Title	Shape-Keeping Technique and Its Application to Checkmate Problem Composition
Author(s)	Ishitobi, Taichi; Cincotti, Alessandro; Iida, Hiroyuki
Citation	Artificial Intelligence and Game Aesthetics: the 2013 AIIDE Workshop (WS-13-19): 7-10
Issue Date	2013-10-14
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/11606
Rights	Copyright (C) 2013 AAAI Press. Taichi Ishitobi, Alessandro Cincotti, Hiroyuki Iida, Artificial Intelligence and Game Aesthetics: the 2013 AIIDE Workshop (WS-13-19), 2013, 7-10.
Description	

Shape-Keeping Technique and Its Application to Checkmate Problem Composition

Taichi Ishitobi, Alessandro Cincotti, Hiroyuki Iida
School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan

Abstract

The checkmate problem in Shogi (Japanese Chess) is a puzzle within the game itself. These puzzles have enjoyed a long play and have been the subject of centuries of analysis. The subject of this research is defining the aesthetic criteria of great Shogi problems, and finding new methods for composing interesting checkmate problems in Shogi. First we examine the results of previous studies of aesthetics in Shogi checkmate problems. For this purpose, we focus on the Proof Number Search algorithm and record the data while solving checkmate problems. We analyzed these data and we calculated the proof number related to the evaluation of the checkmate problem. Good checkmate problems have large proof numbers. Next, we present a new technique for automatic composition of checkmate problems in Shogi. This technique uses already existing checkmate problems in Shogi and develops them further. Finally, we can compose new checkmate problems which have bigger proof numbers than original ones. This work is not yet sufficient unto itself.

Introduction

Shogi is an ancient Japanese game like Chess, but empirically and aesthetically more complex (2002 Iida), (2012 Iqbal). The checkmate problems in Shogi (Tsume-Shogi) are mating problems which can be seen as a subdomain of Shogi. There are a few noticeable differences and similarities between Shogi and Western Chess. In Shogi a player can reuse (drop) pieces which have been captured from the opponent. The checkmate problem in Shogi commences from devised positions, like as with checkmate problems in western Chess. The objective of the mating puzzle is to checkmate the enemy king. Differently from checkmate problem in western Chess, in Shogi the player must check the king continuously. For more detailed descriptions about Shogi, please refer to (2002 Iida).

The checkmate problem in Shogi has a long history and a rich tradition. Consequently, the aesthetics of the checkmate problems in Shogi have long been the subject of study, and good checkmate problems in Shogi have been collected in Shogi books a few times like Shogi-Zuko. Many contests for checkmate problems have been held, and all the best problems have been decided by professional Shogi players and popular votes.

What are the aesthetic criteria of checkmate problems in Shogi? Some results are presented in (1994 Koyama). Successively, researchers have found techniques for composing good checkmate problem in Shogi. For example, it is the position of King, space around a piece, etc. Therefore, it is difficult to combine and propose a numerical value. In this paper, we will show the values that are related to the technique for composing good checkmate problems, and the aesthetic of these problems.

Proof Number Search

The domain of checkmate problems in Shogi is popular in Japan, and is a good application for the AND/OR tree search. The AND/OR tree is a type of game tree where the nodes have only three possible values: true, false and unknown. The checkmate problem is an AND/OR tree structure (checkmate, cannot checkmate and unknown). The average branching factor of checkmate problems in Shogi is about 5 (2001 Seo). Naturally, this number is much smaller than that of whole-game Shogi. Problems with solutions of more than 17 steps cannot be solved by simple brute-force methods like depth-first minimax enhanced with $\alpha - \beta$ only (1975 Knuth).

To solve checkmate problems effectively, we turned our attention to the Proof Number Search (PNS) as proposed by Allis et al. (1994 Allis). Using proof numbers it is possible to identify the potential solution tree (PST) for which the probability of becoming a solution tree is the highest among all PST's. A PST is defined as a search tree with one continuation at every max node and all continuations at min nodes, of which the leaf nodes are either won nodes for the max player or unexpanded nodes. It may become a solution tree by expanding unsolved leaf nodes and proving that they all lead to a win for the max player. The PNS produces two special values called the proof number and the disproof number.

