MACHINE AS ONE PLAYER IN INDIAN COWRY BOARD GAME: BASIC PLAYING STRATEGIES

Pouyan Davoudian

Department of Studies in Computer Science, University of Mysore, India

P. Nagabhushan

Indian Institute of Information Technology, Allahabad, India;

Department of Studies in Computer Science, University of Mysore, India

ABSTRACT

Cowry game is an ancient board game from India, also known as Chowka Bhara. It is a race game of chance and strategy for 2-4 players, in which playing pieces are moved around a square board according to the throw of special dice (cowry shells). This game involves decision-making under uncertainty and imprecision with multiple players, and therefore can be considered as an appropriate model for real-life problems that contain stochastic components. In this research, we propose and analyze a few basic playing strategies for Cowry game, and describe the framework for the implementation of these strategies. We also provide an experimental comparison of the proposed strategies to evaluate their performances. The comprehensive study of Cowry game presented in this work can be used to gain a better understanding of the game, and may result in the formulation and implementation of more advanced strategies. It can also serve as a basis for producing better artificial players in similar strategic race games.

Keywords: Chowka Bhara, Strategy, Random, Fast, Balanced, Cowry shell, Race board game, Artificial player.

Cite this Article: Pouyan Davoudian and P. Nagabhushan, Machine as one Player in Indian Cowry Board Game: Basic Playing Strategies, *International Journal of Computer Engineering and Technology*, 10(1), 2019, pp. 174-183.

<http://iaeme.com/Home/issue/IJCET?Volume=10&Issue=1>

1. INTRODUCTION

The idea of programming computers to play board games has been around since the emergence of computation. Complex board games like *Chess*, *Checkers* and *Go* have been historically acknowledged as great test domains for exploring and developing various types of approaches in artificial intelligence and machine learning [1]. As a result, highly optimized

techniques have been evolved for such games, and they have been improved dramatically to an extent where a machine can play at expert level, competing with world-class human players [2, 3]. Comparatively, far fewer successful programs have been designed to play games like *Ludo* and its variant race games, most of which are descended from the ancient Indian games *Pachisi* and *Chaupar*. We argue that there is a compelling reason behind it. Game theory classifies these games into two fundamentally different categories, and the methods applicable to one are not necessarily suitable for the other.

In games such as *Chess*, *Checkers* and *Go* the players have complete knowledge about the current state of the game. They know what options will be available for the next move, as well as the result of every possible move. Even though the players cannot predict their opponent's future move, they have complete knowledge of all possible moves the opponent will be allowed to make [4]. These games are commonly known as “perfect information”.

Race games like *Ludo*, however, contain an element of chance because of dice. The players cannot predict their dice roll and consequently, they do not know what options they may have for the next move. Similarly, the players do not know in advance what moves their opponent can possibly make because they cannot know of the opponent's dice roll [4]. Due to uncertainty about the future moves, finding the best move in the current state of these games might be challenging. Therefore, the players are often required to estimate probabilities and calculate risk. This type of game is called “imperfect information”.

While a great deal of effort has gone into designing computer programs to play perfect information two-player board games, many strategic games are more complex, involving imperfect information with more than two players. One of such games is *Cowry* game, popularly known as *Chowka Bhara*, which is a traditional game of India. It is a multi-player, imperfect information, stochastic, strategic race board game, yet to be played at expert level by a machine. Despite our extensive search in relevant academic literature, we could not identify any scientific work tackling this problem. In this research, we explore, implement, and evaluate a few basic playing strategies for *Cowry* game, which can result in the discovery of more advanced strategies to improve its gameplay.

In the remainder of this paper, we provide a short description of *Cowry* game, as well as its history in Section 2. We also summarize the rules of the game which are considered for the implementation in this work. Section 3 presents a brief overview of our proposed strategies and the logic behind them. In Section 4, we provide details about designing and programming the game environment. We also give an illustrated description of the framework used for the implementation of the proposed strategies. In Section 5, we evaluate the performance of the strategies and provide a comparative analysis of the obtained experimental results. Finally, Section 6 highlights the achievements and concludes the paper by giving guidelines for future research.

2. GAME DESCRIPTION AND RULES

Cowry game is a race board game for two to four players, in which the players race their tokens or “pieces” around the board in a spiral, with the objective of being the first to move all of their pieces to the innermost location of the board. Like many complex race games, *Cowry* game involves a combination of luck and skill [5]. Piece movement is determined by throwing four cowry shells (a special variety of seashells, shown in Figure 1), and hence it can be considered as a game of chance. However, there is also an element of strategy to the game. Players have four pieces to move, and so with each throw of cowry shells they must choose from different options for moving their pieces, and predict possible countermoves by the opponents.

