New Simulation Strategy of MCTS for Connect6

Shi-Jim Yen and Jung-Kuei Yang

This paper proposed a new simulation strategy of Monte Carlo Tree Search for Connect6. The core idea of the new simulation strategy is to put the search on proving or disproving a sudden-death property at first stage, and it searches the most promising move at second stage when the first stage cannot find its solution. The experimental results show that the new simulation strategy of MCTS can perform much efficiency than traditional MCTS on those positions that has TSS solution in Connect6.

I. Introduction

A. Monte-Carlo Tree Search

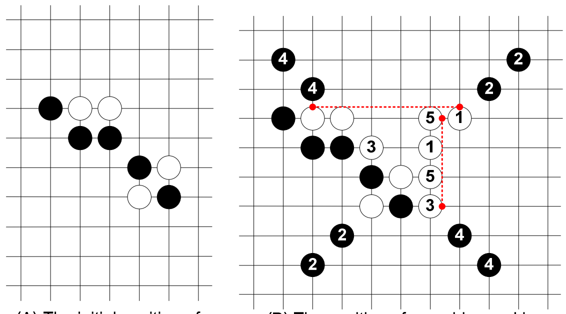
Searching is a way we use to solve problems as well as a skill of programs to exhibit their intelligence. Recently, MCTS has become a very popular game search method, and it has been applied to many game searches. The core idea of MCTS is to sample from enormous branches by playout from leaf node, and it corrects the mistakes in the upper sampling position [1] by developing the correspondent branches of search tree.

Traditional MCTS does not fit with the property of sudden-death game. Sudden-death is a property of quickly deciding the winner of a game. Therefore, in every position, one has to consider whether there is sudden-death. If one neglects this feature, the other side will win.

B. The Introduction of Connect6

Connect6 has two important features: enormous candidate moves and sudden-death. Enormous candidate moves make its branching factors quite high; sudden-death makes searching complexity increase. The property of sudden-death in Connect6 is Threat Space Search, TSS [1][2]. TSS is the common search method in Connect-*k* games. About the definition of threat in Connect6 and its kinds, please refer to [4][5]. *Connection* is a most commonly used information when searching in Connect-*k* games. For information on saving and calculating Connection, please see [6][7].

When in one position, if any side can find the process to win through TSS, this side can win through it. Because the rules of Connect6 allow the player to place two stones simultaneously except for the first move, TSS can be divided into double-threat TSS and single-threat TSS in Connect6. This paper uses *T2 solution* to represent the process to win through double-threat TSS and *TSS solution* to represent the process to win through single-threat TSS.



(A) The initial position of searching

(B) The position of searching end in CTSS, the White Wins

Figure 1. A sudden-death position with T2 solution and its CTSS solution

Figure 1 is an example of T2 solution. For this position, the White can search by Conservative double-threat TSS to find solution in level 5 of the search tree. Numbers in the figure represent the cells[2] in each searching level. For this position, if one neglects sudden-death in the initial search position, the search will have a significant impact

Shi-Jim Yen is with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan (e-mail: sjyen@mail.ndhu.edu.tw).

Jung-Kuei Yang is a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, and he is also with the Department of Applied Foreign Languages, Lan Yang Institute of Technology, I Lan, Taiwan. (e-mail: jungkuei@gmail.com).

[1] Position represents all the state sets of cells on the board after one side play stones. The state of cell is divided into three types: Empty, Black, and White.

[2] The intersections to place stones on the board is called cell in this paper.

on the correctness.

C. AND/OR Tree

This study uses an AND/OR tree to develop the search tree to fit the situation of Connect6. For any side, if there are T2 or TSS solutions under the initial search position, it must find the solution out, and the AND/OR tree is an appropriate means of doing this. Figure 2 is an example of AND/OR search tree.

