# NOTES

## CONNECT6

*I-Chen Wu*[1], *Dei-Yen Huang*[1], *and Hsiu-Chen Chang*[1]

Hsinchu, Taiwan

ABSTRACT

This note introduces the game Connect6, a member of the family of the *k*-in-a-row games, and investigates some related issues. We analyze the fairness of Connect6 and show that Connect6 is potentially fair. Then we describe other characteristics of Connect6, e.g., the high game-tree and state-space complexities. Thereafter we present some threat-based winning strategies for Connect6 players or programs. Finally, the note describes the current developments of Connect6 and lists five new challenges.

## 1. INTRODUCTION

Traditionally, the game *k-in-a-row* is defined as follows. Two players, henceforth represented as *B* and *W*, alternately place one stone, black and white respectively, on one empty *square*[2] of an $m \times n$ board; B is assumed to play first. The player who first obtains *k* consecutive stones (horizontally, vertically or diagonally) of his own colour wins the game. Recently, Wu and Huang (2005) presented a new family of *k*-in-a-row games, *Connect(m,n,k,p,q)*, which are analogous to the traditional *k*-in-a-row games, except that B places *q* stones initially and then both W and B alternately place *p* stones subsequently. The additional parameter *q* is a key that significantly influences the fairness. The games in the family are also referred to as *Connect* games. For simplicity, *Connect(k,p,q)* denotes the games *Connect(∞,∞,k,p,q)*, played on infinite boards.

A well-known and popular Connect game is *five-in-a-row*, also called *Go-Moku*. Go-Moku in the free style (without any restriction on B) is *Connect(15,15,5,1,1)*. However, fairness has been a major issue for Go-Moku. The game is known to favour B over W when played in the free style. After each move made by B, B has one more stone than W, while W only has the same number of stones as B after his[3] move. In order to reduce this unfairness, the Japanese Professional Renju Association (1903) imposed some new rules to restrict the play of B for professionals. For example, B is forbidden from playing double three and double four. Experiences of professionals indicate that the game with these restrictions, known as *Renju*, still favours B. Theoretically, B has been proved to win in the free style by Allis (1994) and Allis, Van den Herik and Huntjens (1996), and under Renju restrictions by Wágner and Virág (2001). The Renju International Federation (RIF) (1998) attempted to increase the fairness of the game by imposing new rules for the first five moves. The RIF (2003) called again for new opening rules and listed the requirements for new rules, indicating that the current rules still need to be improved for this game. However, adding more rules also increases the difficulty of learning the game. Besides, the fairness problem for Go-Moku or Renju has an important side effect of reducing the board size. Sakata and Ikawa (1981) mentioned that increasing the board size raised B's advantage. Hence, the standard board size was set to $15 \times 15$. Indeed, a smaller board lowers the complexity of the game and thus makes it easier to solve the game.

Among the Connect games, *Connect(m,n,6,2,1)* games for all *m* and *n*, referred to as *Connect6*, are interesting research topics for the notions of fairness and game complexity. Connect6 games are *intuitively*

[1] Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan. Email: icwu@csie.nctu.edu.tw, {deryann,hc}@java.csie.nctu.edu.tw.

[2] Practically, stones are placed on empty intersections of Renju or Go boards.

[3] For brevity we will use the pronoun he (his) where he or she (his or her) is meant.

fair, in the sense that one player always has *one* more stone than the other after making each move. After Connect6 was presented by Wu and Huang (2005) during the 11th Advances in Computer Games in Taipei, Taiwan, tens of thousands of players, including many Renju dan players, played this game on a Taiwan game site developed by ThinkNewIdea Inc. (2005). So far, these dan players have not been able to identify which player the game favours. In Section 2 we return to this topic.

As for the board sizes of the Connect6 games, Wu and Huang (2005) set them to infinity, i.e., *Connect*(6,2,1), to maximize the game complexities. However, for playing games at the Computer Olympiad we recommend boards with 19 × 19 squares or 59 × 59 squares in order to make the game feasible to play. For *Connect*(19,19,6,2,1), players can simply use Go boards to play. For *Connect*(59,59,6,2,1), players can play on computers or by putting 3 × 3 Go boards together. Note that 3 × 3 Go boards can be put together to form a board with 59 × 59 squares since placing two Go boards together creates one additional line between the two boards. The *Connect*(19,19,6,2,1) game is initially recommended for Computer Olympiad game contests, while *Connect*(59,59,6,2,1) could be used in games among professionals or used in tie-break games in the future.

