A Deep Reinforcement Learning Approach to The Ancient Indian Game - Chowka Bhara

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

10-10-2021 / 12-10-2021

CITATION

Patil, Annapurna P; RAGHAVENDRA, SANJAY; Srinarasi, Shruthi; Ram, Reshma (2021): A Deep Reinforcement Learning Approach to The Ancient Indian Game - Chowka Bhara. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.16780414.v1>

DOI

[10.36227/techrxiv.16780414.v1](https://doi.org/10.36227/techrxiv.16780414.v1)

A Deep Reinforcement Learning Approach to The Ancient Indian Game - *Chowka Bhara*

Annapurna P Patil
Ramaiah Institute of Technology
Bangalore, India
annapurnap2@msrit.edu

Shruthi Srinarasi
Ramaiah Institute of Technology
Bangalore, India
shruthi223@gmail.com

Sanjay Raghavendra*
Ramaiah Institute of Technology
Bangalore, India
sanjay14112000@gmail.com

Reshma Ram
Ramaiah Institute of Technology
Bangalore, India
reshmaram2000@gmail.com

Abstract—Reinforcement Learning (RL) is the study of how Artificial Intelligence (AI) agents learn to make their own decisions in an environment to maximize the cumulative reward received. Although there has been notable progress in the application of RL for games, the category of ancient Indian games has remained almost untouched. *Chowka Bhara* is one such ancient Indian board game. This work aims at developing a Q-Learning-based RL *Chowka Bhara* player whose strategies and methodologies are obtained from three Strategic Players viz. Fast Player, Random Player, and Balanced Player. It is observed through the experimental results that the Q-Learning Player outperforms all three Strategic Players.

Keywords—Reinforcement Learning, *Chowka Bhara*, Strategic Player, Q-Learning Player, Indian Board Games

I. INTRODUCTION

Chowka Bhara is a two to four-player board game [1] that is similar to the popular board game Ludo. This game is an example of a “fully observable” system with an element of chance introduced by the roll of special dice (cowry shells) and an element of strategy. The game is controlled by tossing the four cowry shells and counting how many fall 'as it is' versus those that land 'inverted'. The game's objective is for the player to get all their four pawns to the innermost square on the board (seen in Fig. 1).

Artificial intelligence in games [2] involves perception and decision-making in game environments. The two main challenges of applying AI in games are -

- The state-space of the game is very large and;
- Learning policies to make decisions in a dynamic unknown environment is difficult.

However, Reinforcement Learning (RL) [3] algorithms like Q-Learning (QL) [4] help overcome these challenges to a certain extent.

Being one of the three basic paradigms of Machine Learning (ML), RL is the study of how intelligent agents make decisions in an environment to maximize their cumulative reward. It is a policy-based and reward-based system. Both reinforcement learning and deep learning have been used previously and have delivered great results in gaming, computer vision, Natural Language Processing (NLP), etc.

RL and Deep Reinforcement Learning (DRL) have been observed to perform exceedingly well in games [5][6][7][8]. For modern games, a lot of focus has been on very difficult competitive games like StarCraft [9], and they have also achieved great results from DeepMind and OpenAI. Although several categories of games have been implemented using AI, one category that has been neglected is the regional, ancient Indian board games. Through this work, a novel RL approach for a traditional Indian game such as *Chowka Bhara* is introduced that helps to bridge the gap between ancient games and modern artificial intelligence methods.

Fig. 1. Traditional *Chowka Bhara* board

A. Motivation

Although there has been increasing use of reinforcement learning methods in gaming, the domain of ancient Indian games [10] has not been explored enough. This work aims to design a QL player that can successfully play *Chowka Bhara* which helps highlight Indian history and culture and forms a stepping stone to more automation of other unexplored ancient Indian games like Indian Ludo [11].

Chowka Bhara is partially a game of chance but it also involves critical thinking and careful planning. The game also enhances counting skills. Further, the *Chowka Bhara* game can also be extended to modeling real-life problems since the game involves stochastic components like decision-making

under uncertainty, a problem most people face in the real world.

