Solving Tria and Tessera: *n*-in-a-Row Variations with a Twist

Lillian Alice Wu
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*—This paper serves as an introduction to the games of Tria and Tessera with varying rule sets. We attempt to solve Tria & Tessera with the *n*-in-a-Row variations game solver. We implement the solver with different techniques including AlphaBeta pruning, transposition tables, and move ordering. For Tria, an analysis on the effectiveness of different approaches is completed and the minimum board size of the winning game is found. Furthermore, the characteristics and winning strategies of each of Tria variation is explored and discussed. Lastly, the challenges surrounding the implementation of Tessera are considered. These challenges are analysed as the reasons behind them are further discussed.**

***Index Terms*—Tria, Tessera, Pente, tree search, AlphaBeta pruning, transposition table, Zobrist hashing, move ordering, Three-in-A-Row, Tic-Tac-Toe**

I. Introduction

The games "Tria" and "Tessera" are both variations of the known game Pente. In Pente, players take turns placing stones on a 19×19 grid as they try to form a line of five in a row. Pente borrows this objective from the game Gomoku. In Gomoku, the objective is also forming a line of five stones, however, in Pente, reaching this objective is more complex since players are allowed to capture their opponent's stones.

The goals of Tria and Tessera are similar. As in Gomoku, player's are required to get x number of stones in a row (three for Tria and four for Tessera). They are similar to traditional Three-in-a-Row and Four-in-a-Row games, but with an extra rule extracted from Pente, that is, allowing players to capture stones. This thesis explores the positional game Tria with multiple board sizes and various rule sets. The goal is to find the smallest board size for a player to win the game. Different constraints on the board size and winning conditions are examined. Most of the previous research on Pente uses a Neural Network [1]. For both Tria & Tessera, we use the tree search approach AlphaBeta. Different methods and enhancements are implemented to optimize AlphaBeta, to reduce the number of nodes investigated, and to improve runtime.

A. Game Rules

1) The game board has a $m \times n$ grid. The values of m and n depend on the game.
2) There are two players: one playing as Black and one playing as White. The player playing Black goes first.
3) Players take turns to place stones on the board.
4) Capturing
   (a) Capture-1 rule: A stone is captured when it is directly surrounded on two sides by the opponent's stone.
   (b) Capture-2 rule: A pair of stones is captured when it is directly surrounded on two sides by the opponent's stone.

   Fig. 1 shows how stones are captured using the Capture-1 and Capture-2 rules. The line of stones can be horizontal, vertical or diagonal. There is no limit to how many stones a player can capture.
5) After the capturing condition is met, it is required for the player to remove the opponent's captured stone(s) so the spot(s) become(s) available again. Both players are free to place their stones in that space in the following rounds.
6) In an n-in-a-Row game, a player with n stones connected in an uninterrupted horizontal, vertical, or diagonal line wins. For example, in a 3-in-a-Row game, a player with three stones connected in a horizontal, vertical, or diagonal line wins and in a 4-in-a-Row game, a player with 4 stones connected wins.

In the game of Pente [2], a player capturing 5 pairs of stones (with only the Capture-2 rule applied) immediately wins the game. However, we exclude this rule to stick to the objective of forming a line.

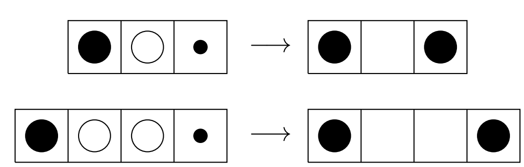

Fig. 1. The process of Capture-1 (top) and Capture-2 (bottom).

B. Game Moves Strategy

The following are four strategies behind the moves that players can make.

- Win: If the player has two stones in a row, they can place a third to get three in a row.

This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Data Science and Knowledge Engineering, Maastricht University. Supervisor: Jos Uiterwijk

- Threat: create more than two threats so the opponent cannot counter them all. It could be a capturing threat and a winning threat at the same time.
- Block: If the opponent has two stones in a row, or if the opponent is about to capture your stone, place a stone to block their move and counter their threat.
- Blocking a multi-threat: block the possibility for the opponent to form a multi-threat. If the multi-threat is already formed, always try to block the opponent from winning, even if it means letting them capture a stone. If both threats are winning threats, then the opponent can only block both threats simultaneously if possible, otherwise the player to move will win.

C. Game variations and Definition

There are four variations of Tria.

- Tria-1: Capturing-1 rule enabled.
- Tria-2: Capturing-2 rule enabled.
- Tria-12: Both Capture-1 and Capture-2 rules enabled.
- Tria-0: The Three-in-a-Row game with no capturing allowed. It is the Tic-Tac-Toe game on larger boards.