The remainder of this paper is structured as follows. Section 2 discusses the fairness issue of the Connect games. Section 3 describes the characteristics of the Connect6 games. Section 4 describes the threat-based winning strategies for Connect6 players and programs. Section 5 presents some of the current developments in Connect6. Finally, Section 6 summarizes this paper and lists five new challenges.

## 2. FAIRNESS

This section discusses the fairness issue of Connect games. Subsection 2.1 provides the definitions of fairness. Subsection 2.2 uses the strategy-stealing arguments to show fairness. Subsection 2.3 raises the fairness issue related to breakaway moves. Subsection 2.4 investigates the fairness of Connect6, while Subsection 2.5 investigates the fairness of other Connect games.

### 2.1 Definitions

In Van den Herik, Uiterwijk, and Van Rijswijck (2002) an adequate definition of fairness is given. It reads: “a game is considered *fair* if it is a draw and both players have a roughly equal probability of making a mistake.” However, practically, it is hard to have a perfect model for calculating the probability of making a mistake, since some undiscovered strategies such as making breakaway moves, as described in Subsection 2.3, may result in different probabilities. On the contrary, it is relatively easy and possible to show when a game is *unfair*. Below we provide three distinct definitions for unfair games.

**Definition 1:** A game is *definitely unfair*, if it has been proved that some player wins the game.
For example, since B wins in Go-Moku (in the free style) as described above, Go-Moku is definitely unfair.

**Definition 2:** A game is *monotonically unfair*, if it has been proved that one player does not win the game, but it has not been proved for the other player.
For example, for *Connect*($k,p,p$) or *Connect*($m,n,k,p,p$), W cannot win, based on the so-called *strategy-stealing argument*, as described in Subsection 2.2. Thus, *Connect*(6,1,1), *Connect*(7,1,1) and *Connect*(6,2,2) are all monotonically unfair, since B has not been proved to win or tie in these games. However, since *Connect*(8,1,1) has been proved to be a draw by Zetters (1980), it is not monotonically unfair.

**Definition 3:** A game is *empirically unfair*, if most players, in particular professionals, have claimed that the game favours some player.
For example, before Go-Moku was solved, Go-Moku was empirically unfair, since most professionals claimed that B would win.

**Definition 4:** A game is considered *potentially fair*, if it has not yet been shown or claimed to be definitely unfair, monotonically unfair, or empirically unfair.

This definition indicates that a potentially fair game for the time being may not remain potentially fair in the future. If a game remains potentially fair any longer, it could have a higher chance to be fair.

### 2.2 Strategy-Stealing Argument

The strategy-stealing argument was first raised by Nash in 1949 (cf. Berlekamp, Conway and Guy, in 2003) and used by other researchers, such as Hales and Jewett (1963), Csirmaz (1980), and Pluhar (1994), in the area of combinatorial game theory. The argument shows that W does not have a winning strategy in *Connect*($m,n,k,p,p$). Assume by contradictory that W has a winning strategy, say $S$. Consider the following. B makes the first move at random. Then, B simply follows $S$ (*stealing* W's strategy) which leads to a win. If the strategy $S$ requires B to place on the squares where B placed earlier, B chooses other squares at random again, instead. Thus, B still remains following the strategy $S$ to win the game, contradictory to the assumption.

Based on the same argument, we can also show the following: the higher the value of $q$ in a Connect game, the higher the chance of B winning. Similarly, assume that B has a winning strategy $S$ in *Connect*($m,n,k,p,q$). Then, B simply follows the strategy $S$ to win in *Connect*($m,n,k,p,q+1$), as above. The result hints that the parameter $q$ in Connect games is a key significantly influencing the fairness. More fair and interesting Connect games may be found by adjusting this parameter.

