Research on the Opening Book of the computer game of draughts

Xueping Liu1, Kai Zhao1, Can Tian2, Yueqi Yang2, Shi Yu2
1Engineering Training Center, Shenyang Aerospace University, Shenyang, People’s Republic of China
2Faculty of Aerospace Engineering, Shenyang Aerospace University, Shenyang, People’s Republic of China
✉E-mail: liuxueping024@163.com

Abstract: Opening Book is a kind of assistive technologies to enhance the performance of computer games. The opening stages of the game method generally used to query the database generated. This method improving search efficiency and avoids the missing strategic of traditional evaluation systems. This paper studies the technical problems of using the Opening Book of draughts, and introduces the generation and usage of the Opening Book. The authors also discussed the detail of statistics in Opening Book. Besides, this paper presents a new idea of introducing the information of the Opening Book into the traditional Alpha-Beta valuation system. The experiment proved that the method proposed in this paper can effectively solve the issues of the start of the draughts and improve the game level of the draughts.

1 Introduction

The current computer game [1–4] is an important research direction in the field of artificial intelligence. Its complexity and challenge attract many scholars to study. The study of the game of chess is an important branch of computer game research. Draughts [5, 6] are evolved from national draughts from all over the world. They can be traced back to the pharaohs of ancient Egypt. From ancient times to the present, draughts such as America draughts, Italian draughts, Spanish checkers, and other representative checkers appeared. Board of draughts [7–9] is a kind of square board, made up of 10 × 10 dark and light squares. Usually, those squares are called ‘Black and White square’. The place where chessmen can move are black squares. As shown in Fig. 1, Black squares are numbered from 1 to 50, which are called chess areas.

Draughts game is played by two players. Each player has 20 chessmen and draughts require both players to move along the bevel, and the chessmen can jump the opponent’s chessmen then eat those chessmen which are jumped until all the chessmen on one are eaten by the other player or all chessmen cannot continue moving on the board. In the draughts process, ‘draw’ as a final is allowed [10].

Because of the rules like chessmen must be captured by jumping over the opponent’s chessmen, Jump over as many chessmen as possible etc. Therefore, it is necessary to force the opponent to lose chessmen by using the opposing positions or holding opponent’s chessmen, or to change Men to King in advance to change the strategy of comparing the two sides.

A draughts game process is divided into three stages: the start-stage, the mid-stage, and the end-stage. The beginning of a match is called the start-stage, which means the first 10–15 moves after the start of the match.

The common beginning of draughts [11], including the Lærafr staart system, the centrally symmetry start system, the Rosenborg start system, the Sprint start system, the Polish start system, and the French start system. Although these opening systems have their own different characteristics, they all jointly follow the following principles: the centre of competition, the mutual coordination among chessmen, and strong support. In addition, it is significant to pay attention to the connection between chessmen. Moving of chessmen in the position that can make the opponent’s pieces become the king need to be considered clearly. The start-stage is the foundation of the overall situation [12–15], which has a great effect on the whole process and result. The first step in the start-stage is to define the strategic objectives, organise and plan the troops, and seize the initiative of the game.

Therefore, the position of the start-stage is very crucial in draughts computer game system [16–20]. The traditional checkers game system adopts much blindness and lacks of strategy in the start-stage to adopt the same method of selecting the moves of relying on the value assessment of chess pieces as in the mid-stage and the end-stage. This is obviously inconsistent with the principle of start-stage. It is extremely unfavourable to the development of the situation in the mid-stage and end-stage. To solve this problem, it is necessary to draw lessons from various classic and mature starting methods of draughts, to develop a start-stage database of draughts game system to apply the initial tactics.

2 Importance of Opening Book

The so-called Opening Book [21–25] is a special form of database in which the draughts classic and mature moves’ information are organised. In the start-stage, the system selects the more favourable move by calling the information in the Opening Book and cooperating with the traditional search and valuation system. Compared with the traditional search and evaluation system, the draughts game system with the Opening database on the traditional basis shortens the search time in the start-stage. At the same time, pieces move more strategically, and avoid the simple use of the game system of the Opening Book cannot deal with some unpopular situation. The use of opening database can form a more secure, favourable situation.

