

The Framework Design of Intelligent Checkers

Biao Wang¹, Lijuan Jia^{2(✉)}, He Quan¹, and Changsong Zheng¹

¹ School of Automation, Harbin University of Science and Technology,
Harbin 150080, Heilongjiang, China

² School of Computer Science and Technology, Harbin University of Science and Technology,
Harbin 150080, Heilongjiang, China
515333669@qq.com

Abstract. As one of the research direction in the field of artificial intelligence, the computer game has a great practical significance. Checkers become a represent of complete knowledge game that because of it has few chess pieces, and moves easily, So it can reflect the thought of the computer game better. First, the recursive algorithm of artificial intelligence can achieve the independent moves and checkmates in draughts game program. Second, the optimization of evaluation function makes the program choose beneficial move ways. Third, the game tree can improve the level of computer in-depth thinking, simplify the search scope, and save computer thinking time by pruning, which improve the level of chess program. In this paper, overall framework of intelligent checkers is built from board representation, game tree, valuation optimization. Based on the accumulation of experience, algorithm upgrade, depth analysis of existing framework, then we give a more optimized system framework and then strengthen the computer game level.

Keywords: Computer game · Checkers · Framework design · Game tree

1 Introduction

The national contest of college students' computer game [1] has been held four sessions since 2011, aimed at regarding high resistance [2] board games as the research carrier. We can simulate our interest in learning program and improve the team cooperation ability by the contest. Checkers as one of the computer game events at the competition, the oldest and most popular one of the puzzles all over the world. The competition abroad has been held since the 1970s, the old age, students and youth who all take part in the competition, The ability is more and more strong and the compete increasingly fierce; From the beginning of this century, the domestic hosts game championship, the domestic has carried out college students' computer game series since recent years, which obtained the response of the college students amateur. There are few of quality strong university in the contest. Computer game reflects comprehensive knowledge level of the student's team.

Our framework of checkers to mainly optimize the combination of recursive algorithm, game tree, ALPHA BETA algorithm and algorithm of mini-max [3]; then we optimize these algorithms in game competition. We divide moves of chess into three

categories of kill, sacrifice and layout. So chess can judge the situation in the corresponding position and the framework is intelligent.

In this paper, the first part of the pieces introduces how chess move, the use of algorithm and the situation of optimization; in the second part, we improve the framework by the experience in competition; finally, we make a summary.

2 Initial Design Checkers Framework

2.1 Understanding of Checkers

The international rules of 10*10 draughts board are the same black and white checkerboard, the board is in the middle of the both side, the right corner of each player is blank grid. As shown in Fig. 1.

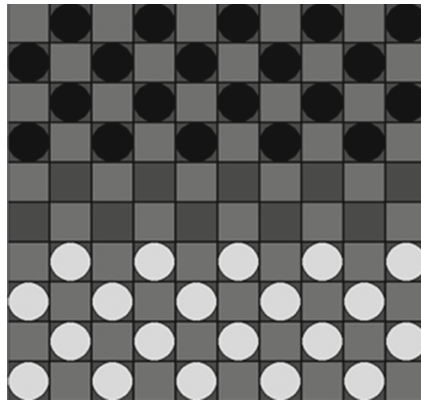


Fig. 1. Checkers 100

This paper involves several jargons:

Kill: For a given position, after we win the right of moving the chess, if the chess can jump to eat, the movement of maximum number of killed chess is regarded as optimal killed.

Sacrifice: For a given position, after we win the right of moving the chess and all movements, the movement of killed chess is regarded as sacrifice.

Layout: For a given position, after we win the right of moving the chess and all movements, the movement of being not killed chess is regarded as layout.

If we could kill chess, we would kill the chess in checker rules. And we are based on the movement of maximum number of killed chess. Therefore, so many factors should be considered in the algorithm design. Among them, kill is to find out the several ways to kill most chess and select the optimal way to move [4]. Sacrifice forces the opponent to kill our chess; furthermore, we can kill more chess of opponent to make opponent loss more chesses. This is especially important, and may even turn around the situation. Therefore, we use game tree [5] on sacrifice algorithm method, the depth of exploration

can achieve greater harm to the match; layout is launched when we have yet fight. Because the solid foundation plays the significant role in the future offensive, in this way can blow to the opponent. Rather, we are in the troubled [6].