### 2.3 Breakaway and Unfairness

This subsection investigates the fairness issue related to breakaway moves. For a Connect game, a *breakaway move* means to place stones far away from the major battle field, where most stones have been placed. W's first move (after B's first move) is called an *initial breakaway move*, if it is also a breakaway move.

For *Connect*($k,p,q$) the following holds. If W makes an initial breakaway move without a penalty and B is or will be forced to defend itself against W's move, then the game is played like *Connect*($k,p,p$) with W playing first. Such games are highly likely to be monotonically unfair or even definitely unfair. Therefore, it is important to prove that W has a penalty for the initial breakaway move.

### 2.4 Connect6

Connect6 games are *intuitively* fair, in the sense that one player always has *one* more stone than the other after making each move. This subsection shows that Connect6 is, at least, potentially fair for the time being, based on the following three observations.

1. Connect6 has not yet been shown to be definitely unfair.
2. Connect6 has not yet been shown to be monotonically unfair. Most importantly, Wu and Huang (2005) proved that B wins for *Connect*(6,2,3), hinting that W loses if making an initial breakaway move. Due to the penalty, Connect6 cannot be shown to be monotonically unfair or definitely unfair from the argument in Subsection 2.3.
3. Connect6 has not yet been claimed to be empirically unfair. Connect6 has been played by tens of thousands of players including many Renju dan players, over an online game system developed by ThinkNewIdea Inc. (2005). So far, none of these dan players have been able to claim which player, if any, the game favours.

The above three points conclude that Connect6 is potentially fair for the time being. Further evidence is surely needed to determine the fairness of this game or understand this game better.

### 2.5 Connect Games

In addition to the research for Go-Moku and Renju, many other studies have been undertaken relating to the fairness of $k$-in-a-row. Zetters (1980) proved that W ties when $k \geq 8$. Many solved Connect games with $p = q = 1$ are listed by Van den Herik, Uiterwijk, and Van Rijswijck (2002).

Since W does not win in *Connect*($m,n,k,p,p$) based on the strategy-stealing argument as described in Subsection 2.1, many researchers followed an asymmetric version of rules, called *Maker-Breaker*, for simplicity of combinatorial analysis. In this version, W is not allowed to win. In contrast to Maker-Breaker, the version using the normal rule (without the restriction) is called *Maker-Maker*. Let $\delta$ denote $k - p$. In the Maker-Breaker version, Pluhar (2002) proved that W ties under the following condition $C$: $\delta \geq 80 \log_2 p + 160$ and $p \geq 1000$. The condition $C$ is roughly like $\delta = \Omega(\log_2 p)$. This result implies that W still ties under the same condition in the Maker-Maker version. The result can be easily extended to the following corollary.

**Corollary 1**. *For Connect(k,p,q), let k and p satisfy the condition C (as defined above). For all q, where* $1 \leq q \leq p$, *both B and W tie.* □

Now, investigate unfair Connect games (where either B or W wins). First, B wins when $p < \lfloor q/\delta^2 \rfloor (4\delta + 4)$. For example, if B places $\delta^2$ stones on $\delta \times \delta$ squares as a group, then W requires $4\delta + 4$ stones to defend the group. Thus, the above result is obtained when B lets $\lfloor q/\delta^2 \rfloor$ groups be far away from one another. If $q$ is not a multiple of $\delta^2$, we can obtain tighter results. For example, for 4 × 4 squares, B can add 8 additional stones around the four corners (Wu and Huang, 2005) such that W requires more white stones to defend for each additional black stone. If ($q$ mod $\delta^2$) $\leq 8 \lfloor q/\delta^2 \rfloor$, B can put ($q$ mod $\delta^2$) stones around the corners of the $\lfloor q/\delta^2 \rfloor$ groups, and therefore B wins when $p < \lfloor q/\delta^2 \rfloor (4\delta + 4) + (q \bmod \delta^2)$. Otherwise, B can put $8 \lfloor q/\delta^2 \rfloor$ stones around the corners of the $\lfloor q/\delta^2 \rfloor$ groups, and therefore B wins when $p < \lfloor q/\delta^2 \rfloor (4\delta + 4) + 8 \lfloor q/\delta^2 \rfloor$. Thus, the following corollary is obtained.