There are also four variations of Tessera. Similar rules apply but the objective is to form a line of four instead of three:

- Tessera-0: The Four-in-a-Row game with no capturing rule.
- Tessera-1: Capture-1 rule enabled.
- Tessera-2: Capture-2 rule enabled.
- Tesssea-12: Both Capture-1 and Capture-2 rules enabled.

D. Problem Statement and Research Questions

In this thesis, we will attempt to examine the games Tria & Tessera and find out their characteristics. An *n*-in-a-Row variations solver will be implemented and designed to be as efficient as possible for different rule sets variations. The goal is to answer the following research questions with the completed program by conducting several experiments.

1) What is the smallest board size for a player to win the game?
2) Does adding the new rule sets give the first player a higher chance of winning?
3) How do the new capturing rule sets influence the length of the game?

II. Related Work

The games Tria and Tessera are inspired by Pente. Pente is played on a 19×19 board. For a board that is as large as 19×19 and has the Capture-2 rule added, the game tree becomes very large. Therefore, the branching factor is too large to investigate the search tree in a short amount of time [1]. That is why previous studies on Pente have been done with Neural Networks [3]. Unlike Pente, the game tree of Tria is much smaller because the objective of forming a line of three is less challenging than in Pente. Thus, the tree search approach is viable for this game. Tria is similar to Three-in-a-Row in terms of the game objective since they both have the objective of forming a line of three connected stones. Therefore, we follow a similar approach to Three-in-a-Row. Three-in-a-Row is a complete solved game. It is known that 3×4 is the smallest board size for a winning game to take place. Regarding Tessera, 5×6 is the smallest board size of a winning Four-in-a-Row game to take place [4].

III. Approach

A. AlphaBeta

AlphaBeta is used to solve these games [5] [6]. AlphaBeta is essentially the MiniMax algorithm plus a nodes-pruning enhancement for efficiency. Therefore, when the maximizing player finds a winning state, AlphaBeta prunes the rest of the nodes and returns the winning value to the previous depth. This way, there is no need to go through every node of a tree. The AlphaBeta value indicating the appropriate time to make the cut-off depends on whether the player is maximizing or minimizing. In short, AlphaBeta iteratively eliminates nodes and sub-trees that are less than optimal.

B. Transposition Tables

Different order of steps could result in the same position. Yet, they are proceeded with the same computation. In order to reduce the duplicate computation from the same position, we store the information of the explored position to a transposition table [7]. The transposition-table technique consists of two steps: hashing and storing.

Hashing is the process of converting the board position to a hash value. We use hashing to represent each distinct position of the board. Here the Zobrist hashing [8] approach is used. First, we initialize Zobrist hashing with two sets of $m \times n$ numbers of randomized values of 64 bits. Each value of the first $m \times n$ set represents when a black stone is on that spot. Likewise, the values of the second set denote instances of white stones on the board. The Zobrist hash value of a board is obtained by XORing each respected value of the stones on the board.

Storing is important to maximize the use of the transposition table. The transposition table has a limited size of 2^{27} entries. To reduce memory usage, each hash value is stored to a designated entry. The index is simply the first 27 bits of the Zobrist hashing. By doing so, there is no need to store the full 64 bits of the hash value to the table, just the last 37 bits (64 – 27). The first 27 bits can be looked up from its index. By only using 37 bits of the memory, we can utilize the other 27 bits for information storing. The information stored are the results and the depth level.

1) Result: can be 1, –1 or 0, indicating a win, a lose, or a draw respectively.
2) Depth level: the depth level is used when a collision occurs. A collision occurs when two hashing values share the same first 27 bits, but do not share the rest of the 37 bits. Hence they are not the same board positions. When a collision happens the position with shallower depth is preferred and is stored in the transposition table.

Each board position has multiple symmetry positions that share the same computation. Thus during each iteration we can also produce its symmetrical hashing by flipping the board vertically, horizontally and both. By doing so we get four hash values in total. We then use the smallest Zobrist hashing index. In this way all four symmetrical positions will be mapped onto the same entry in the transposition table. Implementing symmetries can be a great improvement for finding the winning node in the transposition table.

The transposition table can only store up to 2^{27} positions. Since there are more positions than 2^{27}, it is important to determine which positions stay in the transposition when the two positions' hash values have the same first 27 bits. We choose to keep the position with the shallower depth. That is because when it is higher in the tree, the position is more generic and is more likely to reoccur. More importantly, a position higher in the tree will probably have caused more investigated nodes than a position lower in the tree, and consequently can give larger variation in tree size.