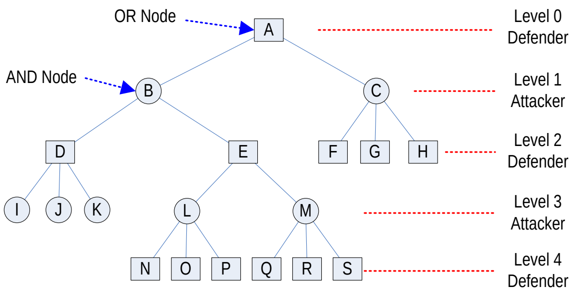


Figure 2. An AND/OR search tree. Rectangle represents an OR-Node (Defensive side), and circle represents an AND-Node (Offensive side).

II. New Simulation Strategy in MCTS

This section introduces new simulation strategy of MCTS for Connect6. This study only describes the strategies which are different from traditional MCTS.

A. The proposed new search architecture of MCTS

The new simulation strategy this study proposed is shown in Figure 3. The main difference is the strategy of expanding leaf node. For control the development of search tree, traditional MCTS develops one leaf node in every simulation. The new expansion strategy this study proposed expands two level or more nodes from leaf node in every simulation.

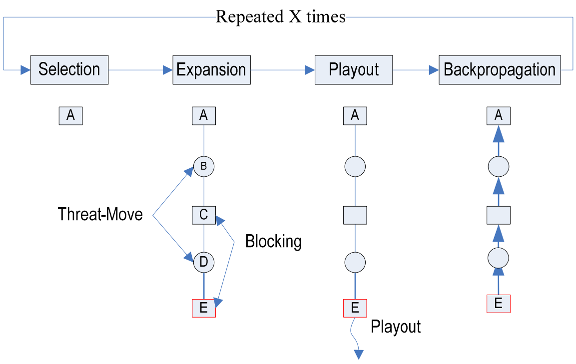


Figure 3. The strategy of expanding nodes from leaf node for MCTS in Connect6

In Figure 3, the nodes expanded by expansion strategy are four nodes. Nodes B, C, D, and E are expanded from node A in this figure. Therefore, the place which MCTS runs playout is from node E.

The selection strategy of new MCTS is same as traditional MCTS. The enhancement of simulation strategy is expansion, playout, and back-propagation. The differences are described as follows.

B. Expansion strategy

When MCTS uses playout to do predict, it supposes that the importance of each move are the same. This is not suitable on the sudden-death game because the importance of candidate moves obviously is not equal. Therefore, this study uses 2-Stage to develop candidate moves.

Based on the property of sudden-death, this study supposes that all the sudden-death states under one position should have to be explored if resource is enough. It means the *Threat-Move*[3] under one position should have to be explored firstly in Connect6, so 2-Stage is adapted.

Besides, if all the sudden-death states have to be explored, the search algorithm does not need to repeat evaluations for these states. Therefore, this study thinks Threat-Move should be expanded before playout runs until there is not Threat-Move. This expansion strategy can ensure that the candidate moves that playout uses from leaf node have the same importance (non-threat moves).

But, the efficiency is worse when this study uses this strategy to searches the T2 or TSS solution. The best strategy is expanding two level nodes based on the experiments.

C. Playout Strategy

For sudden-death game, the key to win is to search for sudden-death position. When one side reaches this state, this side can win. Because Connect6 has enormous candidate moves in every position, this study thinks how to find the nearest sudden-death position from the initial search position is more important than to explore the endgame position.

Therefore, this study limits the depth of playout according to the property of sudden-death game.

[3] Threat-Move represents double-threat or single-threat moves.

D. Back-propagation Strategy

Because this study not only uses playout to predict the leaf node, but also uses CTSS to search for double-threat moves, it includes the reporting-back of both situations.

For the situation of reporting-back when CTSS finds the T2 solution, this study uses different *Wins* value based on the level of the evaluation node. This design is in order to highlight the importance of sudden-death in different level from the initial search position.

From Figure 4, it is much important to find the sudden-death from level 2 to level 4. Therefore, this study decrease the *Wins* value based on the level of search tree. The *Wins* value of node A is greater than node B in Figure 4.

This strategy is only used at when Attacker finds the CTSS solution. This study does not use this strategy on when Defender finds the CTSS solution.