**Corollary 2**. *Let* $\delta = k - p$. *For Connect(k,p,q) games, B wins when*
$p < \lfloor q/\delta^2 \rfloor (4\delta + 4) + \min(q \bmod \delta^2, 8 \lfloor q/\delta^2 \rfloor)$. □

Moreover, empirical experiments by Wu and Huang (2005) suggest the following conjecture: Connect games with $\delta \leq 3$ are likely to be empirically unfair. From the above, an open problem is whether there are some more fair and interesting Connect games, where $\delta > 3$, $\delta = O(\log_2 p)$, and $p \geq \lfloor q/\delta^2 \rfloor (4\delta + 4) + \min(q \bmod \delta^2, 8 \lfloor q/\delta^2 \rfloor)$.

## 3. CHARACTERISTICS

The four main characteristics of Connect6 are listed below.

1. Connect6 is potentially fair, based on the argument in Subsection 2.4.
2. The rules of Connect6 are quite simple to learn. In contrast, Renju includes some prohibited moves and International Renju even includes an additional number of opening rules.
3. Connect6 is a symmetric game, if the first move by B is disregarded.
4. Both game-tree and state-space complexities for Connect6 are quite high, as described below.

We first consider the game-tree complexity of Connect6. This complexity is much higher than that of Go-Moku and Renju, since placing on two squares per move increases the branching factor by about a factor of half of the board size. For *Connect(19,19,6,2,1)*, assume that the average game length is still 30, the same as that for Go-Moku (Allis, 1994). Since the number of squares on which one stone can be placed is about 300, the number of possibilities for each move is about (300*300/2). The game-tree complexity is thus approximately $(300*300/2)^{30} \approx 10^{140}$, which is significantly higher than that of Go-Moku. The complexity is even higher when using a larger board such as *Connect*(59,59,6,2,1).

Then we consider the state-space complexity. The state-space complexity of *Connect*(19,19,6,2,1) is $10^{172}$, almost the same as that in Go. This complexity is much higher when a larger board, such as *Connect*(59,59,6,2,1), is used.

## 4. THREAT-BASED WINNING STRATEGIES

Analogously as in Go-Moku or Renju, threats are the key to winning Connect6 games. This section describes threat-based winning strategies for Connect6 players and programs. Subsection 4.1 defines threats for Connect6. Subsection 4.2 describes some winning strategies with threat-based search.

### 4.1 Threats

**Definition 5**. *For Connect6, assume that one player, say W, cannot connect six. B is said to have t threats, if and only if W needs to place t stones to prevent B from winning in B's next move.* □

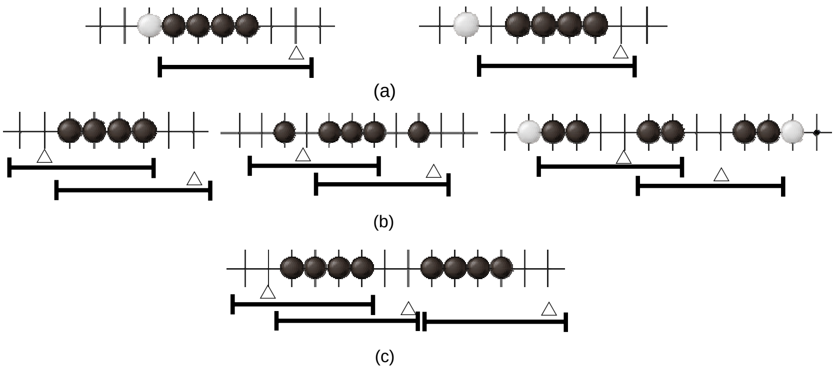



Figure 1: Threat patterns for Connect6. (a) One threat, (b) two threats and (c) three threats.

Threats in Connect6 are defined in Definition 5. In Figures 1 (a), (b), and (c), B has one, two, and three threats, respectively. A move is called a *single-threat move*, if the position has one threat after the move; and a *double-threat move*, if the position has two threats. In the case of three threats, B wins because W requires three stones to defend but only has two stones for a single move. Therefore, a winning strategy is to have at least three threats.