Figure 1 Cowry shells

Cowry game is one of the most ancient board games originated in India. Its history can be traced back to the times of epic “Mahabharata”. *Cowry* game and its variations are still popular and being played in different regions of India under various names, such as *Chowka Bara*, *Chakaara*, *Katte Mane*, *Ashta Chamma*, *Pagdi*, *Kavidi Kali*, *Thaayam*, etc. Despite several regional variations of this game, the description and rules presented in this section are considered for our standard implementation throughout this research work.

The board is normally a square divided into five rows and columns. Each player is assigned a color (typically red, yellow, green and blue), and has four pieces of the same color. The outer middle squares on each side of the board represent the starting squares for each player, where initially four pieces of a player are placed. The starting squares are specially marked on the board and also function as “safe squares”. Only one piece can be placed on any square of the board with the exception of safe squares, which can accommodate any number of pieces. The innermost square of the board is the finishing square for all players. Figure 2 depicts the game board and shows safe squares.

The players start on their respective starting squares and proceed to race their pieces anticlockwise around the board along the outer squares. Upon reaching the square on the left side of their starting square, the players continue by moving the pieces up to the inner squares and proceed clockwise towards the finishing square. The path for one of the players is shown in Figure 2 with dotted lines.

Figure 2 Board of *Cowry* game with safe squares, and path for a player

To start the game, each player throws the cowry shells and the player with the highest value moves first. The players alternate turns around the board in an anticlockwise direction. The first player to bring all four pieces to the finishing square wins. The first winner then leaves the game and others continue playing to identify second and third winners, and the last player is the final loser.

Four cowry shells are used to decide how far to move the players' pieces. They are thrown and the number of cowry shells landing with their openings upwards indicates the number of squares the player should move. However, when all four cowry shells land with their openings downwards, the value of the throw is considered to be 8. Thus, the possible values for a single throw of four cowry shells are 1, 2, 3, 4 and 8. When the players throw either a 4 or an 8, they will receive an additional or "bonus" throw of cowry shells in that turn. If the bonus throw results in a 4 or an 8 again, the players will obtain another bonus throw. However, if the third throw is also a 4 or an 8, they will not be allowed to make any move and the turn will pass to the next player.

In case the players get two or more consecutive throws, they can move one or more of their pieces according to the obtained values. Nevertheless, only one piece can be moved with each throw of cowry shells. For instance, if the players throw an 8, they will get an additional chance to throw the cowry shells. If on the bonus throw they get a 3, then they will have the option to move one piece 8 squares forward, and another piece 3 squares forward. They may also choose to move the same piece 11 squares forward, as long as two separate and legal moves can be made: either 8 and then 3, or 3 and then 8.

After throwing the cowry shells at the beginning of each turn, the players must, if possible, move their pieces forward along the path designated for that player, and according to the value indicated by the cowry shells. Forfeiting the turn voluntarily is not permitted in this game. If no legal move is possible, the turn will automatically pass to the next player.

The players may not land their pieces on any square (apart from a safe square) occupied by another piece of the same color. The players may, however, eliminate or "hit" an opponent's piece on a non-safe square by landing their piece upon it. The eliminated piece is then returned to its starting square, and the player making the hit is awarded a bonus throw. The players are not allowed to move their pieces into the inner squares unless they have hit at least one piece of any of the opponents. Once this "first-hit" condition is fulfilled, all four pieces of the player will be eligible to enter the inner squares.

In order to reach the finishing square, the players require a precise throw. If the cowry shells show a value larger than the number of remaining squares for a piece to reach the finishing square, that specific piece cannot be moved. In case that is the last piece of a player on the board, the player will not be able to play and the turn will pass to the next player.

3. STRATEGIES

According to Shannon [6], a player's strategy may be defined as a game plan for selecting a move which the player will make at any stage of the game. It is a standard procedure that specifies what decisions a player should make for every possible situation during the game. Without going into game theory, for the particular case of *Cowry* game, we can easily identify the following main types of strategies:

1. Enumerate all possible valid moves in a given position and arrange them according to some criteria. Then, select the first move from the list. This approach always results in the selection of the same move in the same position. Such a strategy is often known as a "pure" strategy.