2.2 Board Representations

There are two commonly used expressions of the board. One is two-dimensional array board, the other is bit boards [7]. In which two dimensional array boards sees simple and intuitive, so it's easy to find a potential problem and shorten the time of debugging. And we use the two-dimensional array board frequently. Surely, considering the efficiency issues, bit board will have obvious advantages. This paper adapts the two-dimensional array board.

As shown in Fig. 1, we can use 10*10 integer two-dimensional array to represent the 10*10 squares, where 0 means no pieces, 1 means black chess, 2 means king of black; -1 means white chess; -2 means white king. The corresponding numerical can real-time express the board.

2.3 Way to Generate

When we get the right of move, we choose the optimal movement in all movements. Here introduce the concept of generator moves [8], which generate all reasonable way to move. As shown in Fig. 2.

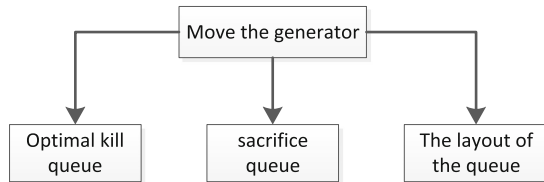


Fig. 2. Move the generator

All reasonable way is divided into three categories: kill, sacrifice and layout. And we take them into the corresponding queue, we can analyze the content of the three queue. Obviously, when all three queues is NULL, which means no chess can go or pieces. According to the rules, we will be jailed for check. And we can see, optimal kill moves in the queue more priority than sacrifice and layout. That is to say, kill queue is not null, sacrifice and layout queen has no means. Sacrifice and layout queue is the same level.

2.4 Game Tree Implementation

Game tree can accurately record the game of two people or a program process. So the game tree algorithm is mainly used in the process of the sacrifice queue, in the sacrifice queue to open game tree. In case of sacrifice, we would simulate it, the opponent will

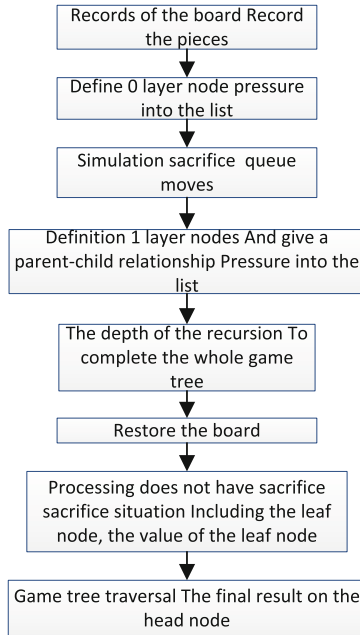


Fig. 3. Depth of 1 game tree creation process

jump to eat, we also simulate it, and then we observe the situation in the following rounds. Now we suppose the depth of game tree steps is 1, which means, from the beginning of sacrifice to the right of having chess no longer has the advantage kill. At the same time, we can record the initial board; we also press the final board into leaf nodes of tree. Finally, after the game tree traversal, we can get the best results. Here, evaluate good standard is to record difference of the other pieces is killed and killed our pieces, which if greater than zero, said the sacrifice is advantage; If it is equal to zero, same said casualties; If less than zero, say we are cheated.

Game tree information need to simulate all processes of optimal kill the queue for ensuring the completeness of the algorithm. All levels use the algorithm of the max value relationship from leaf nodes back to head node. Finally, the best information will be stored in the head nodes. Step depth is assumed to be 1. If step depth is 2 that means: from first one sacrifice to the end of kill each other, we still have the right to play chess (Kill queue is null), and sacrifice queue is not null. Furthermore, we move the chess basing on the best information. After a program test: In the normal computer game within the time allowed, a computer can further step 4; Node numbers is more than 18000.