Table 1 is statistical information of 1561 draughts games about moves of the first hand and the result. There is no significant difference of winning percentage between ‘the first hand’ and ‘the second hand’. With different moves for the first step, there is a clear difference between the first hand and the second hand in winning percentage. The most commonly used move is 32–28 (Lærafr staart system), accounting for 59.58% of the total number of games, excluding the use of a smaller number of initial moves of interference. The winning percentage of this move has no obvious advantages compared with other moves but has a higher rate of draw. There was 21.08% of the games started with the move of 33–29 (Rosenborg staart system). The winning percentage was significantly higher than that of other moves. Thus, the choice of 33–29 moves can help the first hand to produce a stable, favourable situation in the process of game. If moves with higher winning percentage for first hand were taken, it still can be draw or even win at last unless pieces moved by mistakes. These statistics above
The system used Zobrist hash [26–28] technology to initialise a two-dimensional array Zobrist [50] [5] at program starting up and filling the array with random numbers. The random number was 64-bit integer data and stored in a pre-set file. Each time the system started, simply read the data in the file to complete the data initialisation.

Hash calculated by all the chess areas on the information in the array Zobrist each element corresponding bitwise exclusive or calculation. The system needed to be calculated the hash of the new situation generated by the current situation. After the hash of the current situation and the random number of the chess areas pair that had been changed before the move are exclusive, the information corresponding to the chess areas after the change was subjected to the exclusive-OR operation.

For \( \Psi(P_i) \) had the following definition:

\[
\Psi(P_i) = P_i \otimes \Psi(P_1) \otimes \cdots \otimes \Psi(P_{n-1}) \otimes P_n 
\]

For \( P_i (i=0,1,2,\ldots,49) \), there is \( P_i = \text{Zobrist}[a_i][\text{type}] \).

If the hash value of a certain situation (A situation) was \( H_i \), then the following relationship:

\[
H_i = 49 \Psi(P_i) \quad (i = 0) 
\]

Situation B was created after a move being taken in situation A, and the chess areas which changed were represented by \( a_1, a_2, a_3 \ldots \).

Then the following relationship:

\[
Q_n = \text{Zobrist}[a_i][\text{type}] 
\]

Therefore, the hash value \( H_i \) of B situation could be obtained from the following formula.

\[
H_i = H_i \otimes \Psi(P_n) \otimes \Psi(Q_n) 
\]

### 3.3 Structural design of Opening Book

In the draughts, different moved may produce the same situation, and the same moves may also correspond to different situations. Therefore, these factors should be considered when designing the Opening Book. Fig. 2 shows a demonstration of two different situations.

Figs. 2a and 2b were two different situations in the draughts. At this point, the blue could move the pieces. There were 11 legal moves of the situation shown in Fig. 2a and 2 legal moves of the situation shown in Fig. 2b. It would generate the same situation, the situation was generated as shown in Fig. 3, after the situation in Fig. 2a adopting the moves of 31–27, and Fig. 2b adopting the moves of 38–27.

From this, we could know that different moves in different situations may produce the same situation. Therefore, the situation generator of the game system to select records in the first 20 steps for each game could be generated the situation information to create the database.
different moves in draughts may produce the same situation, and the same moves may also correspond to different situations. Therefore, these factors should be considered when designing the Opening Book. Before generating Opening Book, a structured array of BookInfo named ‘Open Library’ with a certain number of elements should be called.

To ensure the corresponding information of the first 20 steps of the 1561 matching was used as much as possible in the Opening Book. We counted the information, and in the results, we found that in the case of the same hash, no different situation was found, there was no hash conflict. The information in the Opening Book was stored in an array constructed by the structure BookInfo. However, the generated hash was a 64-bit long data, and that would make some difficulties in the using of hashes to access the database. Therefore, there was a necessary to make proper calculation for hash to shorten the length of the hash, and to determine the capacity of the constructed structure.