**Lemma 1**. *In Connect6, consider a single line only. Placing one stone on the line, if it does not connect six yet, increases the number of threats by at most two.*

**Proof**. Let B place a stone on square *s* of the line. For each side of *s*, let W place one stone on the empty square closest to *s*. Then, by placing these stones, at most two (one for each side), W must be able to defend all the new threats created by *s*. This implies that placing one stone on the line increases the threat number by at most two. □

**Definition 6**. *In Connect6, consider a single line only. The line includes a dead-l threat for one player, say B, if B only needs to add (4–l) additional stones to generate one threat. Similarly, the line includes a live-l threat for B, if B only needs to add (4–l) additional stones to generate two threats.* □

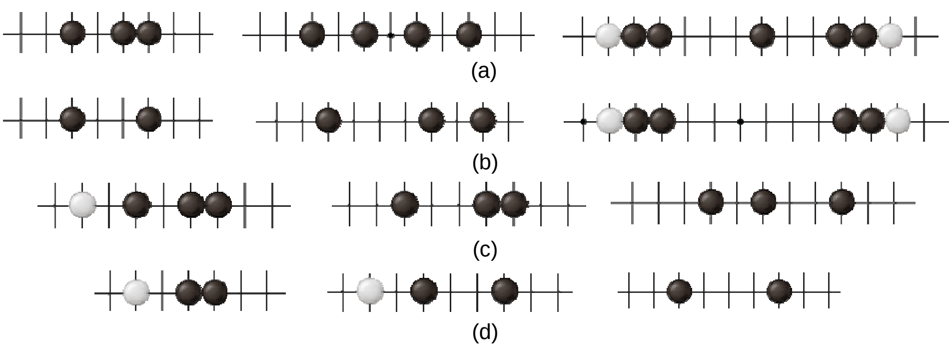
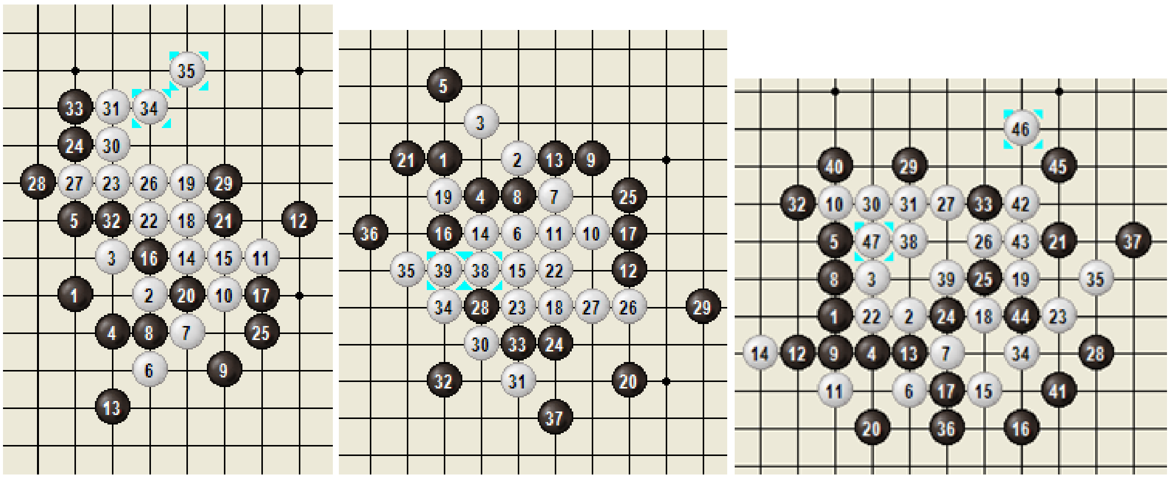



Figure 2: Live-*l* and dead-*l* threats for Connect6. (a) live-3, (b) live-2, (c) dead-3, and (d) dead-2 threats.