C. Move Ordering

To maximize the efficiency of AlphaBeta, it is very useful to order the moves so that better moves are tried first. By doing so, it is more likely to prune the nodes earlier and reduce the number of nodes needed to be investigated.

The moves are ordered by the number of lines the Black player could still possibly form from that move. This is denoted as the degree of a move [9]. Furthermore, for variations where capturing is allowed, the number of captures that can be made from the move also contributes to the value of the degree. Thus, the degree is calculated by:

$$degree = lines + (Capture1) \times 3 + (Capture2) \times 1$$

In the formula above, *lines* is the number of lines Black can form from the move. *Capture1* and *Capture2* are the numbers of captures that can be made from the Capture-1 rule and Capture-2 rule respectively. We have a higher multiplier value of *Capture1* than the one of *Capture2* to prevent being counter-captured by the opponent in Capture-1. More on counter-capturing is discussed later. (See Section IV.E)

The White player is sharing the same set of degrees because its objective is to prevent the Black player from forming a line. Thus, by playing the best move from the Black player, the White player can weaken the Black player. Fig. 2 shows an example of the degree of moves on two 4×4 boards.

3	4	4	3
4	7	7	4
4	7	7	4
3	4	4	3

3	3	3	2
3	●	○	6
4	5	5	3
2	4	3	3

Fig. 2. Degree calculation of Tria-1 on the empty board and after the moves on **b3** and **c3**

D. Win or Block

The capturing rules significantly increase the number of nodes. More information about this can be found in the results sections. In order to narrow down the possible moves, in each iteration, we check if the position is only one stone away from termination. This means that if either of the players can win the game by playing one move, they will do so immediately. Under two conditions this may happen:

1) The current player can win the game by making a move. Thus, it should instantly choose this move without having to check other moves and the game ends.
2) The current player's opponent can win the next round. Thus, the player needs to find a way to resolve this threat, either by playing the winning move (of its opponent) first or destroying the soon-to-be line with capturing.

In Tria-1, the second condition is especially useful after each capturing is made by the opponent. Capturing immediately creates a threat for the current player, so it reduces the search tree. For Tria-2, it is not necessary to fill the space from previous capturing.

E. Loop Prevention

The capturing rule can cause a loop. A loop is formed when the two players continuously capture the same set of stones while the rest of the board is unchanged. Loops cause the program to investigate an infinite number of steps. For example, in Tria-1 a loop is easily formed when there is a line with X O _ O →X _ X O →X O _ O and so on. In Tria-2, loops also occur in more complex ways and with multiple captures involved.

We handle loops by keeping track of a list of boards. The list consists of the progression of the board. After a move, we first check if the position exists in the list already. If so, it means that a loop has formed. In this case, we assign zero (a draw) to this position and we remove all the boards followed by this position from the list. Otherwise, if it does not exist in the list, we add it to the list and the algorithm proceeds.

IV. Tria – Experiments and Discussion

The experiments are divided into two parts. In Tria, first, we explore the different methods and how effective they are. Second, we analyze different types of Tria individually.

1) Comparison of the Enhancements: For this experiment, we look at the number of nodes with different enhancements enabled, including AlphaBeta, Transposition Table, and Move Ordering. From the number of nodes explored, we observe how large the game tree is and how the approach effectively reduces the number of nodes.

2) Tria variants game analysis and comparison: One of the research goals is to determine the influence on game length with different capturing rule sets. Thus, for this experiment, we compare the number of nodes with various versions of Tria. Additionally, we select some of the Tria variants and do strategic analysis on the gameplays for more understanding of their characteristics.

TABLE I
NUMBER OF NODES OF VARIATIONS OF TRIA ON DIFFERENT BOARD SIZES

	Tria-0		Tria-1		Tria-2	
Board size	AB	+WB	AB	+WB	AB	+WB
3×3	*16811	*2951	1843	246	*16,811	*2,951
3×4	4,4624	589	12,950	232	66,476	749
4×4	245,560	418	140,396	280	2,399,163	450
4×5	15,595,108	1062	-	560	-	2,752

*Note: values proceeded by * indicates that the variation result in a draw; the Black player wins on all variations without **

A. Enhancements

1) Win or Block Enhancement: From Table I we see that having the Win or Block strategy significantly reduces the number of nodes needed. In this table, AB stands for AlphaBeta and WB for the Win or Block enhancement. When running Tria-1 and Tria-2 without the "Win or Block" strategy, the game trees exceed 3,000 depths and the program crashes on just the 4×5 board. Without this strategy we are unable to further run experiments on larger boards, therefore, for the experiments onward, all experiments adapt the Win or Block strategy.