The proof number for an AND/OR tree search is the minimum number of unsolved leaf nodes that need to be solved in order to win in the root. Similarly, the disproof number is the minimum number of unsolved leaf nodes that need to be solved in order to lose in the root. When the terminal node is N, proof number is pn, disproof number is dn, and these values are defined as follows:

$$(N.pn, N.dn) = \begin{cases} (0, \infty) & \text{if N is True} \\ (\infty, 0) & \text{if N is False} \\ (1, 1) & \text{otherwise(N is unknown)} \end{cases}$$

If N is an internal node and C is the set of its children then the value of N is defined as follows:

$$(N.pn, N.dn) = \begin{cases} (\min_{c \in C} c.pn, \sum_{c \in C} c.dn) & \text{if N is OR node} \\ (\sum_{c \in C} c.pn, \min_{c \in C} c.dn) & \text{if N is AND node} \end{cases}$$

The value of the root node can be calculated recursively.

The algorithm of Proof Number Search is shown in the following pseudo code.

Pseudo code 1 Proof Number Search

```
create_root(root)
while proof(root) ≠ 0 and disproof(root) ≠ 0 do
    most_prooving := most_proving_node(root)
    expand_node(most_proving)
    update_proof_numbers(most_proving)
end while
if proof(root) = 0 then
    return ν
else
    return ¬ν
end if
```

The ν shows proving or disproving of the root. The most_proving_node(N) is the function which returns the next search node. This function is defined in the following pseudo code.

Pseudo code 2 most_proving_node(J)

```
while is_an_internal_node(J) do
    if max_node(J) then
        J := leftmost_child_with_equal_proof_number(J)
    else
        J := leftmost_child_with_equal_disproof_number(J)
    end if
end while
return J
```

Thus, the most proving node means the easiest node for proving or disproving. The Proof Number Search always considers the most proving node, because if the terminal node value was decided then the value of many internal nodes can be decided recursively. Therefore, the Proof Number Search can be used to decide the value of the root node by deciding values of other nodes as soon as possible.

Aesthetics of Checkmate Problem in Shogi based on Proof Number Assessment

There are various criteria and considerations which could lead one to understand the aesthetics of the checkmate problems in Shogi. Other studies have defined the aesthetic in terms of the shape of the board or the space around the king and other things, but these are not numerical values and they are difficult to quantify. We approach the aesthetic evaluation by numerical values. Because of this, we focus on the searching value, namely, the proof number and disproof number.

The proof number is the minimum number of positions that must be checked in order to win. We consider the proof number as the measure of problem difficulty (2012 Ishitobi). Disproof number is the minimum number of positions that must be searched in order to lose. We consider the disproof number as the scaler measure of checkmate problems in Shogi. In order to investigate this hypothesis, we performed an experiment and recorded the proof and disproof numbers during the search.

Experiments and Results

Table 1: 7-step contest problems and general problems

Ranking	MAX PN	MAX DN	Iterations
1st	116	3258	235574
2nd	32	3985	58276
General	19	1720	62136

Table 2: 15-step Kanju Award-winning problems and general problems

Year	MAX PN	MAX DN	Iterations
1998	47	42427	1066572
2001	49	36417	1023204
2005	67	117561	4462292
2011	25	62046	741920
General	32	5528	173553

Table 1 and 2 show some values of 7-step contest problems, 15-step Kanju Award-winning problems and general problems. The 7-step contest problems' rankings were decided by the vote of unranked Shogi players. The "Kanju Award" is the most prestigious award for checkmate problems in Shogi, decided by experts. Among the collection of checkmate problems, only 15-step problems are selected. General problems compose the serious player's stock knowledge of the game, and are included for reference but not for study in this experiment. We collected forty 7-step problems and twenty 15-step problems from the books and calculated the average of these data. It is noteworthy that contest and Kanju Award-winning problems are evaluated by experts, but general problems are not.

The maximum proof number and disproof number are the values recorded during the search before checkmate. The proof number is related to the difficulty, therefore the maximum proof number shows the peak of difficulty. The disproof number is related to the scale of checkmate problem in Shogi, therefore the maximum value of disproof number is related to the lowest scale. The number of iterations shows how many times the root node is renewed. We compared these numbers, and the results are shown below.

Among general problems, good checkmate problems have large proof numbers. The 7-step contest problems' maximum proof numbers appear in rank-order. The 15-step Kanju Award-winning problems have larger proof numbers compared to general problems too. Thus, we found that proof number is important for judging the checkmate problem in Shogi. Also, the maximum disproof number is different on problems with the same maximum proof numbers. In our future work, we hope to understand better the meaning of the disproof numbers in this context.

