A-22

The Complexity of King Chase Chess

Tatsuie Tsukiji* Eiji Yamaguchi†

Abstract

Generalized chess is a natural extension of chess to n n board. Fraenkel and Lichtenstein proved that generalized chess is EXPTIME complete. In this paper we exhibit another construction of an EXPTIME complete chess problem on the generalized chessboard. Our construction assumes that the player's king is checkmated by the opponent's just one move and there is no way to be saved. Under this assumption for the player to win he (or she) has to continue checking the opponent's king in his every turn until finally invoking checkmate. Along these attacks, moreover, the opponent is always forced to move his king away from the checked squares. In a word, our construction is so-called a king chase chess problem, and we show that even king chase chess with generalization is EXPTIME complete.

Keywords and phrases: combinatorial game, chess, king chase, EXPTIME complete.

1 Introduction

In this paper we discuss the computational complexity of generalized chess. Storer proved that generalized chess is PSPACE-complete with generalized 50-move draw rule [8]. Fraenkel and Lichtenstein proved that EXPTIME-complete without that rule [3]. In this paper we give another construction of an EXPTIME complete chess problem.

Combinatorial game theory studies computational complexity of two-player perfect-information games where the common winning rule is that the last player who is able to move wins [2]. This paper also takes this simple winning rule into account; No other rules for win or draw, e.g. we do not have the generalized 50-move draw. Of course the rules for moves of the characters are the same as in 8×8 chess: pawns, rooks, bishops, knights, queens and kings move as in the standard chess. There is one king per side played on an $n \times n$ chessboard. Queens are allowed to be at most one per side. White moves first. The number of white and black pawns, rooks, bishops and knights each increases as at most linear of n. More formulation for generalized chess game can be found in [3, 8].

A major difference of our construction from the previous ones is that the white king is set as reedy for checkmate by one move and there is no way to be saved. In Figure 1 the white king (K) at the center is surrounded by the black pawns (Ps) directed to move downwards so that not only the king cannot move at all but also any one move of white (e.g. invading from outside area) cannot prohibit checkmate by the two black pawns. This figure is placed an isolated position of the generalized chessboard.

To win White must make a rush of checks until getting checkmate. In our construction, Black must move his king for removing these checks; He can neither take the checking piece nor move a piece between the checking piece and the king. Although there is no formal rule or genre formulating such a king chase property of chess, there are already many beautiful and often very amusing, compositions having the property. Here is one of them by Manolis Strakis in Figure 2.

One can see a wonderful development of king chase problems in the Japanese chess, shogi. In shogi king chase is the most popular genre of problem compositions called tsume-shogi, which requires continuous checks ending up as checkmate. A history of tsume-shogi problems has been over 400 years, one of them is the famous classics made by Ito families about three hundred years ago [4]. Adachi et. al. proved that generalized shogi is EXPTIME complete [1]. Yokota et. al. proved that even generalized tsume-shogi is EXPTIME-complete [9].

In this paper we compose a hard chess problem having the king chase property. Previous compositions and techniques therein [3, 8] may not help us so much because they are too symmetric in between Black and White on both strategy and structure, though the king chase property requires asymmetric constructions; The black king takes a tour around the chessboard during which the white king is completely fixed within Figure 1.

In this paper we are interested in the computational complexity of a language of chess positions designed on $n \times n$ board where the white king is as in Figure 1 yet White has the winning strategy. We call this language as Generalized King Chase Chess or GKCC in short. Obviously GKCC belongs to EXPTIME. Indeed, a GKCC position uses at most $O(n)$ squares on the $n \times n$ board filled by one of the 12 choices of materials. Thus the number of possible GKCC positions is $\binom{n^2}{O(n)} 13^{O(n)} = 2^{O(n \log n)}$ ($13 = 12+$ one for blank mark). Within this number the same phase appears again, implying that GKCC $\in$ EXPTIME. In this paper, we compose EXPTIME hard GKCC positions.

Theorem 1. Generalized king chase chess is EXPTIME complete.