2) Transposition Table and Symmetry: For the experiments on Transposition Table, we investigate the number of nodes explored while Transposition Table is in use. Furthermore, we compare the number of nodes when symmetry is added to see how it impacts the number of nodes needed to run the solver.

TABLE II
NUMBER OF NODES WITH TRANSPOSITION TABLE + SYMMETRY

	Tria-1			Tria-2		
Board size	TT(-S)	TT(S)	% reduced	TT(-S)	TT(S)	% reduced
3×3	197	187	5.08%	*1,332	*731	45.12%
3×4	204	196	3.92%	570	496	12.98%
4×5	504	468	7.14%	1,684	1,655	1.72%
5×6	694	690	0.58%	12,667	12,603	0.51%
2×4	*899	*438	51.28%	*1546	*719	53.49%

*Note: values proceeded by * indicates that the variation results in a draw; the Black player wins on all variations without **

Table II demonstrates the number of nodes reduced when the three board symmetries are added to avoid duplicate computation. In the table, TT(S) stands for Transpositon Table with symmetry and TT(-S) for the Transposition Table enhancement without symmetry. Transposition Table with symmetry has around 6% less nodes than without symmetry for both Tria-1 and Tria-2. Having symmetry does reduce a small portion of nodes, but it does not remove as many as expected. An explanation for this is that a solution (winning node) is found before many symmetries occur. Therefore, since the 3×3 Tria-2 results in a draw, all possible nodes are explored and 45% of nodes are reduced, which is higher than all other games that are not draws. We test Tria-1 and Tria-2 on the 2×4 board, where both variations result in a draw. The outcome is that there is a much higher rate of nodes reduced for the 2×4 board as well, at around 50% for both Tria-1 and Tria-2 variations. Similar results is found for Tria-0 and Tria-12. (See Appendix.B.)

B. Enhancement Comparison between Tria versions

1) Tria-1 Enhancement Comparison: We compare the different enhancements added to AlphaBeta from board size 3×3 to 10×10 of Tria-1.

TABLE III
TRIA-1 – NUMBER OF NODES WITH DIFFERENT ENHANCEMENTS

Board size	AB	AB+TT	AB+MO	AB+TT+MO	Best %
3×3	246	187	367	220	76.02%
3×4	232	196	62	41	17.67%
4×4	280	252	66	63	22.50%
4×5	560	468	88	51	9.11%
5×5	816	643	114	68	8.33%
5×6	753	690	134	75	9.96%
6×6	825	762	144	120	14.55%
6×7	1,081	936	202	194	17.95%
7×7	1,179	1,034	220	215	18.24%
7×8	4,689	3,219	462	462	9.85%
8×8	7,239	5,262	960	913	12.61%
8×9	1,907	1,784	2,446	2,156	93.55%
9×9	2,051	1,928	10,649	8,335	94.00%
9×10	2,588	2,278	18,780	13,824	88.02%
10×10	2,768	2,458	63,463	49,253	88.80%

Note: numbers with gray shade represent the best result from the row; the Black player wins on all variations above

Table III shows the number of nodes used with different enhancement combinations on Tria-1. MO stands for move ordering as explained in section III.C. AlphaBeta + Transposition Table continuously reduces nodes on all boards as expected. On average, AlphaBeta + Transposition Table reduces around 15% of nodes from having only AlphaBeta. Having AlphaBeta + Move Ordering seems much more effective at first, reducing on average around 70% of nodes up to the board size 8×8. Once the board size reaches 8×9, AlphaBeta + Move Ordering becomes worse than having only AlphaBeta.

One possible reason that Move Ordering is worse on boards larger than 8×9 is that there is a change in preferred moves on larger boards; therefore the ranking system does not reflect how good a move is anymore so the number of nodes explored increased.

2) Tria-2 Enhancement Comparison: Table IV shows the number of nodes used with different enhancement combinations on Tria-2. AlphaBeta + Transposition Table reduces 52.3% of nodes on average while AlphaBeta + Move Ordering reduces on average 81.6%. AlphaBeta + Transposition Table reduces around 15% of nodes from only having the AlphaBeta enhancement. Combining the two gives us the best result of cutting down 94.6% of nodes. For Tria-2, the performance of Move Ordering is consistent and greatly improved the results.

3) Tria-12 Enhancement Comparison: Table V shows the number of nodes used with different enhancement combinations on Tria-12. It has a similar outcome as Tria-2. AlphaBeta + Transposition Table reduces 55.2% of nodes on average while AlphaBeta + Move Ordering reduces on average 76.8%. Combining the two gives us the best result of cutting down 87.7% of nodes. In Tria-12, the performance of Move Ordering is also consistent and improved the results.