2. Make a list of all legal moves in a position, and then pick one move completely at random. Clearly, this procedure does not always choose the same move in the same position, hence it is considered to be a “mixed” strategy.

In this section, we propose three distinct playing strategies among which the *fast* and *balanced* strategies come under the category of a pure strategy, whereas the *random* strategy falls into the definition of a mixed strategy.

3.1. Random Strategy

The most elementary strategy to follow is playing randomly. In this strategy, a player always selects a random move out of the possible valid moves during the entire game. A random strategy, of course, has an extremely poor performance. It makes no attempt to choose a good move and hence has no significant advantage for winning games [7].

Our previous analysis of randomness model for playing *Cowry* game [8] demonstrates that the drawbacks of the random strategy seem to outweigh any probable benefits. It also suggests that having any basic strategy could be better than playing randomly. Some of the disadvantages can be highlighted as follows:

- The random strategy is not concerned with important basic knowledge of *Cowry* game, such as hits, threats, first-hit condition, etc.
- There exist trivial strategies in this game which can outperform random strategy, but fail against even the most inexperienced human players.
- The random strategy often results in games that last considerably longer than games involving more sophisticated strategies or human games.

Although playing randomly is not really a strategy, a random-move generator is implemented and used as a benchmark to compare the performance of other proposed strategies throughout this research work.

3.2. Fast Strategy (Moving the Foremost Piece)

In this strategy, a player always chooses to move the piece which has traveled the maximum number of squares around the board from its starting square. The intuitive idea behind moving the “foremost” piece is that the player has used the most number of moves for the advancement of such a piece, and hence it is the most valuable. The elimination of a piece which is closest to the finishing square can greatly impact the player’s progress and the fate of the game. Thus, it should be moved faster than other pieces to reach the finishing square as soon as possible.

Furthermore, giving preference to play with only one piece, while keeping the rest of the pieces in a safe square, may decrease the risk of piece elimination. The foremost piece is, in fact, the only piece which is moving at any given time, being exposed to probable threats. Therefore, as pointed out in [9], this strategy may be viewed as a depth-first approach.

3.3. Balanced Strategy (Moving the Hindmost Piece)

As opposed to the previous strategy, a player may prefer to employ a breadth-first approach by always moving the “hindmost” piece, which is the piece that has covered the minimum distance from its starting square. In this strategy, a player starts the game with any arbitrary piece and keeps on playing with the same piece until it lands on a safe square, based on the cowry shells outcome. Upon reaching a safe square, the player stops proceeding with that piece any further and continues by moving one of the pieces located on the starting square. And so the process is repeated for the new piece. As soon as the active piece reaches a safe

square, the priority of moving is taken away from it and is given to the hindmost piece, closest to the starting square.

This strategy ultimately results in balanced progress of all the player's pieces, moving close to each other and in the form of a cluster. Thus, it may be regarded as a balanced strategy. Using this approach, a player can still benefit from relatively low threat level since only one of the pieces is actively moving on non-safe squares at any instant, facing the risk of elimination. Moreover, the pieces moving together can somewhat reinforce each other in the sense that cautious opponents often hesitate to chase or hit a piece protected by other pieces, because such an attempt will put the attacking piece in danger of elimination.

4. IMPLEMENTATION AND ANALYSIS

In this section, we summarize how the game environment was designed. We then provide a method to identify the valid moves in any given position of the board. Finally, we discuss the design and implementation of the proposed pure strategies.

4.1. Game Setup

In our previous work [8], we implemented the game setup by devising the general algorithm of the game, as well as the required data structures to store the game variables. We also described a method for the generation of random numbers to simulate the throw of four cowry shells. We argued that due to the asymmetrical shape of the shells, the probabilities for the occurrence of different values (1, 2, 3, 4 and 8) are not equal. Thus, we performed the experiment of throwing four cowry shells together for a large number of times and obtained the empirical probability for the occurrence of each outcome. We state these results in Table 1.

Table 1 Empirical probability for the occurrence of outcomes in a throw of 4 cowry shells

Cowry Shells Outcome	1	2	3	4	8
Probability of Occurrence	24.3 %	38.1 %	23.6 %	7.4 %	6.6 %

From this table, we observe that the obtained probability distribution is not uniform. The work [8] demonstrates how we employed a pseudo-random number generator to produce uniformly distributed random numbers, and then transformed them to random numbers with a non-uniform probability distribution that we desired.