The proof numbers nicely illustrate rankings for 7-step contest problems. However, sometimes contest problems did not line up by maximum proof number. The proof number is related to the aesthetic evaluation, but does not show a direct and proportional relationship–it is not precise.

Additionally, our corrected data is few for calculating statistics data. In future work, we collect more data.

Checkmate Problem Composition

Previous Works

Checkmate problem composition has been studied by Noshita (1996 Noshita) and Hirose (1998 Hirose).

Noshita starts from a random position and tries to fix it in order to arrive at a novel checkmate problem. In this way, he could create 13- to 19-step problems, and sometimes 21-step problems or even more. Noshita's method can produce medium long step problems and it is good for fixing checkmate problems, but composition problems depend on randomness.

Hirose uses a back tracking technique, starting from an already known checkmate problem to create a new checkmate problem. In this way, he can create from 3- to 9-step checkmate problems. Hirose's method can produce interesting checkmate problems, but the computational cost is high and it is difficult to compose greater than 9-step problems.

The Shape Keeping Technique

Like Hirose's method, Shape Keeping Technique creates new checkmate problems using problems previously created by other people. As opposed to Hirose, we keep the same shape and same number of pieces.

We hypothesized that upgrading an original checkmate problem to a better problem might be possible by changing the kind of pieces.

The pseudo code of shape keeping technique is shown in Pseudo code 3.

First, we replace all original pieces with strong pieces except for the king. The possible moves of strong pieces are shown in Figure 1.

Pseudo code 3 Shape Keeping Technique

```
/*P is the original checkmate problem in shogi*/
change_all_pieces_to_strong_pieces(P)
while is_position_changed(P) do
    for every piece p in P (except the king) do
        L := []
        for every possible piece l in shogi do
            P := change_board(P, p, l)
            L := Add(L, DFPN(P))
        end for
        P := change_board(P, p, choose(L))
    end for
end while
```

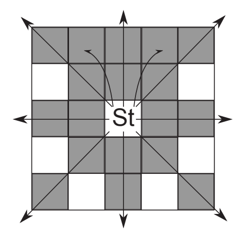

Figure 1: Possible moves of a strong piece

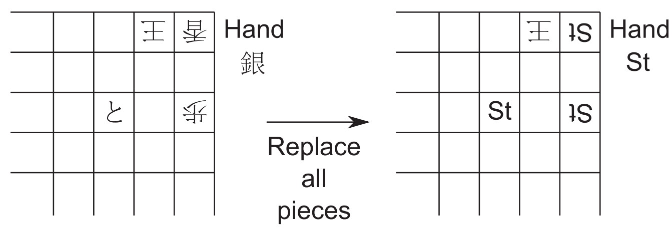

Figure 2: Piece replacement with a strong piece

An example of replacement is shown in Figure 2.

The function change_board(P, p, l) returns a new board where the piece p has been replaced with the piece l. The function Add(L, DFPN(P)) adds to the list L the proof number and the disproof number calculate by DFPN (2002 Nagai) on the position P. The DFPN is more efficient than PNS, but DFPN behaves the same as PNS. Also, the results obtained by PNS and DFPN are not so different, therefore we use DFPN here. If there are strong pieces on the board, then the function choose(L) returns the piece with the highest proof number. Otherwise the function choose(L) returns the piece with the highest disproof number.

To choose always the piece with the highest proof number is not a good strategy because we get new positions where it is too difficult to checkmate. Moreover, in our experience it seems better to not include the pieces with the ratio proof number / disproof number > 1.0.

Some Examples

The first example is shown in Figure 3 and Table 3.

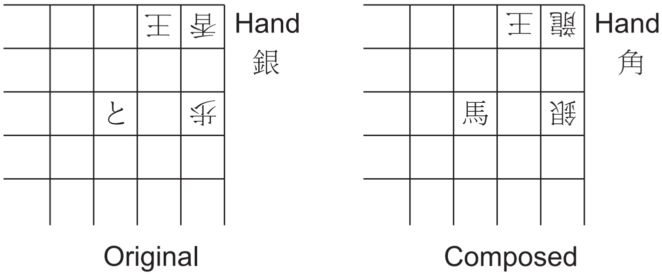

Figure 3: Composed problem derived from a 3-step problem

Table 3: Experimental results for the first example

	MAX PN	MAX DN	Iterations
Original	1	27	42
Composed	13	1951	16979

The original problem is a 3-step problem, but the composed problem needs 11 steps to be solved. The original problem has only weak pieces on the board and the branching factor is low. As result, we can find the solution easily. In contrast, the composed problem has many powerful pieces on the board and the branching factor is high. As shown in Table 3, the proof number, disproof number, and number of iterations of the composed problem are higher than the original problem, but the composed problem does not satisfy checkmate rules in Shogi because it has some possible solutions for checkmate.