TABLE IV
TRIA-2 – NUMBER OF NODES WITH DIFFERENT ENHANCEMENT

Board size	AB	AB+TT	AB+MO	AB+TT+MO	Best %
3×3	*2,951	*731	*978	*244	8.27%
3×4	749	496	60	39	5.21%
4×4	450	349	62	59	13.11%
4×5	2,752	1,655	402	124	4.51%
5×5	3,736	2,275	3,438	588	15.74%
5×6	27,044	12,603	1,024	413	1.53%
6×6	42,304	19,741	142	118	0.28%
6×7	330,316	65,888	206	195	0.06%
7×7	730,936	161,550	278	264	0.04%
7×8	-	1,177,694	742	703	-
8×8	-	3,125,007	6,126	3,434	-
8×9	-	12,440,845	25,528	16,366	-
9×9	-	-	211,754	127,388	-
9×10	-	-	1,837,672	901,453	-
10×10	-	-	23,834,524	9,037,408	-

*Note: numbers with gray shade represent the best result from the row; values proceeded by * indicates that the variation result in a draw; the Black player wins on all variations without **

TABLE V
TRIA-12 – NUMBER OF NODES WITH DIFFERENT ENHANCEMENT

Board size	AB	AB+TT	AB+MO	AB+TT+MO	Best %
3×3	246	187	367	220	76.02%
3×4	248	212	62	41	16.53%
4×4	280	252	66	63	22.50%
4×5	779	681	384	123	15.79%
5×5	1,043	856	368	156	14.96%
5×6	2,150	1,583	1,056	230	10.70%
6×6	2,354	1,787	144	120	5.10%
6×7	7,603	4,051	204	196	2.58%
7×7	8,506	625	220	215	2.53%
7×8	74,664	11,039	508	508	0.68%
8×8	82,080	12,703	1,038	994	1.21%
8×9	738,731	31,161	3,803	3,052	0.41%
9×9	876,278	35,778	10,790	9,102	1.04%
9×10	10,310,646	68,477	5,533	5,109	0.05%
10×10	11,139,866	76,849	17,286	15,077	0.14%

Note: numbers with gray shade represent the best result from the row; the Black player wins on all variations above

TABLE VI
BEST NUMBER OF NODES OF TRIA VARIATIONS AND BOARD SIZES

Board size	Tria-0	Tria-1	Tria-2	Tria-12
1×5	*41	17	*57	25
3×3	*244	187	*244	187
3×4	39	39	39	41
4×4	59	59	59	63
4×5	49	51	124	123
5×5	66	66	588	156
5×6	69	71	413	230
6×6	118	118	118	120
6×7	189	190	195	196
7×7	258	245	264	215
7×8	672	801	703	508
8×8	3,251	3,110	3,434	994
8×9	15,124	1,784	16,366	3,052
9×9	52,103	1,928	127,388	9,102
9×10	116,559	2,278	901,453	5,109
10×10	133,933	2,458	9,037,408	15,077

*: values proceeded by * indicates that the variation result in a draw.

C. Tria variations comparison

Table VI shows the number of nodes used on different board sizes and game rules variations. Each variation is using the best combination of enhancements found. The first player wins on all variations except the variations that end with draws (denoted by *). The smallest board size of a winning game on Tria-1 and Tria-12 is just 1×5, while it's 3×4 for both Tria-2 and Tria-0.

From board sizes 3×4 up to 7×8, Tria-0 tends to need the least amount of nodes to find a winning node, though all variants of games require a fairly similar number of nodes. Once the board size reaches 8×9, Tria-1 drastically uses less nodes than other Tria variations. We can hypothesize that Tria-1 gives the first player more advantages on larger boards than when the game is played on smaller boards. Tria-2 tends to use the most number of nodes while Tria-12 falls somewhere between Tria-1 and Tria-2 since its capturing rule is a combination of the two games. In Tria-2, after a capturing move is tried, it increases the number of nodes to be investigated in its parent node. Thus, Tria-2 uses more nodes than the other variations. The winning strategies of all Tria variations will be further discussed in later sections.

D. Game Analysis

To understand more about the strategy of each Tria variation, we make an analysis on some selected games.