First, we removed the duplicate 64-bit long hash to make each hash unique. The hashes were then bitwise ANDed with 255 (0xFF), 4095 (0xFFF), 65535 (0xFFFF), 1048575 (0xFFFFF), and 16777215 (0xFFFFFFFF), respectively. The resulting value was called a ‘short hash,’ and the value that was used to shorten the hash could be used as the size of the structure array. The generated short hash was used as the index of the array. Table 2 was generated for different short hash statistics.

Table 2 Different short hash statistics

| Type     | Total | Use | Collision |
|----------|-------|-----|-----------|
| 0xFF     | 255   | 128 | 128       |
| 0xFFFF   | 4095  | 2048| 2048      |
| 0xFFFFF  | 65535 | 27983| 6        |
| 0xFFFF75 | 1048575| 27983| 0        |
| 0xFFFFFFFF| 16777215| 27983| 0        |

if MoverNumber=0
return 0;
else if MoverNumber=1
return Mover;
else
foreach Mover in OpenBook
if find Mover in OpenBook
Add information in list
if Number of mover in list=0
goto Alpha-bete to get best mover
Get Best Mover in List

3.4 Generation of the Opening Book

Different moves in draughts may produce the same situation, and the same moves may also correspond to different situations. Therefore, these factors should be considered when designing the Opening Book. Before generating Opening Book, a structured array of BookInfo named ‘Open Library’ with a certain number of elements should be called.

3.5 Use of Opening Book

In the Opening Book, we first needed the program needs to read data file of the Opening Book, and then filled in the relevant data information after initialising the array. HashKey default value was –1, Opening Book may encounter uncounted situations in actual use, the hash generated by these situations to find the array information was obviously meaningless. So, we defaulted value of –1 for the HashKey to determine whether the structure of the array makes sense, to avoid one by one each of the chess area caused by the waste of computing resources. The relevant pseudo-code was as follows (Fig. 5):

The use of Opening Book first needed to determine the number of legal moves. If there was no legal move or only one, the relevant information should be returned. If there were multiple legitimate moves, these moves needed to be selected.

For these moves to be selected, the system generated the situation information after completing the moves one by one and generates a hash and finishes searching the array of structures. The information in each of the Chess areas was then compared one by one to see if the situation was the same, and then the moves and results of the same situations were stored in the list of candidates. After all the moves traversal was completed, the preferred method was to score high. If all results stored for pending moves were negative, there was still situation information that was successfully match Opening Book. Use traditional evaluation, search methods in moves that had not been matched with the database. If there was absolutely no matching situation information, the best moves should be selected according to the traditional evaluation and search techniques.

3.6 Improvement of traditional search method

In order to improve the utilisation efficiency of Opening Book, the game system start-stage was used the Alpha-bete to select the most
optimal moves, and added a database weight when the evaluation function was called to evaluate the cotyleden nodes. If the current situation did not match the Opening Book, but the cotyleden node could match the Opening Book and the situation was more favourable (with greater probability of winning or draw), the moves may be preferred. At the same time, it should be considered that the introduction of such a valuation system into the initial stock market would undoubtedly be useless for the game system to spend time matching the data in the mid-stage and the end-stage. Therefore, in the valuation of the database, we find that we would use only for the start-stage according to the search depth of the corresponding adjustments.

4 Experimental results

4.1 Experimental platform

In order to get better game data, we build a distributed system platform. As shown in Fig. 6, the platform is composed of two parts, the host computer and slave computer. The host is responsible for completing the task assignment and data statistics function, and the slave is responsible for completing the game execution. First, the host releases the game commands and parameters to each slave. After receiving the command, the game is executed according to the received parameters; then the host asks the slave to complete the game, and the player will return the game chess and the result to the player. The whole process continues until all the games are completed. The host automatically calculates and counts the results based on the results received.

