

Efficient Time Control in Chinese Chess with Alternating Search

Xun Jiqing^{†1} Reijer Grimbergen^{†2}

Abstract

Time control is an important strategy problem in games. With a good strategy, time can be used more efficiently, using more time for more important decisions. In computer games, a good strategy can provide more time for certain moves, thus allowing deeper search which results in better moves. Furthermore, a good strategy should be adaptive so it can be used in different contexts. In this paper, we propose a new method to improve the efficiency of time control by adjusting the search depth according to the game stage and the time that is left. The experimental results show that the proposed method may be a suitable way to use time more efficiently.

1. Introduction

Time control is a mechanism in the tournament play of almost all two-player board games so that each round of the match can finish in a timely way and the tournament can proceed. Time controls are typically enforced by means of a game clock. Time pressure is the situation of having very little time on a player's clock to complete the remaining moves. Making obvious decisions quickly, thus saving more time for more difficult moves is a strategy that is used by most players.

The aim of our research is to build a game program that is able to use time efficiently in the same way that proficient game players do. In general, this requires an assessment of the difficulty of the position and needs to take into account how much time is left on the clock.

As a first step, we will test a simple time control strategy in Chinese chess to assess the feasibility of our approach. Chinese chess is a popular two-player board game where time control is often used. In this paper, we will introduce a time control strategy in Chinese chess based on the stage of the game and the time remaining.

2. Proposed method

2.1 Previous work

To use time efficiently in Chinese chess, one of the methods [1] is to use different algorithms for deciding a move according to the time left. In general, there are three situations of time control: plenty, sufficient and lacking (time trouble). According to the time left, it is possible to adjust search depth and search method. At this point, it is not clear how a human players is making the decision whether time is plenty, sufficient or lacking. Therefore, we have decided that search depth should depend on the game stage instead of the time left.

2.2 Game stage and search depth

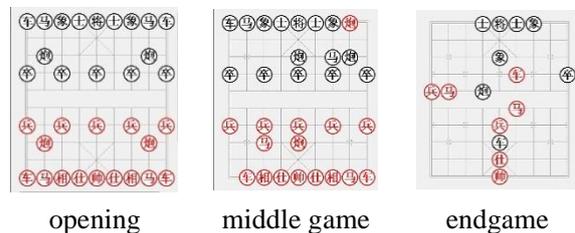


Figure 1. Three stages of Chinese chess

In Figure 1, examples of the three stages in Chinese chess are given. These are called opening, middle game and endgame [2]. The difference between these stages has been defined in Chinese chess literature as follows [3]:

- When a piece has been captured by the opponent side, the game is no longer in the opening stage, but has changed to the middle game.
- In Chinese chess, the pieces called Chariot, Cannon and Horse are considered aggressive pieces. When the total number of aggressive pieces is smaller than six, the game has entered the endgame stage.

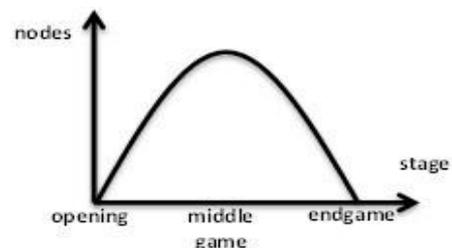


Figure 2: The likely relationship between nodes and stage[1]

Figure 2 shows the likely relationship between nodes and game stage in Chinese chess. In the opening, the

^{†1} Tokyo University of Technology, Graduate School of Bionics, Computer and Media Sciences, Computer Science Program

^{†2} Tokyo University of Technology, School of Computer Science

number of pieces that can move and the amount of freedom they have is limited, resulting in a smaller number of possible moves. In the endgame, the number of remaining pieces on the board is smaller, also reducing the number of possible moves. Therefore, in the middle game stage, the number of search nodes is larger than in the opening and endgame stage.

Our proposed time control strategy is as follows. In the opening stage, increasing the search depth can lay a good foundation of the game. In the middle game stage, the number of nodes is very large. Increasing the depth will take a lot of time. In the endgame stage, increasing the search depth can reduce the number of mistakes and improve the possibility of victory.

3. Experiment

Test environment:

- CPU: Intel Core2 Duo CPU P8600
- RAM: 4GB
- System Type: 32bit
- Programming Language: C++

The original program uses the common NegaScout search strategy[4]. The proposed method also used same search engine but it adjusts the search depth according to the game stage and time left. In the experiment, we have set the time control to ten minutes for each program. The original program played the program with our proposed method for 1000 games.

4. Experiment results

Table 1: Moves and time consumption between the original method (O) and proposed method (P)

stage	Moves(O)	Moves(P)	Time(O)	Time(P)
opening	12.7	12.3	0.56s	0.58s
middle game	32.7	32.1	2.5s	7.68s
end game	35.4	32.8	0.87s	1.56s

Table 1 shows the four important values recorded in the test:

- Moves(O): average number of moves for each game stage of the original method
- Moves(P): average number of moves for each game stage of the proposed method
- Time(O): average time consumption for each move per game stage of the original method
- Time(P): average time consumption for each move per game stage of the proposed method

The results of the date analysis are summarized in Figure 3 and Figure 4.