1) 1×5 Game Analysis (See Fig. 3): The first player can win on many sizes of boards on Tria-1. The smallest board size is 1×5 with only 17 nodes investigated to get the result. The Black player would win by playing in the middle first. The White player is left with two choices of moves, playing adjacent to Black or the corner. The corner move will lead to a win for Black with a double-end empty threat (Open-2), thus playing adjacent to Black is preferred (1). Though playing next to Black allows the Black player to capture White in the next round at **b**, it does block the Black player from forming a line at that location. Then, Black will play by the corner (2), which yields White to capture its stone at **c**. After White makes the move at **d** in (3), though Black's stone is being captured from this move, it allows Black to capture both White's stones remaining on the board (4) and also wins the game (5). This is an example that capturing stones could be an disadvantage to a player. White making the move on **d** in (3) could have ended the game with a draw if it wasn't mandatory to capture stones. In this game, the Black player wins the game by forcing White to capture Black's stone.

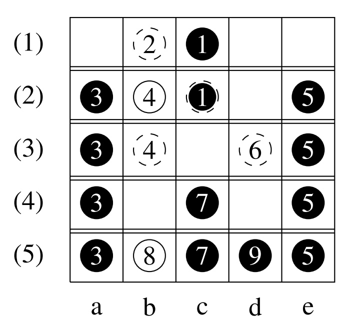

Fig. 3. Tria-1 on the 1×5 board.
Note: Dashed stones are stones that get captured in the next round.

2) Tic-Tac-Toe vs. 3×3 *Tria-1 :*

For Tic-Tac-Toe, the best move on the empty 3×3 board is most likely the one in the center. That is due to the fact that the moves in the center tend to have a higher degree and can contribute to forming lines in all directions. However, that is certainly not the case for Tria-1. On a 3×3 board, making the move right in the center gives the next player the possibility to capture that stone from any direction, thus it is the worst opening move on Tria-1. On the other hand, making moves in the corners is good because the four corners are the spots that are not possible to be captured. Therefore in 3×3 Tria-1, the first player that gets the corner can win the game. (See Fig. 4)

Once a player's stone gets captured, that player needs to make moves on the spot where its stone gets captured previously. Otherwise its opponent can form a line of three in the next round. Since how the opponent will react from being captured is determined, the capturing component gives the captor the advantage of making moves two times in a row rather than eliminating its opponent's stones in the long run like in Tria-2 and variations with the Capture-2 rule.

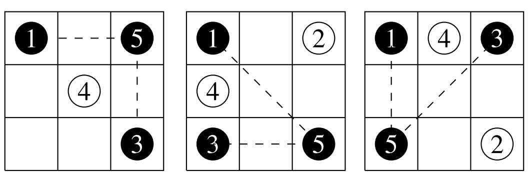

Fig. 4. Three main variations on Tria-1 on the 3×3 board.

E. Winning Strategy on Larger Boards for Tria-1

On larger boards, for example the 5×5 board, playing in the center becomes an advantage again. That is because the center move not only has a higher degree, but it also has the advantage of capturing any stones adjacent to it without the confinement of the border. Therefore, on larger (empty) boards, the White player is restricted from getting close to the center when Black is there. That is why the Black player could win more easily by trying to form lines from all eight directions. On the other hand, players in some cases are forced to make the capturing move when it is available. Otherwise, they can be counter-captured by the opponent which could result in losing the game. In Fig. 5, the Black player is required to capture White at **b4**, otherwise they could be counter-captured by White from **d2** and lose the game.

F. Winning Strategy for Tria-2

The Capture-2 rule from Tria-2 allows player to remove two stones of the opponent with one stone. It is therefore a bigger trade-off than capturing in Tria-1. In Tria-1, when one stone is captured, it can simply be recovered in the next round. Tria-2 has the smallest winning game on 3×4. Yet due to the smaller board size of 3×4, it is less often for Capture-2 to occur. Thus the first player could win on Tria-2 without making or encountering capturing threats.

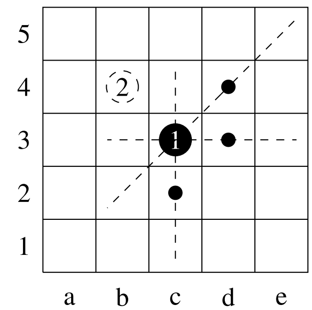

Fig. 5. Tria-1 on the 5×5 board. White playing anywhere adjacent to Black will get captured. Furthermore, the board size is big enough for Black to create a double-end threat in multiple directions.

G. Winning Strategy for Tria-12

Tria-12 follows a similar winning strategy to Tria-1. Due to the strong advantage for the first player in Tria-1, a winning game is often found before any Capture-2 moves are made.

V. Tessera – Experiments and Discussion

A. Challenges

It is much more complex to form a line of four than three. The four-in-a-row game tree has significantly more nodes that need to be explored so it takes a lot longer to reach an ending state. We test our solver on Tessera using the same capturing rules and modified enhancements with the objective to form a line of four. However, we are unable to get much of results from Tessera. The reason is that after a capture is made, it does not decrease the number of nodes to be explored but instead increases it. Here we talk about why Tria is not affected by this.

