

Contents

Abstract	ii
Acknowledgments	iv
1 Introduction	1
1.1 Background	1
1.2 Research Questions	5
1.3 Structure of the Thesis	5
2 Literature Review	9
2.1 Using the Information above the Leaf Nodes	9
2.1.1 Introduction	9
2.1.2 Conspiracy Number Search	10
2.1.3 Proof Number Search	11
2.1.4 Df-pn	14
2.1.5 The Seesaw Effect	16
2.1.6 Deep Proof Number Search	17
2.2 Using the Information beneath the Leaf Nodes	29
2.2.1 Introduction	29
2.2.2 Monte-Carlo Tree Search	29
2.2.3 Monte-Carlo Tree Search Solver	32
2.3 Combining the Information above and beneath the Leaf Nodes	34
2.3.1 Introduction	34
2.3.2 Monte-Carlo Proof Number Search	34
2.4 Chapter Conclusion	36

3	Deep df-pn	37
3.1	Introduction	37
3.2	Basic Idea of Deep df-pn	39
3.3	Deep df-pn in Connect6	42
3.3.1	Connect6	42
3.3.2	Relevance-zone-oriented Deep df-pn	44
3.3.3	Experimental Design	48
3.3.4	Results and Discussion	49
3.3.5	Comparison	50
3.3.6	Finding optimized parameters	51
3.4	Chapter Conclusion	54
4	Single Conspiracy Number	56
4.1	Introduction	56
4.2	Basic Idea of Single Conspiracy Number	58
4.3	Experiments and Discussion	60
4.3.1	Experimental Design	61
4.3.2	Tactical Positions	61
4.3.3	Drawn Positions	66
4.3.4	Opening Positions	68
4.4	Chapter Conclusion	70
5	Probability-based Proof Number Search	71
5.1	Introduction	71
5.2	Probability-based Proof Number Search	73
5.2.1	Main Concept	73
5.2.2	Probability-based Proof Number	74
5.2.3	Algorithm	75
5.3	Benchmarks	75
5.3.1	Monte-Carlo Proof Number Search	75
5.3.2	Monte-Carlo Tree Search Solver	78
5.4	Experiments	82

5.5 Chapter Conclusion	85
6 Conclusion	88
Appendix	90
Bibliography	95
Publications	101

List of Figures

2-1	An example of MIN/MAX tree [40]	11
2-2	An example of conspiracy numbers (the left column of the number list of a node is the evaluation value, and the right column of the number list of a node is its corresponded conspiracy number) [40]	12
2-3	An example of the expansion and updating process in the conspiracy number search (the left column of the number list of a node is the evaluation value, and the right column of the number list of a node is its corresponded conspiracy number) [40]	13
2-4	An example of the seesaw effect: (a) An example game tree (b) Expanding the most-proving node [19]	17
2-5	An example of a suitable tree for an Othello end-game position. This game tree has a uniform depth of 4, and the terminal nodes are reached at game end. [19]	18
2-6	The variation in Othello: The number of # of Iterations and # of Nodes. $R = 1.0$ is PN-search, $R = 0.0$ is depth-first search, and $1.0 > R > 0.0$ is DeepPN. Lower is better. [19]	23
2-7	The reduction rate in Othello: The number of # of Iterations and # of Nodes. $R = 1.0$ is PN-search, $R = 0.0$ is depth-first search, and $1.0 > R > 0.0$ is DeepPN. Lower is better. [19]	24
2-8	# of Iterations in Othello: The changes of Reducing and Increasing Cases for # of Iterations and # of Nodes [19]	25
2-9	# of Nodes in Othello: The changes of Reducing and Increasing Cases for # of Iterations and # of Nodes [19]	26