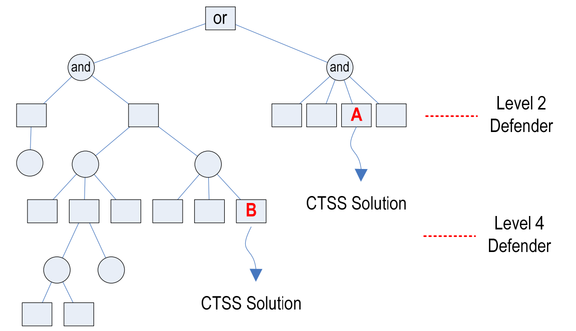


Figure 4. The strategy of expanding nodes from leaf node for MCTS in Connect6

E. Conclusion for new simulation strategy

There are four key points for the new simulation strategy which we proposed.

1. It uses 2-Stage to develop candidate moves.
2. It expands two level nodes in every simulation.
3. It limits the depth of playout.
4. It uses different *Wins* value when CTSS finds T2 solution in different level.

III. Experiment

The experiment aims to analyze positions with T2 solution and TSS solution, the accuracy and efficiency of new simulation strategy of MCTS. This study developed a new version of Kavalan 4.0, named Kavalan 4.1. Kavalan 4.0 was developed based on MCTS technique, and has participated in the 2010 Computer Olympiad, Connect6. Kavalan 4.0 wins the second prize in the tournament of 15th Computer Olympiad, Connect6.

The puzzles this study used in experiment are obtained from Connect6 web site [3]. This study gathers the latest year puzzles, and obtains 30 puzzles for use as an experimental test benchmark for the algorithm. Among the 30 puzzles, this study excludes two, 2008-Q1-1-3 and 2008-Q3-1-4, because they lack T2 or TSS solution. For the 28 puzzles, 13 questions belong to T2 solution, and 15 belong to TSS solution. The classification of puzzles is showed in Table 1.

Table 1. The Classification of Puzzles

T2 solution	2008-Q1-1-1	2008-Q1-1-2	2008-Q1-2-1	2008-Q1-2-2
	2008-Q1-2-4	2008-Q1-3-1	2008-Q1-3-2	2008-Q2-1-1
	2008-Q2-1-2	2008-Q2-2-1	2008-Q2-2-2	2008-Q3-1-1
	2008-Q3-1-2			
TSS solution	2008-Q1-1-4	2008-Q1-1-5	2008-Q1-2-3	2008-Q1-2-5
	2008-Q1-3-3	2008-Q1-3-4	2008-Q1-3-5	2008-Q2-1-3
	2008-Q2-1-4	2008-Q2-1-5	2008-Q2-2-3	2008-Q2-2-4
	2008-Q2-2-5	2008-Q3-1-3	2008-Q3-1-5	

The experiments were performed on 2.0 GHz with 2 GB of memory running Windows XP. The control variable of MCTS is assigned as Table 2.

Table 2. The Control Variable and its Value

Control Variable		Value
Maximum Probing Positions		60,000
Probing Cells		12
Exploration Coefficient		1.2
Heuristic Value to double-threat moves		300
The depth of playout		30
Back-Propagation	CTSS – Attacker Win	12
	CTSS – Attacker Fail	-5
	Playout – Attacker Win	1
	Playout – Attacker Fail	-1

A. T2 solution

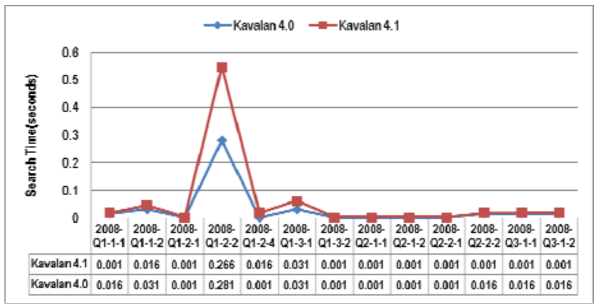
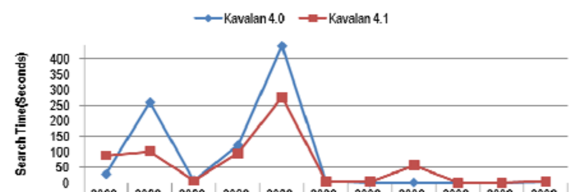