4.2 Analysis of game results

The program with the Opening Book and the program without the Opening Book conducted multiple game experiments, and the statistical results were shown in Fig. 7.

A total of 40,000 games were tested, of which won 30,071 times, 14,582 times was the first hand and 15,489 times was the second hand. It could be seen from the use of opening book played a good role. However, the analysis also found that using the Opening Book, the probability of eating the first five steps significantly reduced, completed the layout process, and no obvious offensive process. Also during the obvious offensive in the first 10 steps, it was clear that the early stages of the game play an important role in the overall opening. From the mid-stage, the two sides adjusted the layout so as to use more favourable situation to enter the end. This shows that Opening Book played a crucial role in mid-stage and end-stage.

In the future, we should consider optimising data structure to improve computational efficiency and data utilisation efficiency. Then we also need to improve the opening database data through the advanced international checkers program (such as Dragon and Dam).

5 Conclusion

In this paper, we discussed the design method of Opening Book of draughts game system. Through statistical analysis of information and sorting, Opening Book can be used in the draughts system. We put the Opening Book in the middle and early stages of the game to the introduction of the traditional Alpha-Beta search, which laid a good foundation for the post office. Finally, the optimisation method is put forward, and the design idea of ‘learning’ with advanced program was put forward, the optimisation method is put forward, and the design idea of ‘learning’ with advanced programs is put forward. We proved the effectiveness of the method through experiments and contributed to the design and research of the beginning of checkers. The method proposed in this paper is universal and can also be used for the opening research of other chess species. Next, we hope to add deep learning in the Opening Book, so as to further improve the level of draughts.

6 Acknowledgments

This work was supported by the National Nature Science Foundation of China (No. 61203170, 61602320) and the Scientific Research General Project of Education Department of Liaoning Province, China (No. L201608).

7 References

[1] Xu, X.H., Deng, Z.L., Wang, J., et al.: ‘Challenging issues facing computer game research’, CAAI Trans. Intell. Syst., 2008, 3 (4), pp. 288–293.
[2] Wang, Y., Qiu, H., Wu, Y.Y., et al.: ‘Research and development of computer games’, CAAI Trans. Intell. Syst., 2016, 6 (6), pp. 788–796.
[3] Silver, D., Huang, A., Maddison, C.J., et al.: ‘Mastering the game of go with deep neural networks and tree search’, Nature, 2016, 529 (7587), pp. 484–489.
[4] Bench-Capon, T.J.M., Dunne, P.E.: ‘Argumentation in artificial intelligence’, Artif. Intell., 2007, 171, (10), pp. 619–641.
[5] Samuel, A.L.: ‘Some studies in machine learning using the game of checkers’, in ‘Computers & thought’ (MIT Press, New York, USA, 1995), pp. 206–226.
[6] An, M.M., Li, S.Q.: ‘An evaluation algorithm of checkers’, J. Beijing Inf. Sci. Technol. Univ., 2017, 32, (2), pp. 85–88.
[7] Li, Y.R., Liu, X.P., Wang, Y.J., et al.: ‘A systematic research about the computer game of checkers based on the decision-making of AI’. Control and Decision Conf., Qingdao, China, 2015, pp. 6437–6442.
[8] Liu, X.P., Li, Y.R.: ‘The application study on improved distributed genetic algorithm in computer games’, Trans. Beijing Inst. Technol., 2017, 37, (10), pp. 1019–1023.
[9] Zheng, C.S., Jia, L.J., Quan, H., et al.: ‘A good role. However, the analysis also found that using the schnook node tree search’, Nature, 2016, 529 (7587), pp. 484–489.
[10] Bench-Capon, T.J.M., Dunne, P.E.: ‘Argumentation in artificial intelligence’, Artif. Intell., 2007, 171, (10), pp. 619–641.
[11] Samuel, A.L.: ‘Some studies in machine learning using the game of checkers’, in ‘Computers & thought’ (MIT Press, New York, USA, 1995), pp. 206–226.
[12] An, M.M., Li, S.Q.: ‘An evaluation algorithm of checkers’, J. Beijing Inf. Sci. Technol. Univ., 2017, 32, (2), pp. 85–88.
[13] Li, Y.R., Liu, X.P., Zhang, H., et al.: ‘Study on system modeling and optimization of static evaluation based on the computer game of draughts’. Control and Decision Conf., Jinan, China, 2016, pp. 4283–4287.
[14] Lincke, T.R.: ‘Strategies for the automatic construction of opening books’, in ‘Revised papers from the second international conference on computers and games’ (Springer, Berlin, Heidelberg, Germany, 2000), pp. 74–86.
[15] Karapetyan, A., Lorentz, R.J.: ‘Generating an opening book for amazons’, in ‘Computers and games’ (Springer, Berlin, Heidelberg, Germany, 2004), pp. 161–174.
[16] Schaeffer, J., Culberson, J., Tredo, N., et al.: ‘A world championship caliber checkers program’, Artif. Intell., 1992, 53, (2–3), pp. 273–289.
[17] Cazenave, T., Liu, J., Teytaud, F., et al.: ‘Learning opening books in partially observable games: using random seeds in Phantom Go’. Computational Intelligence and Games, Santorini Island, Greece, 2016, pp. 1–7.
[18] Steinmetz, E.S.: ‘Computer go and monte carlo tree search: opening book and parallel solutions’, University of Minnesota, 2016.
[16] Wei, Q.G., Wang, J., Xu, X.H., et al.: ‘A study and design of opening-book of computer Chinese chess’, CAAI Trans. Intell. Syst., 2007, 2, (1), pp. 85–89