1) Unable to reduce nodes after captures are made: We are able to reduce nodes after each capture in Tria-1. We have discussed the impact of having the Win or Block strategy in Tria. (See Section IV.A.2.) For Capture-1 rule, it is easy to meet the capturing condition; hence Capture-1 creates more child nodes than Capture-2. The Win or Block strategy forces the current player to fill the captured spots from the opponent (if winning is not possible for the current player) and it effectively reduces the number of nodes used. However, this is not the case for Tessera. In Tessera-1, it is not necessary to fill the captured spot when it is not possible to form a line of four at that location. Even when it is possible to form a line of four there, players can still choose to fill the captured spots later when there is actually a threat from its opponent.

2) More complex objective: There is a big difference in difficulty to achieve their objectives. Like in Tessera, it is not necessary to fill the captured spots with the Capture-2 rule in Tria-2. Yet, due to Tria's objective of forming a line of three, it is much simpler to achieve and thus the solver often finds a solution after trying only the first few nodes. For Four-in-a-Row, it is more complex and requires more built-up to form the line.

B. Experiments

1) Results: In Tria-1, much of the “Blocking” from “Win or Block” comes from after Capture-1 happens. We are therefore able to successfully reduce many nodes in Tria-1. Without being able to narrow down nodes after captures are done, Tessera reacts in a way similar to when Tria-2 is run without the Win or Block strategy. The program goes into depth of thousands and crashes. The Tessera solver with the same enhancements added from Tria is able to solve Tessera game on 4×4 for all Tessera variations, as well as on 4×5 for Tessera-2. (Table VII)

TABLE VII
NUMBER OF NODES OF TESSERA VARIATIONS

	Tessera-0	Tessera-1	Tessera-2	Tessera-12	Outcome
4×4	860,571	7,613,032	1,138,980	12,645,895	draw
4×5	21,753,897	-	51,276,832	-	draw

2) Tessera-1 nodes reduced after captures: We would like to see how the solver would perform with Tessera if we can reduce nodes after captures are made. Therefore we added a rule to Tessera-1 that after each capture is made, the next player is required to fill the captured spot if direct winning is not possible. With this rule added, the first player is able to win on the 4×5 board with 36,233,587 nodes investigated. The winning strategy is essentially capturing stones and creating the threat of forming 4-in-a-row at the same time. The player facing the situation will be forced to fill the spots where its stones were captured and will therefore be unable to block the opponent from forming a line. The process of the game can be found in Fig. 6.

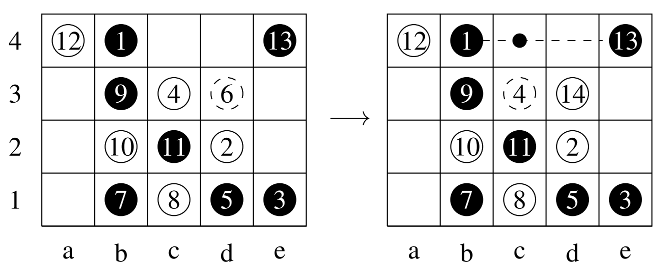

Fig. 6. Tessera variation on the 4×5 board

Fig. 6 shows how the Black player can win. The Black player making the move at **c4** captures White at **c3**, at the same time creating the threat of forming a line of four from b4 to e4.

VI. CONCLUSION

In this research, we seek to understand the game Tria by implementing a solver with various enhancements. It is proven that the enhancement can effectively narrow down the number of nodes needed for different board sizes and game settings. Specifically, we would like to know:

i) What is the smallest board size for a player to win the game?;
ii) Does adding the new rule sets give the first player a higher chance to win?;
iii) How do the new capturing rule sets influence the length of the game?

Our key findings are the following. First, the smallest board size of a winning game for Tria-1 & Tria-12 is at 1×5 and for Tria-2 is at 3×4. Second, from the results of Tria-1, we see that Capture-1 rule gives the first player more advantages since the opponent’s stone is refrained from being adjacent to the first player’s. The same advantage applies to the Tria-12 variation. Third, about how the rule sets affect the length of the game, with our solver Tria-2 requires the most nodes while Tria-1 needs the least.

VII. FUTURE RESEARCH

For Tria, an improvement to Move Ordering can be done in the future. It is demonstrated that Move Ordering significantly reduces the number of nodes needed to find the winning game. Yet the performance of Move Ordering worsens on larger boards for Tria-1. Each variation of Tria can have a more customized Move Ordering. Furthermore, it is possible to adapt the rule from Pente for both Tria and Tessera, where the number of captures allowed is limited. Once a player reaches the maximum number of captures, they also win the game. Doing so could reduce the number of nodes and make it easier to reach the end of the game. Tessera can potentially be solved by the solver if the number of captures is limited.