II. RELATED WORKS

Deep Reinforcement Learning (DRL) [12] combines reinforcement learning and artificial neural networks to design autonomous systems capable of a higher-level understanding of the environment. The training of agents to play games acts as a precursor in developing systems to adapt to the real world. Such techniques can be very beneficial in developing RL agents for games such as *Chowka Bhara* and many more. However, developing DRL methods that can adapt rapidly to new tasks is a significant challenge in RL. Previous works have shown that recurrent networks can support meta-learning in a fully supervised context, but this approach can be extended to an RL setting. Deep meta reinforcement learning [13], one of the possible solutions for the aforementioned tasks, includes the Two-Step Tasks Problem, which does not require huge training sets to develop a model and hence it is not as dependent on the past as other methods. This helps the model adapt to new tasks rapidly, leading to a robust RL agent.

The main goal of RL is to find a sequence of steps in a sequential decision problem that can obtain a maximum cumulative reward. QL is one of the most popular algorithms to achieve this goal, but QL with Deep Neural Networks often suffers from overestimations. *Double Q-Learning Algorithm* [14] solves this to a great extent. Two value functions are learned by assigning each experience randomly to update one of the two value functions, such that there are two sets of weights. Such Deep Q Networks (DQN) are possible solutions to reduce overestimations in RL implementations of games such as *Chowka Bhara*.

Due to the inherent similarities in the rules and strategies of Ludo with those of *Chowka Bhara*, concepts of Q-learning and TD-based players [15] can be closely followed in implementing the latter. TD-based players choose the next step by considering the value functions of each state, whereas Q-Learning-based players use action-value quality pairs to choose the next step from the given/current step. Each position is assigned by a real number normalized by taking an average of 4; this number indicates the number of players in that particular position.

III. RULES AND STRATEGIES OF THE GAME

Chowka Bhara is a two to four-player 5x5 board game similar to the popular board game Ludo. Four *cowry* shells act as the dice in this fully observable system where the moves made by the players depend on the outcome of throwing the shells, i.e., the number of shells that fall inverted as opposed to that of the shells that fall “as it is”. Players take turns in throwing the shells with possible outcomes being 1, 2, 3, 4 and 8 with probabilities {0.243, 0.381, 0.236, 0.074, 0.066} respectively [16]. If all the four shells land inverted, it is known as *chowka* (which translates to four), and if they all land ‘as it is’ it is termed as *bhara* (which translates to eight), a shell landing inverted is counted as 1, 0 if it lands ‘as it is’.

If the throw is 4 or 8, the player gets an extra throw. On throwing the shells, depending on the value of the outcome, the player must choose which action to take from a set of valid moves, limiting the different moves possible in the game. Such valid moves are as follows:

- A pawn can enter the inner square ring only if at least one opponent’s pawn has been defeated.
- At any given instant during the game, only one pawn should be present in a non-safe square. Safe squares have no such constraints. Two or more pawns of the same player cannot be placed in the same non-safe square, and two pawns of opposing players in the same square lead to the dismissal of the pawn that was previously present in that square.
- If a pawn reaches the innermost square, no other moves are possible. Consequently, if the outcome of a throw exceeds the number of boxes between the pawn’s current position and the innermost square, the move cannot be made.

The game begins with all pawns in the players’ respective ‘home’ squares, as seen in Fig. 2. The player whose all four pawns move to the center square the fastest wins the game. Additionally, the game board has an outer square ring that consists of 4 safe squares marked with an ‘x’ where none of the pawns in these boxes can be “captured” by those of the opponents. In any of the other squares at a time, only one player’s pawn can be present. On “capturing” an opponent’s pawn, the player also gets an extra throw. Further, capturing an opponent’s pawn sends the captured pawn back to its home square. A player can enter the inner square only after tracing the entire outer ring and defeating at least one opponent’s pawn. The players navigate their pawns in an anticlockwise direction throughout the game, first through the outer square, then through the inner square in the opposite direction.

Fig. 2. Red dashed lines indicate the path taken by the red pawns, and blue dashed lines indicate the path taken by blue pawns in the game.