The program searches the optimal solution with ALPHA - BETA search [9], which reduces some node traversal and saves time. Furthermore, program is able to deal data with a deeper level.

The following steps give a depth of 1 game tree creation process. And the process showed in Fig. 4.

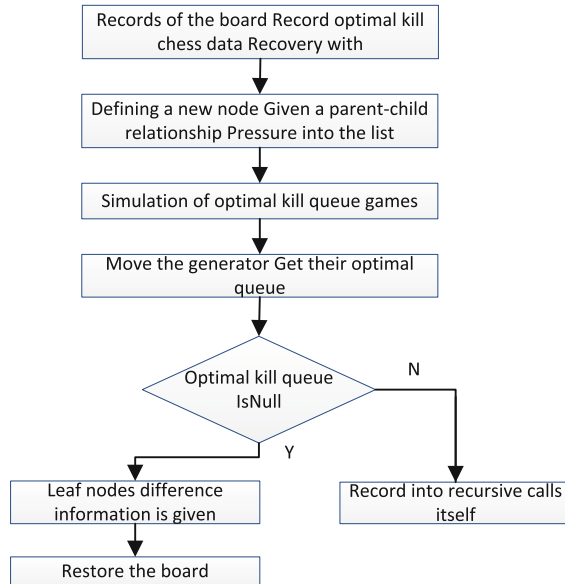


Fig. 4. Recursive procedure

When considering the steps in the depth of 2, just sacrifice queue in the case of in game tree. Actually the layout queue may also have a different way, so sacrifice in step 2 is not the only way. We can find the best optimal movement in step 1 and step 2 sacrifices. That is to say, if step 2 sacrifice situations were bad, we could choose best in step 1; in step 2 we can choose sacrifice and moves in layout. So we can optimize for depth use a few steps at the same use a few game tree. And we find the optimal solution in the game tree.

2.5 Evaluation Function

For computer games, a lot of kind chess in a limited period of time is can't be totally search. Although checkers branch is less, we still can't search totally [10], This valuation function is particularly important.

The effect of value function is given a board to evaluate, determine the general trend. Furthermore, we decide how to make the best judgment to moves. Valuation function mainly used optimal selection and layout of the games.

There are many valuation characteristics. Such as the number of king, ordinary pieces quantity, suppressing the other pieces, formation (Including chain, triangle [11], and so on), distribution of pieces, optimal number of pieces, sacrifice piece count, and so on [11]. In many assessment characteristics, we must focus on the number of the pieces. This program can save pieces in order to win; besides, many features need to integrate together. This involves many features of the weight problem. We can roughly determine the weight of each feature by experience. It's important to note that not all features of

proportional relations is invariable; in the opening, mid-game, and endgame, the weight of valuation is not the same. So valuation function [12] needs to adjustment dynamically.

We still need be aware of is the king of valuation, the result of the king being especially precious. The action and damage of king are very powerful. So we try to protect our king and make full use of king. Of course, we can give the king set high proportion. So the game tree part of sacrifice, whether sacrifice into a king or kill the opponent's king, it works.

2.6 Primary Overall Frame Design

The above to comprehensive understanding and analysis of the checkers, completed application framework is shown in Fig. 5.