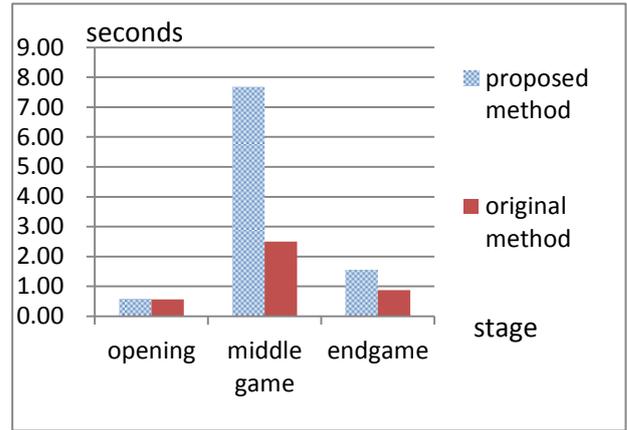


Figure 3: Time consumption between proposed method and original method in three stages

From Figure 3, it is clear that in the middle game and endgame stage the proposed method takes much more time than the original method.

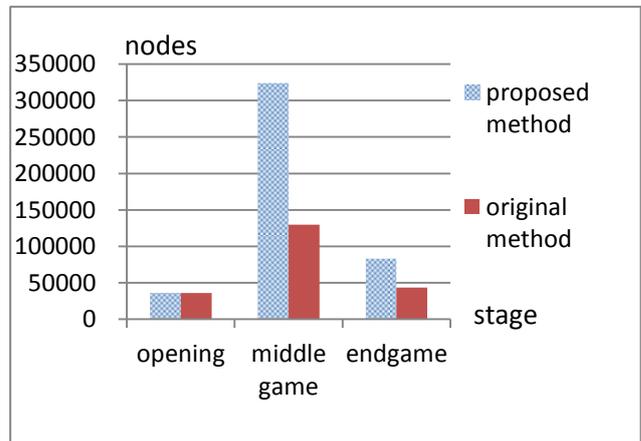


Figure 4: Search nodes between proposed method and original method in three stages

In Figure 4, we can see that proposed method searches many more nodes than the original method in the middle game and endgame stage.

As far as our feasibility test is concerned, the most important question is whether the proposed method was able to use time more efficiently than the original program. In the test we set the time control to ten minutes for each program. In Table 1 we have recorded the time consumption. This time consumption data is summarized in Figure 5. On average, the proposed method left less time on the clock than the original program. This means that with the proposed method it may be possible to use time more efficiently by adjusting the search depth.

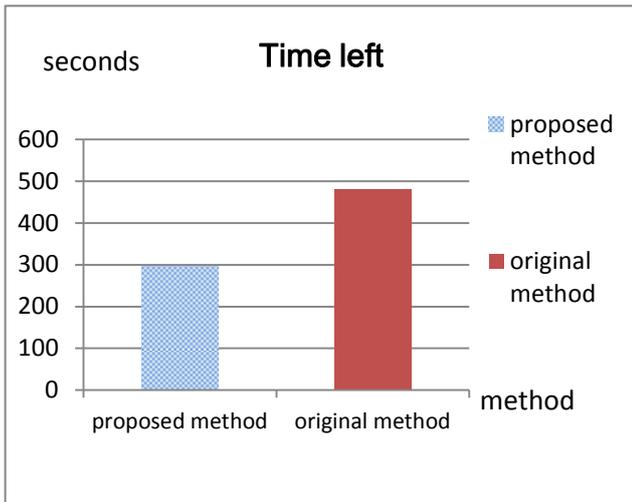


Figure 5: Time left between proposed method and original method

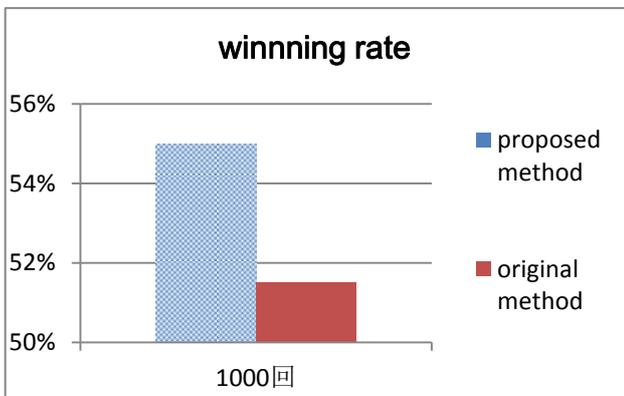


Figure 6: Winning rate between proposed method and original method

Finally, in Figure 6, we can see that the proposed method has a better winning rate than the original method in the test. This means that using time more efficiently provides more time for deeper search, leading to an improvement of playing strength.

5. Conclusion

This research studies how to use time more efficiently. A simple method based on changing search depth in Chinese chess according to game stage and time left was able to use time more efficiently, leading to an improvement of playing strength. Though simple, it is likely that the proposed method is a way to improve the efficiency of time control.

As future work, we will investigate different ways of changing the search based on time left. Specifically, we would like to try changing the search in ways that are closer to the way human players try to improve the efficiency of time use. Furthermore, there are a number of search methods specific to Chinese chess[5] we would like to compare in the future.

Reference

- [1] XU Chang-ming, NAN Xiao-fei, WAN GJ iao, XU Xin-he, "Adaptive time allocation strategy in computer game of Chinese Chess," CAAI Transactions on Intelligent Systems, Vol. 1, No. 2-10, 2006.
- [2] "Chinese Chess Encyclopedia," URL:<http://www.xqbase.com>
- [3] Ling Cao. The Great Master Teaches You Learning Chinese Chess. Jilin: Jilin Publishing Group, 2011.
- [4] Xiaochun Wang. PC Game Programming. Chongqing: Chongqing University Press, 2002.
- [5] Xinhe Xu, Jiao Wang: "Key Technologies Analysis of Chinese Chess Computer Game", 2005.