2-10	The variation in Hex: The changes of Reducing and Increasing Cases for # of Iterations and # of Nodes [19]	27
2-11	The reduction rate in Hex: # of Iterations and # of Nodes for Hex(4). $R = 1.0$ is PN-search, $R = 0.0$ is depth-first search, and $1.0 > R > 0.0$ is DeepPN. Lower is better. [19]	27
2-12	Hex: The detail of Fig 2-11. This figure is zoomed $1.0 \leq R \leq 0.9$. The lower is better. [19]	28
2-13	One iteration of the general MCTS approach [7]	31
3-1	Relationship between PN-search, df-pn, DeepPN and Deep df-pn	39
3-2	An example of the seesaw effect: (a) An example game tree (b) Expanding the most-proving node [19]	41
3-3	An example of (a) relevance zone Z and (b) relevance zone Z' [55]	43
3-4	Example position 1 of Connect6 (Black is to move and Black wins)	45
3-5	Example position 2 of Connect6 (Black is to move and Black wins)	45
3-6	Deep df-pn and df-pn compared in node number (including repeatedly traversed nodes) with various values of parameter E and D for position 1 (Df-pn when $D = 1$)	46
3-7	Deep df-pn and df-pn compared in seesaw effect number with various values of parameter E and D for position 1 (Df-pn when $D = 1$)	46
3-8	Deep df-pn and df-pn compared in node number (including repeatedly traversed nodes) with various values of parameter E and D for position 2 (Df-pn when $D = 1$)	47
3-9	Deep df-pn and df-pn compared in seesaw effect number with various values of parameter E and D for position 2 (Df-pn when $D = 1$)	47
3-10	Node number (including repeatedly traversed nodes) of $1 + \epsilon$ trick with various values of parameter ϵ for position 1	51
3-11	Seesaw effect number of $1 + \epsilon$ trick with various values of parameter ϵ for position 1	52
3-12	Node number (including repeatedly traversed nodes) of $1 + \epsilon$ trick with various values of parameter ϵ for position 2	52

3-13	Seesaw effect number of $1 + \epsilon$ trick with various values of parameter ϵ for position 2	53
4-1	An example of tactical position where Red is to move (Red wins)	61
4-2	Red's MIN/MAX value and SCN in position P_0 with different search depth ($T = 600$)	62
4-3	Black's MIN/MAX value and SCN in position P_1 with different search depth ($T = 600$)	62
4-4	Histogram of $RSCN_{red}$ and $RSCN_{black}$ ($T = 600$)	65
4-5	Histogram of Vl_{red} and Vl_{black}	65
4-6	Histogram of $RSCN$ in winning positions, losing positions and drawn positions ($T = 600$)	67
4-7	Histogram of Vl in winning positions, losing positions and drawn positions	67
4-8	Relationship between high possibility, low possibility and normal possibility of changing the MIN/MAX values of positions not less than the threshold T in advance of the opponent scaled by $RSCN$	67
4-9	$RSCNs$ for different handicap openings with $T = 200$	68
4-10	$RSCNs$ for different handicap openings with $T = 600$	68
5-1	Two examples of updating PN by MIN rule in MCPN-search (the square represents the OR node).	77
5-2	Two examples of updating PPN by OR rule in PPN-search (the square represents the OR node). Notice that $PPN = 1 - PN$	77
5-3	Two examples of updating PN by SUM rule in MCPN-search (the circle represents the AND node).	78
5-4	Two examples of updating PPN by AND rule in PPN-search (the circle represents the AND node). Notice that $PPN = 1 - PN$	78
5-5	Two examples of updating simulation values by taking the average in the UCT solver or the pure MCTS solver (the square represents the OR node).	81
5-6	Two examples of updating PPN by OR rule in PPN-search (the square represents the OR node).	82
5-7	Comparison of average search time for a P-game tree with 2 branches and 20 layers.	84

5-8	Comparison of average numbers of iterations for a P-game tree with 2 branches and 20 layers.	84
5-9	Comparison of the error rate of selected moves for each iteration on P-game trees with 2 branches and 20 layers.	85
5-10	Comparison of average search time for a P-game tree with 8 branches and 8 layers.	85
5-11	Comparison of average numbers of iterations for a P-game tree with 8 branches and 8 layers.	86
5-12	Comparison of the error rate of selected moves for each iteration on P-game trees with 8 branches and 8 layers.	86
6-1	Example position 3 of Connect6 (Black is to move and Black wins)	90
6-2	Example position 4 of Connect6 (Black is to move and Black wins)	90
6-3	Example position 5 of Connect6 (Black is to move and Black wins)	91
6-4	Example position 6 of Connect6 (Black is to move and Black wins)	91
6-5	Example position 7 of Connect6 (White is to move and White wins)	91
6-6	Example position 8 of Connect6 (White is to move and White wins)	91

List of Tables

3.1	Different behaviors by changing parameters	42
3.2	Deep df-pn and $1 + \epsilon$ trick compared in the best case (The number in the bracket represents the reduction percentage compared with df-pn)	53
3.3	Experimental data of Deep df-pn using hill-climbing method (The number in the bracket represents the difference between Deep df-pn using hill-climbing method and Deep df-pn in the best case)	54