We may not care miscellaneous accounts in general chess composition. One of them is reachability of our construction from the standard starting position. One can check easily that our construction is reachable due to the

*Human Informatics, Nagoya University, Nagoya 464-8601, Japan

E-mail address : tsukiji@info.human.nagoya-u.ac.jp

†Toshiba Corporation, Tokyo 105-8001, Japan

E-mail address : age@gix.or.jp

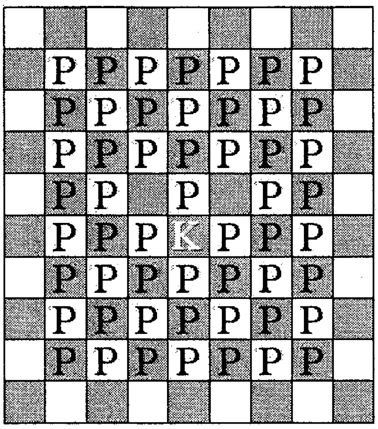

Figure 1: Hung-uped white king.

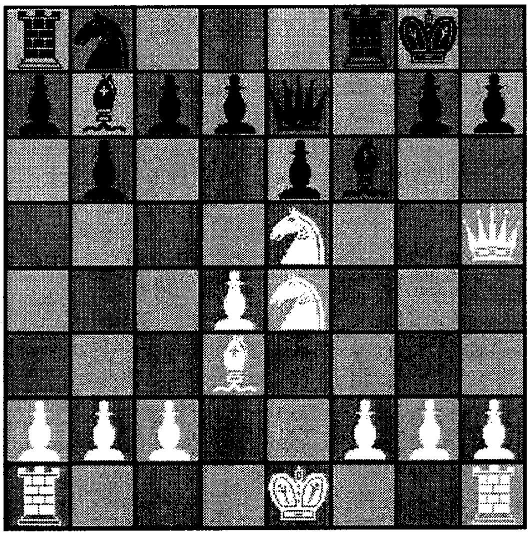

Figure 2: King chase chess problem.

pawn promotion rule. Although knight is not preferred to be used in generalized chess [8][1], here we use a few pieces of white knight for fixing variables; our knight moves as usual, traveling one horizontal and two vertical or one vertical and two horizontal. Finally and most of all, our construction as well as previous ones has little spirit or flavor of 8×8 chess. Our main interest and contribution might be in theories of combinatorial chess games rather than 8×8 chess played in the street.

References

[1] H. Adachi, H. Kamekawa and S. Iwata. Shogi on $n \times n$ board is complete in exponential time, *IEICE Transactions in Japanese*, Vol. J70-D, No. 10, pp. 1843–1852, 1987.

[2] E. D. Demaine. Playing games with algorithms: algorithmic combinatorial game theory, *Mathematical Foundations of Computer Science (MFCS 2001)*, Vol. 2136, pp. 18–32, 2001.

[3] A. S. Fraenkel and D. Lichenstein. Computing a perfect strategy for chess requires time exponential in n. *J. Combinatorial Theory*, Ser. A., Vol. 31, No. 2, pp. 199–214, 1981.

[4] K. Ito. *Shogi Muso*, 1734.

[5] J. M. Robson. The complexity of GO. *Proc. IFIP 1983*, pp. 413–417, 1983.

[6] J. M. Robson. *N* by *N* checkers is EXPTIME complete. *SIAM J. Computing*, Vol. 13, No. 2, pp. 252–267, 1984.

[7] L. J. Stockmeyer and A. K. Chandra. Provably difficult combinatorial games. *SIAM J. Computing*, Vol. 8, No. 2, pp. 151–174, 1979.

[8] J. A. Storer. On the Complexity of Chess, *Journal of Computer and System Sciences*, Vol. 27, No. 1, pp. 77–100, 1983.

[9] M. Yokota, T. Tsukiji, T. Kitagawa, G. Morohashi, and S. Iwata. Exptime-completeness of generalized tsume-shogi. *IEICE Transactions in Japanese*, Vol. D-1 No. 3, pp. 239–246, 2001.

[1] [8] uses only queens, pawns and kings. [3] uses the pieces excepting knight.