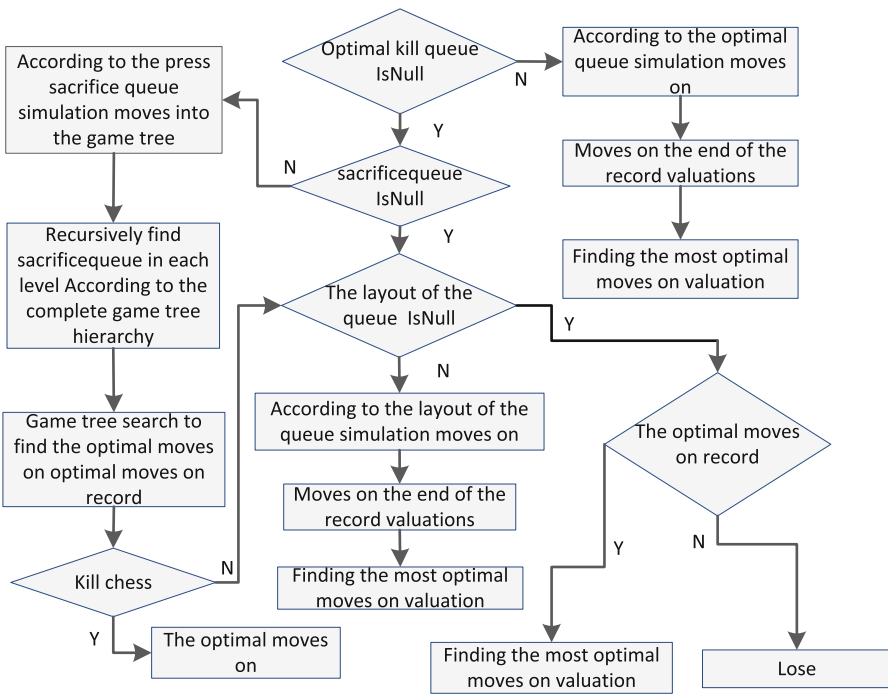


Fig. 5. Primary overall frame design

From Fig. 5 can be seen kill movement and layout way are based on a valuation function to determine. Sacrifice part with game tree, when on the attack can achieve step 4, layer can achieve the higher than 14th floor (With the algorithm optimization and upgrading, you can reach a higher level). Not only introduced in the valuation is optimally killed, but also introduced sacrificed. If the step was 3, while the rest of that, we could prevent continuous sacrifice within each other in three steps. To some extent, we ensure number of chess. So in general, the framework of application level has been very good at chess, we have also made very good result. Of course, as we further study and

explore, the framework can also optimize [13] and upgrade for achieving a better game level.

2.7 Analysis on the Running Status of Program Design Under the Primary Framework

The statistics of the program running time in the primary framework are shown in Table 1.

Table 1. The statistics of program running time

Options	Level	Average time (ms)
Attack	1	300
	2	4223
	3	9101
	4	15306
	5	45542
Defense	1	654
	2	5321
	3	12056
	4	32678

Comprehensive program running time and game requirements, the attack is divided into 4 layers, defense, 3 layers. Due to exist many possible to layout, so the defensive time required more than attack. In the current level, this had exceeded the amateur level. The framework of the game in the national has awarded two awards in 2015 computer game competition, the group, third. The game framework is simple, but the attack ability. Of course, this design of framework also exists some problems, after this game, aiming at design; we have made a further optimization.

3 Optimal Design of Checkers Framework

3.1 To Improve Analysis

Framework of game tree works in sacrifice queue and the optimal kill and layout part is weaker; almost no combined with optimal kill and layout, which may ignore some important situation. Furthermore, we can make errors or miss some good opportunity. Sacrifice part moves or not is not determined by the situation, but to judge whether the whole process is beneficial, if we could kill the more chess of opponent. Of course, It is no problem that at point of view in the number of pieces is preferred. However, if more than kill a chess pieces, make the situation is bad for us and formations, this strategy may not be wise.

So come to the conclusion: first, the game tree lists all possible situation; second, the game tree leaf nodes as the final judge are achieved by valuation function.

3.2 Optimize Moves

From the above analysis, if the game tree has all possible situations, so sacrifice and layout can be considered together. Optimization moves generator is shown in Fig. 6.

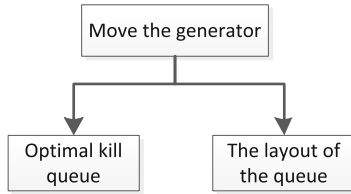


Fig. 6. Optimization of the move generator

The optimized way generator only superior to kill and layout. That is to say, in view of a parent node, his all subsequent nodes are either kill queue all elements or queue all elements layout [14]. On the one hand, this situation conforms to the rules; on the other hand, sacrifice thoughts are included in this situation. In game tree have the thought of sacrifice and coordinate all the circumstances.