Figure 5. The comparison chart of T2 solution

From Figure 5, it shows that when MCTS searches T2 solution, Kavalan 4.0 is better than Kavalan 4.1. This is mainly because the puzzles with T2 solution, its solution must search from the level 1 of search tree. In this case, the expansion strategy of leaf node will be a waste of computing time.

B. TSS solution

From Figure 6, it shows that when Kavalan 4.1 searches TSS solution, it is better than Kavalan 4.0. TSS solution is more complex question than T2 solution, so searching for TSS solution the states it must probing is quite enormous when the solution is not in level 1.



	2008-Q1-1-4	2008-Q1-2-3	2008-Q1-2-5	2008-Q1-3-3	2008-Q1-3-4	2008-Q2-1-3	2008-Q2-1-4	2008-Q2-2-3	2008-Q2-2-4	2008-Q3-1-3	2008-Q3-1-5
Kavalan 4.0	28.188	258.844	7.906	121.563	441.296	6.140	1.000	1.219	0.625	1.656	3.640
Kavalan 4.1	88.406	101.328	6.469	95.359	276.063	4.500	3.406	55.578	0.203	0.141	4.734

Figure 6. The comparison chart of TSS solution

In Figure 6, this study excludes 4 puzzles because it cannot find the TSS solution under the limit of maximum probing positions. So far the Kavalan can solve 73%(11/15) puzzles with TSS solution.

C. Performance Analysis

This study tested the playing strength of the proposed new simulation strategy of MCTS (Kavalan 4.1) against traditional MCTS (Kavalan 4.0). The result of contest is that Kavalan 4.1 won 5 of 6 games.

Based on the new simulation strategy described in this study, the Kavalan 4.1 is significantly stronger than Kavalan 4.0, and can beat it 83% of the time.

IV. Conclusion

This paper investigates new MCTS technique for Connect6 game. The contribution of this paper is mainly to present a new MCTS simulation method to enhancement the MCTS for sudden-death game, Connect6.

The experiments show that this strategy outperforms traditional MCTS on those positions that have TSS solution. Furthermore, the contest of Kavalan 4.0 and Kavalan 4.1 show that the new simulation strategy of MCTS outperforms traditional MCTS on the search performance. Therefore, it is useful for sudden-death game.

V. References

[1] L.V. Allis, “Searching for solutions in games and artificial intelligence,” Ph.D. Thesis, University of Limburg, Maastricht, The Netherlands, 1994.

[2] L.V. Allis, H.J. van den Herik and M.P.H. Huntjens, “Go-Moku Solved by New Search Techniques,” Computational Intelligence, vol. 12, pp. 7-23, 1996.

[3] Taiwan Connect6 Association, Connect6 homepage, Available: http://www.connect6.org/.

[4] I-C. Wu, D.-Y. Huang and H.-C. Chang, “Connect6,” ICGA Journal, Vol. 28, No. 4, pp. 234-241, 2005.

[5] I-C. Wu and S.-J. Yen, “NCTU6 Wins Connect6 Tournament,” *ICGA Journal*, vol. 29(3), pp. 157–158, September 2006.

[6] C.-M. Xu, Z.-M. Ma, X.-H. Xu. “A method to construct knowledge table-base in k-in-a-row games,” Proceedings of the 2009 ACM symposium on Applied Computing, pp. 929-933, 2009.

[7] S.-J. Yen, J.-K. Yang, "The Bitboard Design and Bitwise Computing in Connect6," the 14th Game Programming Workshop (GPW-09), November 13-15, 2009, Hakone Seminar House, Kanagawa, Japan: Proceeding, pp. 95-98, 2009.