[17] Li, L., Huang, H., Deng, L.: ‘Dynamic opening-book in computer games’. Control and Decision Conf., Mianyang, China, 2011, pp. 3451–3454

[18] Hong, K., Kim, J., Lee, H., et al.: ‘Data-driven computer go based on hadoop’. Advanced Information Networking and Applications Workshops (WAINA), Victoria, Canada, 2014, pp. 347–351

[19] Steinmetz, E.S., Gini, M.L.: ‘Mining expert play to guide Monte Carlo search in the opening moves of go’. IJCAI, Buenos Aires, Argentina, 2015, pp. 801–807

[20] Michalowski, M., Boddy M., S., Neilsen, M.: ‘Bayesian learning of generalized board positions for improved move prediction in computer go’. AAAI, 2011

[21] Buro, M.: ‘Toward opening book learning’, ICCA J., 1999, 22, (2), pp. 98–102

[22] Donninger, C., Lorenz, U.: ‘Innovative opening-book handling’, in ‘Advances in computer games’ (Springer, Berlin Heidelberg, Germany, 2005), pp. 1–10

[23] Mullins, J., Drake, P.: ‘Using human knowledge to improve opening strategy in computer go’. Int. Conf. Artificial Intelligence, Atlanta, GA, USA, 2010, pp. 730–734

[24] Audouard, P., Chaslot, G., Hoock J., B., et al.: ‘Grid coevolution for adaptive simulations: application to the building of opening books in the game of go’, in ‘Workshops on applications of evolutionary computation’ (Springer, Berlin, Heidelberg, 2009), pp. 323–332

[25] Gaudel, R., Hoock J., B., Pérez, J., et al.: ‘A principled method for exploiting opening books’, in ‘International conference on computers and games’ (Springer, Berlin, Heidelberg, 2010), pp. 136–144

[26] Zobrist, A.L.: ‘A new hashing method with application for game playing’, ICCA J., 1970, 13, (2), pp. 69–73

[27] Cheng, L., Yajie, W., Fei, L., et al.: ‘Debugging computer game program using Zobrist hashing’. Control and Decision Conf., Taiyuan, China, 2012, pp. 1514–1517

[28] Huizhan, L., Chenjun, X., Hongye, L., et al.: ‘Hash table in Chinese chess’. Control and Decision Conf., Taiyuan, China, 2012, pp. 3286–3291