3.3 Valuation Function Research

Leaf nodes in game tree are determined by valuation function, so the valuation function is trusted to reflect the actual status of board. If valuation function can't bear the responsibility, making a wrong decision, valuation function would have an impact on the level of chess.

So the valuation functions that need to add more comprehensive content, the details need further consideration. The most important is the ability to grasp the situation of the state overall situation being good for us. On the basis of the analysis more details would have greater significance. If the valuation function not do so, under the framework of the program is not always to beyond the previous frame or even worse, this is also the significance of the existence of the front frame. If the valuation function is not complete and accurate grasping chess, it is better to analysis kill a few simple pieces in sacrifice.

Judge to the details situation involve multiple factors. Emphasis weight is also different in the different stages. Of course, it is important to reservation power of pieces; and according to the actual situation, to some extent, in order to better determine relationship to piece power opponent with me, there is a certain equivalence relation between king and normal.

If the search is a fixed depth, horizon effect will appear [15]. That is to say, when the search is up to a fixed depth, valuation function to evaluate leaf node. If the leaf node in the stage of frequent to change pieces, it is clear that this assessment is not accurate. Because even if the situation now in our favor, but the other party can immediately jump to eat our many pieces; obviously we would get into a passive or failure. There are some methods solve the horizon effect, we can use the valuation function enhancement processing, let the valuation function can analyze a simple potential changes in the piece

power. Such as chess can be analyzed in a exchange state, Killed state or be sacrifice state, based on the early primary system framework [16], detection and analysis of the several situation are easy, and the time complexity is not high also.

There are plenty of features for valuation function needs analysis, will be accessible to the king, and so on, for example, only the more features will be more able to reflect the real chessboard, this chess level will be stronger.

3.4 Optimization Design of Overall Framework

Based on the above understanding and analysis of the checkers, optimization program framework is shown in Fig. 7.

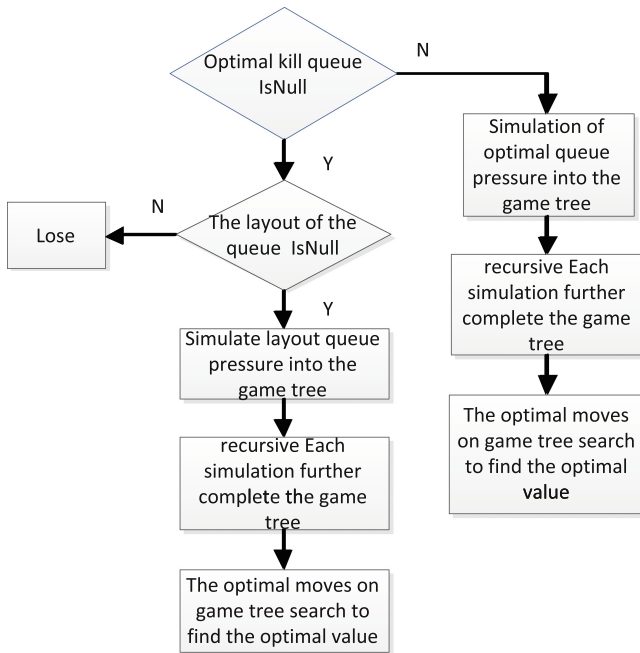


Fig. 7. Optimization framework

Seen from the Fig. 7 game tree is the entire generator moves in a reasonable way, the building of the game tree is similar to the early design process. Under the framework of game tree each layer contains all the possible moves, the final assessment of the leaf nodes are performed by the valuation function. Finally, the optimal solution is chosen.

Obviously, the design framework of coordination in Fig. 7 is better than Fig. 5, besides, the design framework of coordination in Fig. 6 is more perfect. In general, framework in Fig. 6 is more perfect for checkers, piece level also have larger development space and better compatibility [17]. With optimization algorithm and new exploration of the algorithm, chess level will also increase quickly. For the more characteristic parameters involved in the process of moves, artificial neural network and machine

learning [18] can be applied to optimize the evaluation function. Over rivals and debug member of accumulating good layout information and moves on strategy, the evaluation function can adjust themselves.

