken based on evaluation score. Evaluation is the process of calculating score of a position. In each of these processes requires data representation to store position information.

An efficient chessboard representation is needed to build a good chess engine. As Reul [2] notes, the benefit of such data structures ”lies in their simplicity and ease, and the possibility to implement specific and complex chess knowledge in a more compact and transparent way”.

At present there are two basic representation methods used in chess engines, representation with arrays and representation with bitboards. Array representations were widely used at the beginning of computer chess research. The array representation was first mentioned by
Shannon on 1950 [3]. Then, along with the development of hardware, especially 64-bit CPUs, a bitboard representation method (a set of bit-vectors or bitmaps) was discovered. The bitboard representation method has advantages in terms of efficiency compared to the array representation method [4].

The idea of bitboard representation was developed by Atkin and Slate on 1988 [5]. Atkin and Slate’s research describes an approach using 12 64-bit integers to represent 6 types of black chess pieces and 6 types of white chess pieces. Research by Adelson-Velskii et al. on 1970 [6] developed the same idea separately at the same time as Atkin and Slate’s research. The advantage of this method is that the evaluation process can be carried out quickly using bit operations [4]. This approach is widely used in chess engines until now.

Some research attempts to improve the performance of bitboard representations. Research R. M. Hyatt on 1999 [7] succeeded in increasing the performance of bitboard representation at the move generation stage by the bitmap rotation method. However, this method is not good when used in the search stage because it is less concise [8]. Therefore, Vuckovic on 2012 [8] proposed a new method of representation that is more concise, namely Compact Chessboard Representation (C.C.R). The C.C.R method can provide good performance at the search stage because it is concise. However, C.C.R is not fast enough when used at the evaluation stage. Therefore, we proposed a hybrid approach that combine C.C.R and bitboard representation to solve the inefficiency problem in chessboard representation.

2. Proposed System
The development of data representation with a hybrid approach in this research was carried out in conjunction with the development of a chess engine. The design of the developed chess engine consists of move generation, searching, and evaluation. C.C.R representation is used in the move generation and searching stages. At the evaluation stage the conversion process is carried out from C.C.R to Bitboard. The general design of a chess engine is shown in Figure 1.

2.1. Move Generation
In this research, move generation process uses the C.C.R representation to store the position that the move will be raised for. C.C.R representation was chosen because it is concise. Position representation using the C.C.R representation only requires 32 bytes of memory while a representation using bitboard requires a minimum of 96 bytes of memory. Input from the move generation process is the position of the pieces represented in C.C.R. Output of the move generation process is a list of movement candidates. Illustration of input and output of the move generation process is shown in Figure 2.
2.2. Search
Searching process is carried out by tracing the movement candidates from the move generation process. Input from the searching process is a movement candidate, and the output is a chosen move. Illustration of input and output searching process is shown in Figure 3.

Searching process in this study uses the Alpha-Beta algorithm. Alpha-Beta method is a simpler and more efficient than the Minimax algorithm. This is because the Alpha-Beta algorithm trims sub-trees that have no effect on the final grade [9, 10].

The results of a comparison of the search process for the sample position in Figure 4 using the Alpha-Beta and Minimax algorithms are shown in Table 1. Comparison results show that the Alpha-Beta and Minimax algorithms provide the same output. However, number of nodes traversed by the Alpha-Beta algorithm is less than the Minimax algorithm. From these results, it can be concluded that the Alpha-Beta algorithm is more efficient than the Minimax algorithm when used in the searching process in a chess engine.

2.3. Evaluation
Evaluation process is done to calculate the heuristic value of a chess position at the terminal node of the search process. Heuristic value used in this study is the material value and position value. Material value is calculated based on the number of pieces left and the weight of each piece. The position value is calculated based on the position held by the piece. For example, a pawn in the middle position will be given a higher value than a pawn in the back. Input from
the evaluation process is a position in the representation of C.C.R. Output of this process is the value in integer. Illustration of input and output of the evaluation process is shown in Figure 5.

Position evaluation process can be done by two methods. Process of evaluating the position of the first method is done by calculating the heuristic value directly from the representation of C.C.R. Second method evaluation process is done by calculating the heuristic value from the bitboard representation. Both methods are applied in this study. This is done to compare the performance of hybrid representation-based chess systems with C.C.R-based chess systems only. Position in the bitboard representation is obtained by performing the C.C.R position conversion process.

3. Result
Evaluation process of the proposed method was conducted by two methods, search speed test and chess match test. Speed test is used to measure the speed of searching process. Chess match is used to find out the effect of the representation developed on the overall engine performance.
3.1. Search Speed Test
Search speed test is carried out with three scenarios. The three scenarios represent the condition of the board at the opening, middle, and end game. These three scenarios are used to test the consistency of the search process speed from the beginning to the end of the game. The speed test results are shown in Table 2. Speed are shown in nodes per second. The speed in nodes per second is measured by counting the number of nodes visited by the machine during the search process divided by the length of the processing time.

Table 2. Search speed test in node per second

| Position   | Bitboard | C.C.R  | Hybrid |
|------------|----------|--------|--------|
| Opening    | 521225   | 618475 | 662625 |
| Middle     | 350175   | 601750 | 659525 |
| End Game   | 1029150  | 799075 | 788800 |

3.2. Matches against other computers
The chess match test is carried out by playing a chess match against the bitboard based engine and the C.C.R based engine. The results of the match test are shown in Table 3.

Table 3. Matches result

| Opponent  | Win | Draw | Lose |
|-----------|-----|------|------|
| vs Bitboard | 3   | 1    | 6    |
| vs C.C.R  | 6   | 2    | 2    |

4. Conclusion
The utilization of the representation method that combines the C.C.R and bitboard methods has succeeded in improving chess engine performance in terms of speed and memory usage in the opening and middle game. C.C.R representation method is best used in the process of move generation and searching, while the bitboard representation is good for the evaluation process. However, the hybrid representation method has not been able to provide optimal results in terms of speed in the end game compared to bitboard. This is because at the end of the game many empty squares are still checked by the hybrid representation method. Therefore, future research can focus on increasing the speed of the search in chess engine in the end game. This is because the engine based on hybrid representation method already has good performance in the opening and middle game. However, it is still not good in the end game. One approach that can be proposed is to use representations that dynamically adjusted. For example, the hybrid representation is used during the opening and middle game. Then, the bitboard representation is used fully when entering the end game.

Reference
[1] Ensmenger N 2012 Social studies of science 42 5–30
[2] Reul F 2009 New architectures in computer chess Ph.D. thesis series: TiCC Ph.D. Series Volume: 6
[3] Shannon C E 1950 The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 41 256–275
[4] Hyatt R 2004 University of Alabama at Birmingham
[5] Atkin L and Slate D 1988 Computer chess compendium ed Levy D (Berlin, Heidelberg: Springer-Verlag) chap Chess 4.5-The Northwestern University Chess Program, pp 80–103 ISBN 0-387-91331-9 URL http://dl.acm.org/citation.cfm?id=61701.67010

[6] Adel’son-Vel’skii G M, Arlazarov V L, Bitman A, Zhivotovskii A and Uskov A 1970 Russian Mathematical Surveys 25 221

[7] Hyatt R M 1999 ICGA Journal 22 213–222

[8] Vučković V 2012 Yugoslav Journal of Operations Research 22 265–284

[9] Campbell M S and Marsland T 1983 Artificial Intelligence 20 347 – 367 ISSN 0004-3702 URL http://www.sciencedirect.com/science/article/pii/0004370283900012

[10] Pearl J 1982 Communications of the ACM 25 559–564