The second example is shown in Figure 4 and Table 4.

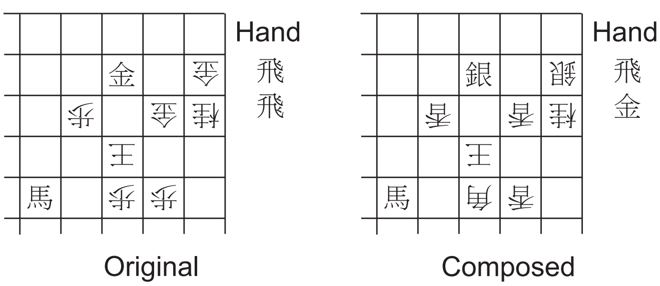

Figure 4: Composed problem derived from a 5-step problem

Table 4: Experimental results for the second example

	MAX PN	MAX DN	Iteration
Original	8	455	2080
Composite	81	31347	1486717

The original problem is a 5-step problem, but the composed problem needs 17 steps to be solved. As shown in Table 4, the proof number, disproof number, and number of iterations of the composed problem are higher than the original problem. This problem has meaningless pieces on the board, but we could find only one answer.

Concluding Remarks

In this paper, we showed two main contributions:

- the relationship between the aesthetic evaluation of Shogi checkmate problems and proof numbers
- the ability to compose new checkmate problems with larger proof numbers using Shape Keeping Technique

In the examples presented, we could compose checkmate problems with higher proof numbers than the original problems. However, we could not always satisfy all the rules of Shogi. We think it is difficult to satisfy checkmate problem rules using only Shape Keeping Technique. Thus, we propose a combination of Shape Keeping Technique and Noshita's way. We are hopeful that in the future, by incorporating Noshita's way, we can fix the checkmate problems to satisfy the checkmate problem rules in Shogi.

References

L. V. Allis, M. van der Meulen, and H. J. van den Herik. Proof-number search. *Artificial Intelligence*, 66(1):91-124, 1994.

K. Noshita. A Note on Algorithmic Generation of Tsume-Shogi Problems. In *Game Programming Workshop in Japan '96*, pages 27-33, Kanagawa, Japan, 1996.

M. Hirose, T. Ito, and H. Matsubara. Automatic Composition of Tsume-Shogi by Reverse Method. *Journal of Japanese Society for Artificial Intelligence*, 13(3):452-460, 1998, in Japanese.

K. Koyama, Y. Kuwano. Quantitative Analysis of Impression on Tsume-shogi (Mating Problems of Japanese Chess), *Information Processing Society of Japan*, 35(11):2338-2346, 1994, in Japanese.

T. Ishitobi, H. Iida. Analysis of Tsume-Shogi Problems using Proof-Number, *Game Programming Workshop in Japan 2012*, pages 163-166, Hakone, Japan, 2012, in Japanese.

H. Iida, M. Sakuta and J. Rollason. Computer Shogi, *Artificial Intelligence*, 134(1-2):121-144, 2002.

A. Nagai. Df-pn Algorithm for Searching AND/OR Trees and Its Applications. PhD thesis, Department of Information Science, University of Tokyo, 2002.

D.E. Knuth, R.W. Moore, An analysis of alpha-beta pruning, *Artificial Intelligence*, 6 (4) (1975) 293-326.

M. Seo, H. Iida, and J. W. H. M. Uiterwijk. The PN*-search algorithm: Application to tsume-shogi. *Artificail Intelligence*, 129(1-2):253-277, 2001.

H. Iida, N. Takeshita and J. Yoshimura. (2002), A metric for entertainment of boardgames: its implication for evolution of chess variants., in Ryohei Nakatsu and Jun'ichi Hoshino, ed., 'IWEC' , Kluwer, pp. 65-72 .

Iqbal, A., van der Heijden, H., Guid, M. and Makhmali, A. (2012). Evaluating the Aesthetics of Endgame Studies: A Computational Model of Human Aesthetic Perception, *IEEE Transactions on Computational Intelligence and AI in Games: Special Issue on Computational Aesthetics in Games*, Vol. 4, No. 3, pp. 178-191.