3.5 Analysis of Program Operation Status Under the Optimization Framework

Compared to the framework of basic game, the optimal game framework has a further improvement, the chess of attack and defensive are unified, both of the game tree, the depth of the valuation, get the best plan. Combined with optimization scheme of alpha-beta pruning, permutation table, history heuristic, to reduce the amount of calculation of the game tree, the time complexity is further reduced. Statistics of program running time as shown in Table 2.

Table 2. Statistics of program running time

Game level	Average time (ms)	Finding the table (times)
1	15	10
2	47	50
3	125	200
4	200	450
5	843	1167
6	1752	4523
7	4065	11067
8	16546	52304
9	52663	140009

Visibly, the optimization of framework of the game has a very good optimization; the level of the game has also been greatly improved. Currently taking into account the limitations of the time, the maximum depth of the search is 8 layers, the level of chess has very strong.

4 Conclusion

This team in the checkers framework designed in the fourth session of Chinese college students in computer game series checkers 100, we achieved The second prize of good grades and accumulated a lot of experience, learning a lot of excellent algorithm is benefit for further exploration, and in the late game for the framework has been optimized and improved, so that the level of chess, has a significant improvement. After constant improvement, in the field of checkers 100, we must make greater achievement in the field of checkers.

References

1. Wang, J., Xu, X.: The frontier of artificial intelligence—the national contest of college students' computer game. *Comput. Educ., Beijing* **7**, 14–18 (2012)
2. Wang, X.: *PC Game Programming (Man-Machine Game)*. Chongqing University Press, Chongqing (2002)
3. Yue, W.: A high-performance checkers engine design. *Comput. Prog. Skills Maintenance* **32**(02), 63–68 (2014)
4. Yajing, L.: Main technical analysis of six children of a game of chess computer. *Comput. Knowl. Technol.* **7**(10), 2310–2312 (2011)
5. Russell, S.J.: *Artificial intelligence: a modern approach (3)*. Jianping Yin. Qinghua University Press, Beijing (2013)
6. Sea, K., Haykin, S.: *neural networks and machine learning*. Shen FuRao. mechanical industry publishing house, Beijing (2011)
7. Yajing, L.: Main technical analysis of six children of a game of chess computer. *Comput. Knowl. Technol.* **7**(10), 2310–2312 (2011)
8. Zhang, C., Liu, H.: Game tree heuristic search of alpha beta pruning technology research. *Comput. Eng. Appl.* **16**, 54–56 (2008)
9. Zhang, M., Li, F.: A new game tree search methods. *J. Shandong Univ. (Eng.)* **6**, 1–7 (2009)
10. Shafei, N., van Breugel, F.: Towards model checking of computer games with Java PathFinder. *IEEE Press* **13**, 25–29 (2013)
11. Shu, K., Hu, F.: Chinese chess computer game engine improvement. *J. Artif. Intell.* **10**, 39–42 (2009)
12. Wang, J., Wang, T.: Chinese chess computer game system evaluation function of adaptive genetic algorithm. *J. Northeastern Univ.* **10**, 949–953 (2005)
13. Zhou, W., Wang, Y., Ma, Q.: Changes of the situation to seek advantage game. *J. Syst. Simul.* **17**, 5–9 (2008)
14. Bottcher, N., Martinez, H.P., Serafin, S.: Procedural audio in computer games using motion controllers: an evaluation on the effect and perception. *Hindawi Publishing Corp.* **11**, 94–97 (2013)
15. Xu, C., Ma, Z., Xu, X., Li, X.: Machine game oriented real-time difference study. *J. Comput. Sci.* **8**, 219–223 (2010)
16. Yu, W., Yang, X.: Intellectual games interpretation characteristics and development prospects analysis. *J. Shandong Sports Institute.* **27**, 5–10 (2011)
17. Gao, Y., Chen, S., Liu, X.: Reinforcement learning research review. *Acta Automatica.* **1**, 87–97 (2004)
18. Wallace, S.A., Russell, I., Markov, Z.: Integrating games and machine learning in the undergraduate computer science classroom. *ACM.* **8**, 231–236 (2008)