In order to validate the accuracy of our implementation and to ensure that all of the game rules are applied properly, we executed numerous automatic games when all four players were making randomly chosen legal moves at each turn. We developed a random-move generator which is capable of identifying the valid moves in a given position and moving one of the pieces at random. The obtained results indicate that all players have an equal chance of winning when they all follow a random strategy. Furthermore, we randomly altered the order of players starting the game to show that there is no correlation between the player taking the first turn and the player winning the game.

4.2. Identification of Valid Moves

As described so far, the first step after simulating the throw of cowry shells is to identify which of the player's pieces can legally play the generated value. All the pieces that can make a valid move are then marked and considered for the further process of piece selection. In order to identify whether a move is valid or not, the program needs to access the current position of each piece, and locate its destination or landing square with respect to the given

value. According to the rules of *Cowry* game, a move is considered to be invalid if it falls into any of the following cases:

1. A piece cannot move beyond the finishing square. Therefore, if the value that the player has to play is larger than the value required for a piece to land on the finishing square, then that piece cannot be moved.
2. A piece cannot enter the inner squares unless the first-hit condition is fulfilled. In other words, if the destination of a piece is in the inner squares while the player has not yet eliminated at least one piece of any of the opponents, then that piece is not eligible to be moved.
3. It is not possible for a player to place more than one piece on any square of the board, of course, with the exception of safe squares. Thus, if the destination of a piece is a non-safe square already occupied by another piece of the same player, then such a move is considered to be invalid.

Figure 3 illustrates the process in the form of a binary decision tree. We may note that in case more than one value is generated due to bonus throws, the process has to be repeated for all available values. However, if none of the pieces of a player is eligible to make a legal move with respect to any of the generated values, then there will be no need for any strategic decision-making, and the turn will automatically pass to the next player.

Figure 3 Identification of valid moves

4.3. Implementation of Strategies

The implementation of the fast strategy (moving the foremost piece) is quite straightforward. All that the program needs to do is to calculate the progress of each piece and assign the same value as the score of the corresponding move. The progress of a piece is defined as the number of squares traveled by the piece from its starting square divided by the total length of the track, which is 24. Hence, the score is given by:

$$\text{Score (piece } i \text{)} = \frac{\text{Distance Covered (piece } i \text{)}}{24}$$

In the fast strategy, the piece closest to the finishing square always gets the highest score. Consequently, the piece which has played the previous turn is probably the same piece which has to play the current turn, while the rest of the pieces are safe on the starting square.

Similarly, the balanced strategy (moving the hindmost piece) is implemented by assigning the scores according to the progress of the pieces but in reverse order, that is, the piece with the minimum progress gets the highest score. However, it may be pointed out that in this strategy, only one piece is supposed to be on non-safe squares at any given time. This calls for an additional condition checking step. The program initially needs to check whether the piece

which has played the previous turn is still located on a non-safe square or not. If so, the highest score (any value greater than one) is assigned to the same piece regardless of its progress. Otherwise, if all the pieces are positioned on safe squares, the score is calculated by:

$$Score (piece i) = 1 - \frac{Distance Covered (piece i)}{24}$$

It may be relevant to mention that the foremost and hindmost strategies merely deal with the knowledge of players about their own pieces, without taking the opponents into account. The basic versions of these algorithms when constructed in this fashion, essentially ignore the existence of the opponents' pieces on the board, and therefore may miss the opportunity of eliminating them. The piece which has received the highest score on the basis of its progress may accidentally hit an opponent's piece out of pure luck. The rest of the pieces, however, will never attempt to attack any of the opponents' pieces even if a suitable cowry shells outcome is obtained. It can be justified that moving a piece, which is supposed to be idle, out of a safe square to eliminate an opponent's piece may result in having more than one piece on non-safe squares, being exposed to probable threats. This would be in direct contradiction to the primary logic behind these strategies.

5. EXPERIMENTATION AND RESULTS

In this section, we describe the tests performed on the fast and balanced strategies. We evaluate the obtained experimental results using the "winning percentage" as the performance measure. Following the approach introduced in [9], we conducted the experiments in two separate phases, with 10,000 game runs for each test:

1. Testing the individual performance of each pure strategy playing against three random players.
2. Testing the relative performances of both pure strategies playing against each other in the same game.