Lemma 1 shows that placing one stone on a line increases the number of threats by at most two. Definition 6 defines the *dead-l* and *live-l* threats according to the number of stones that a player must place subsequently in order to generate one or two more threats. For example, Figure 2 illustrates the cases of live-3, live-2, dead-3, and dead-2 threats.

Live-3, live-2, dead-3, and dead-2 threats are also important in Connect6, since one more move (two stones) can transform them into real threats. Among the four threats, live-3, live-2, dead-3 threats are also called *highly potential threats* or *HP-threats*, since one stone can create at least one threat or two stones can create at least two threats. Among HP-threats, live-3 and dead-3 threats are also called *HP3-threats*, which needs only one more stone to have at least one threat. Players usually want to associate real threats with HP-threats, while attacking. This is a rather useful strategy to adopt when playing Connect6.

## 4.2 Threat-Based Search Strategies



(a) (b) (c)

Figure 3: Winning sequences in three positions, respectively starting from the moves (a) at (10,11), (b) at (14,15), (c) at (18,19).

Since Connect6 was presented, many players have learned how to win by making double-threat moves. Double-threat moves are quite powerful, since the opponent is forced to reply to both threats. For example, in Figure 3 (a), after the move at (22, 23), W keeps making double-threat moves to win the game.

Recently, professionals such as Lee (2005) found some Tsumegos that need to mix single-threat and double-threat moves together. For example, in Figure 3 (a), the winning sequence of W starting from (10, 11) needs two single-threat moves at (14,15) and (18,19); in Figure 3 (b), the winning sequence from (14,15) needs one single-threat move at (22,23); in Figure 3 (c), the winning sequence from (18,19) needs one single-threat move at (22,23). The above three Tsumegos have also been solved by Wu and Chang (2006).

However, the search trees mixed with single-threat and double-threat moves are usually tremendously large. In order to reduce the sizes of search trees, Connect6 players and programs should search single-threat moves only under some conditions. Two heuristics for the conditions are as follows.

1. The opponent, called *Defender*, has dead-3 threats but no real threats and no live-3 threats, while the player, called *Attacker*, has one HP3-threat plus at least two additional HP-threats (without counting the single real threat). An example is the move at (22, 23) by Attacker, W, in Figure 3 (b). Defender, B, has only one dead-3 threat and no live-3 threats, while Attacker has one real threat, two HP-3 threats and two HP-threats.

   After placing one stone to defend Attacker's single threat, Defender can place the other either to make a single-threat move or to defend one of Attacker's HP-threats. For the former, Attacker can place one stone to reply to Defender's single threat, while keeping placing the other to make threats in the next move. For the latter, since Defender has no threats, Attacker can use two HP-threats to make a double-threat move.

2. Defender has no real threats and no HP3-threats, while Attacker has at least two additional HP-threats. For example, the move (22,23) in Figure 3 (c).

   Since Defender has no HP3-threats, Defender places one stone to defend Attacker's single threat, while placing the other to defend one of these HP-threats. Thus, Attacker can still make threats in the next move.

For Connect6 programs, it is critical to design good heuristics for the conditions of making single-threat moves while making the tree size small.

## 5. CURRENT DEVELOPMENTS

Connect6 has attracted some attention since its first presentation by Wu and Huang (2005). One game company, ThinkNewIdea Inc. (2005) has already supported an online game system for Connect6, which has been played by tens of thousands of players. In addition, some Renju dan players in Renju clubs in Taiwan also play, study, and teach winning strategies of Connect6. Lee (2005), a Renju dan player, initiated a Connect6 forum, where several Trumegos and Josekis have been published. No players have yet been able to identify which player, if any, the game favours.

A Connect6 program written by Wu and Huang (2005) used an alpha-beta search tree of depth 3 with threat-space search based on double-threat moves only. This program was also connected to the online game system supported by ThinkNewIdea Inc. (2005) to play against human players automatically. The program is strong enough to win against about 70 per cent of the players. A new program written by Wu and Chang (2006) incorporated single-threat moves into threat-space search as described in Subsection 4.2. This program was used to solve several positions including the three in Figure 3.