REFERENCES

[1] Muijrers, V. C. (2011). Training a Back-Propagation Network with Temporal Difference Learning and a database for the board game Pente.
[2] *How to play the game of Pente*. (n.d.). Pente Net. https://www.pente.net/instructions.html
[3] Schaul, T., Schmidhuber, J. (2009, September). Scalable neural networks for board games. In International Conference on Artificial Neural Networks (pp. 1005-1014). Springer, Berlin, Heidelberg.
[4] Uiterwijk, Jos WHM. ”Solving Strong and Weak 4-in-a-Row.” In *2019 IEEE Conference on Games (CoG)*, pp. 1-8. IEEE, 2019.
[5] Knuth, D. E., Moore, R. W. (1975). An analysis of alpha-beta pruning. Artificial intelligence, 6(4), 293-326.
[6] Marsland, T. A. (1986). A review of game-tree pruning. *ICGA journal, 9(1)*, 3-19.
[7] Breuker, D. M., Uiterwijk, J. W., van den Herik, H. J. (1994). Replacement schemes for transposition tables. *ICGA Journal, 17*(4), 183-193.
[8] Zobrist, Albert L. ”A new hashing method with application for game playing.” *ICGA Journal* 13, no. 2 (1990): 69-73.
[9] J.W.H.M. Uiterwijk, Solving Four-in-a-Row (Unpublished work)
[10] Dickey, T. E. (1980). U.S. Patent No. 4,213,616. Washington, DC: U.S. Patent and Trademark Office.
[11] Schrum, J. (2006). Neuro-Evolution in Multi-Player Pente. Department of Computer Sciences University of Texas at Austin.

APPENDIX

A. Tria-0 Enhancement Comparison

Table VIII shows the number of nodes used with different enhancement combinations on Tria-0. It has a similar outcome to Tria-1 where AlphaBeta + Move Ordering works well on smaller boards. Yet when the board size reaches 9×9, AlphaBeta + Transposition Table becomes the best enhancement. On average, AlphaBeta + Transposition Table reduces around 47% of nodes from having AlphaBeta only. With AlphaBeta + Move Ordering seems much more effective at first, reducing

TABLE VIII
ENHANCEMENT COMPARISON FOR TRIA-0

Board size	AB	AB+TT	AB+MO	AB+TT+MO	Best %
3×3	*2,951	*731	*978	*244	8.27%
3×4	598	419	60	39	6.52%
4×4	418	340	62	59	14.11%
4×5	1,062	840	86	49	4.61%
5×5	1,422	1,180	106	66	4.64%
5×6	3,618	2,548	124	69	1.91%
6×6	4,602	3,322	142	118	2.56%
6×7	12,012	6,682	200	189	1.57%
7×7	14,434	8,259	272	258	1.79%
7×8	42,370	17,684	714	672	1.59%
8×8	50,402	21,372	4,476	3,251	6.45%
8×9	136,024	44,524	22,492	15,124	11.12%
9×9	156,634	52,103	186,926	113,853	33.26%
9×10	476,674	116,559	1,596,646	792,719	24.45%
10×10	542,494	133,933	20,659,578	8,019,737	24.69%
				Best % Avg.:	9.84%

*Note: numbers with gray shade represent the best result from the row; values proceeded by * indicates that the variation result in a draw; the Black player wins on all variations without * above*

on average around 90% of nodes up from board size 3×3 to 8×9. Once the board size reaches 8×9, AlphaBeta + Move Ordering becomes worse than having only AlphaBeta.

B. Transposition Table and symmetry Enhancement for Tria-0 and Tria-12

TABLE IX
NUMBER OF NODES WITH TRANSPOSITION TABLE + SYMMETRY

	Tria-0			Tria-12		
Board size	TT(-S)	TT(S)	% reduced	TT(-S)	TT(S)	% reduced
3×3	*1,312	*731	44.28%	197	187	5.08%
3×4	453	419	7.51%	220	212	3.64%
4×5	848	840	0.94%	697	681	2.30%
5×6	2593	2548	1.74%	1,585	1,538	2.97%
2×4	*1035	*506	51.11%	*1398	*677	51.57%

*Note: values proceeded by * indicates that the variation results in a draw; the Black player wins on all variations without **

Table IX demonstrates the number of nodes used for Tria-0 and Tria-12 with symmetry added to the Transposition Table enhancement. It shows that when the variations result in draws, a larger portion of nodes is reduced.