In the first phase, we observed the performance of each pure strategy individually in a game when the other three players were merely making random moves. From the obtained results shown in Table 2, we can observe that the chance of winning for all four players is equal (25% wins) when they all follow the random strategy. Thus, we consider the random strategy as a baseline for the comparison of other proposed strategies. The performances of the fast strategy (51% wins) and the balanced strategy (57% wins) were significantly better than random players, although they did not possess any knowledge of the opponents' pieces on the board. This test clearly demonstrates that a high winning rate against random players can be achieved, simply by giving preference to move one piece at a time while keeping other pieces in safe squares, and hence decreasing the risk of piece elimination. Figure 4 provides a graphical view of the test results.

Table 2 Winning percentage of strategies against 3 random players

Player 1	Player 2	Player 3	Player 4
Random 25.0 %	All Random 25.0 % for each		
Fast 51.4 %	All Random 16.2 % for each		
Balanced 57.1 %	All Random 14.3 % for each

Figure 4 Winning percentage of strategies against 3 random players

For the second phase of experiments, we attempted to evaluate the relative performances of the proposed pure strategies, by allowing them to play against each other in the same game. We executed a large number of automated four-player games, in which two of the players were following the fast strategy and the other two players were employing the balanced strategy. We examined how many times each strategy could win and calculated the winning percentage. Table 3 and Figure 5 highlight the obtained results. It can be observed that the balanced strategy (53.8% wins) managed to outperform the fast strategy (46.2% wins). These results clearly demonstrate the advantage of the balanced strategy over the fast strategy, and the purposelessness of the random strategy against our proposed pure strategies.

Table 3 Winning percentage of the pure strategies against each other in the same game

Strategy	Fast	Balanced
Winning Percentage	46.2 $\pm$ 0.5 %	53.8 $\pm$ 0.5 %

Figure 5 Winning percentage of the pure strategies against each other in the same game

6. CONCLUSIONS AND FUTURE WORK

In this research, we introduced *Cowry* game, a multi-player, imperfect information, strategic race game from India. We also described our motives to develop an artificial player for this game. We classified the playing strategies into two major categories, pure and mixed strategies, and accordingly proposed two basic pure strategies: the fast strategy (moving the foremost piece) and the balanced strategy (moving the hindmost piece). Moreover, we discussed a method for identification of valid moves in any board position, and provided details about the implementation and analysis of the proposed strategies. The experiments were conducted in two independent phases to evaluate the individual performances of the pure strategies against random players, as well as their relative performances against each other. The test results demonstrated that both of the proposed strategies could easily outperform the random players, with the balanced strategy performing slightly better than the fast strategy.

For future work, the basic strategies presented here can be used for evolving better strategies and improving the gameplay. The fast and balanced strategies are weak in playing skill because they possess very limited knowledge of the opponents, and hence lose many opportunities to defeat them. More advanced strategies can be constructed by taking more detailed knowledge of the game into consideration. Furthermore, it is possible, and perhaps advantageous, to combine several pure strategies in order to obtain better artificial players. Changing the playing strategies intelligently at different stages of the game can also result in higher performance, and therefore might be a suitable topic for further research.

REFERENCES

- [1] Gerald Tesauro, Temporal Difference Learning and TD-Gammon, ACM, 38(3), 1995, pp. 58-68.
- [2] M. Campbell, A. J. Hoane, F. Hsu, Deep Blue, Artificial Intelligence, 134(1-2), 2002, pp. 57-83.
- [3] Schaeffer J. et al., Checkers is Solved, Science, 317(5844), 2007, pp. 1518-1522.
- [4] I. Millington, J. Funge, Artificial Intelligence for Games, 2nd Ed., Morgan Kaufmann Publishers, San Francisco, USA, 2009, pp. 667-670.
- [5] David S. Parlett, The Oxford History of Board Games, Oxford Univ. Press, Oxford, New York, 1999, pp. 34-106.
- [6] Claude E. Shannon, Programming a Computer for Playing Chess, Philosophical Magazine, 41(314), 1950, pp. 265-275.
- [7] Imran Ghory, Reinforcement Learning in Board Games, Technical Report CSTR-04-004, CS Dept., Univ. of Bristol, 2004.
- [8] P. Nagabhushan, Pouyan Davoudian, Machine as One Player in Indian Cowry Board Game: Strategies and Analysis of Randomness Model for Playing, International Journal on Recent and Innovation Trends in Computing and Communication, 5(2), 2017, pp. 57-67.
- [9] Faisal Alvi, Moataz Ahmed, Complexity Analysis and Playing Strategies for Ludo and its Variant Race Games, IEEE Conference on Computational Intelligence and Games (CIG'11), 2011.