## 6. SUMMARY AND NEW CHALLENGES

This note investigated several issues related to the game Connect6. For a quick overview we list them below.

1. It discussed the fairness issue of Connect games and showed that Connect6 is still potentially fair.
2. It analyzed other characteristics of Connect6, e.g., high game-tree and state-space complexities.
3. It presented some threat-based winning strategies for Connect6 players or programs.
4. It described the current developments of Connect6.

Since Connect6 is quite new, several new challenges are still open. Below we list five of such challenges.

1. Present more interesting Trumegos or Josekis for Connect6.
2. Develop new search techniques for designing Connect6 programs and for solving more positions.
3. Propose fairness models to assist assessing the fairness of Connect 6 or general Connect games.
4. Solve more Connect games, especially in the cases of $\delta ( = k - p) > 3$ and $\delta = O(\log_2 p)$.
5. Identify some more interesting ones among potentially fair Connect games.

## ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for their valuable comments, which greatly improved this note. The authors would also like to thank Professor Jaap van den Herik for his help and support.

## 7. REFERENCES

Allis, L. V. (1994). *Searching for solutions in games and artificial intelligence*, Ph.D. Thesis, University of Limburg, Maastricht.

Allis, L. V., Herik, H. J. van den, and Huntjens, M. P. H. (1996). Go-Moku Solved by New Search Techniques. *Computational Intelligence*, Vol. 12, pp. 7–23.

Berlekamp, E. R., Conway, J. H., and Guy, R. K. (2003) *Winning Ways for your Mathematical Plays*, Vol. 3, $2^{nd}$ ed., A K Peters. Ltd. Canada.

Csirmaz, L., (1980). On a combinatorial game with an application to Go-Moku, *Discrete Math*., Vol. 29, pp. 19-23.

Hales, A.W., and Jewett, R.I. (1963). Regularity and positional games. Transactions of the American Mathematical Society 106, pp 222–229.

Herik, H. J. van den, Uiterwijk, J.W.H.M., and Rijswijck, J.V. (2002). Games solved: Now and in the future. *Artificial Intelligence*, Vol. 134, pp. 277-311. ISSN 0004-3702.

Japanese Professional Renju Association, (1903). History of Renju Rules, http://www.renjusha.net/database/oldrule.htm.

Lee, S.W. (2005). Joseki and Tsumegos for Connect6. In Connect6 Forum, (in Chinese). http://groups.msn.com/connect6/.

Pluhar, A. (1994). Generalizations of the game k-in-a-row, *Rutcor Res. Rep*. 15-94.

Pluhar, A. (2002). The accelerated k-in-a-row game, *Theoretical Computer Science*. Vol. 271, Nos. 1-2, pp. 865-875. ISSN 0304-3975.

Renju International Federation (1998). The International Rules of Renju, http://www.renju.nu/rifrules.htm.

Renju International Federation (2003). MOM for the RIF General Assembly, http://www.renju.nu/wc2003/MOM_RIF_030805.htm.

Sakata, G. and Ikawa, W. (1981). *Five-In-A-Row. Renju*. The Ishi Press, Inc., Tokyo, Japan.

ThinkNewIdea Limited (2005). CYC game web site (in Chinese). http://www.cycgame.com.

Uiterwijk, J.W.H.M., and Herik, H.J. van den (2000). The advantage of the initiative, *Information Sciences*, Vol. 122, No. 1, pp. 43–58.

Wágner, J. and Virág, I. (2001) Solving Renju, *ICGA Journal*, Vol. 24, No. 1, pp. 30–34.

Wu, I-C. and Chang, H.-C. (2006). Threat-based proof search for Connect6. Technical report, Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan. http://java.csie.nctu.edu.tw/~icwu/technical-reports/tr1-2006.pdf.

Wu, I-C. and Huang, D.-Y. (2006) A New Family of *k*-in-a-row Games. The 11th Advances in Computer Games (ACG11) Conference, Taipei, Taiwan. (to be published)

Zetters, T.G.L. (1980). Problem S.10 proposed by R.K. Guy and J.L. Selfridge. *Amer. Math. Monthly* 86 (1979), solution 87(1980), pp. 575-576.