Strategies in the game of *Chowka Bhara* determine which pawn should be moved next. To increase the winning probabilities of a player, various strategies can be adopted. Such strategies include fast players, random players, and balanced players. The fast player is an Strategic player that,

given the outcome of a throw, chooses the pawn that is farthest from the starting box of the player and closest to the innermost square, given that the pawn has at least one remaining legal move. This player chooses one pawn to move and chooses it until it has reached the goal state resembling the depth-first search. The random player is an Strategic player that, given the outcome of a throw, randomly chooses one among the four pawns to move with a probability of 0.25. All the pawns of a player have equal probabilities (0.25) of being chosen. In a scenario where two pawns have reached the innermost square, one among the other two remaining pawns is chosen with a probability of 0.5 each. The balanced player is an Strategic player who chooses the pawns to move according to the breadth-first search strategy given the outcome of a throw.

IV. PROPOSED Q-LEARNING PLAYER

A. Q-Learning Player

Q-Learning is a model-free, off-policy algorithm that aims to learn those policies that will lead to the highest cumulative reward. Rather than using greedy techniques to estimate the value function, Q-Learning is value-based and updates the value function using mathematical equations like the Bellman equation given below:

$$V(s) = \max_a (R(s, a) + \gamma \sum_{s'} P(s, a, s') V(s')) \quad (1)$$

where s is the current state, s' is the state after taking action a and, γ is the discount factor.

a) State Representation

Chowka Bhara uses a 5x5 board, i.e., there are 25 different squares on the board. There can be up to 4 pawns per player on each square. To reduce the computation time and make the agent's training simpler, raw representations of the states are chosen. Each square on the board is marked as a real number indicating the pawns on the given square. Further, which player's turn is also recorded, and two unary inputs are added for this. Since there are two players, the total number of states will be $(25*2) + 2 = 52$ states.

b) Rewards

The rewards designed to maximize the winning rate of the RL agent are as follows:

- -1 for moving to a non-safe square
- +1 for moving to a safe square
- +5 for capturing an opponent pawn
- -5 for getting a pawn captured
- +10 for getting one pawn to the innermost square
- +100 for winning the game, i.e., getting all four pawns to the innermost square
- -100 for losing the game

The positive rewards encourage the agent to perform those actions, whereas the negative rewards penalize the agent.

B. Experimental Design

The QL Player is made to play against each Strategic Player viz. Random Player, Fast Player, and Balanced Player, and the performance of the players are evaluated. The players are

trained for 10,000 episodes, each using a twenty hidden-layer neural network. A predominant problem in RL is the exploration-exploitation dilemma. To balance exploration and exploitation, the Epsilon-Greedy algorithm is used, taking the value of Epsilon as 0.9. This method randomly chooses either exploration or exploitation and ensures that the exploration-exploitation problem is taken care of.

V. RESULTS

To test the working of the Q-Learning (QL) Player, each of the Strategic Players - Fast Player, Random Player, and Balanced Player - are made to play against the QL Player. As seen in Fig. 3, when trained using 10,000 games, the QL Player outperforms all three Strategic Players. The QL Player's winning percentage against the Random Player, Fast Player, and Balanced Player is observed to be 88.01%, 72.64%, and 70.77% respectively.

Fig. 3. Winning percentage of QL Player and each Strategic Player against each other

Fig. 4 compares the winning percentages of each Strategic Player against the QL Player. The number of games the players are trained using starts at 1,000 games and increases in increments of 1,000 up to 10,000 games each. It is observed that the Balanced Player has the highest winning rate, closely followed by the Fast Player.

Fig. 4. Winning percentages of each Strategic Player against the QL Player

A. QL Player vs. Fast Player

Fig. 5 represents the number of games on the x-axis and the corresponding winning percent of both the QL Player and the Fast Player on the y-axis. The number of games the players

are trained using starts at 1,000 games and increases in increments of 1,000 up to 10,000 games each. It is observed that when the players are trained using a sufficient number of games (greater than 4,000 training games), the QL Player performs and, if not outperforms, the Fast Player.

Fig. 5. Winning percentage of the QL Player against the Fast Player

B. QL Player vs. Random Player

Fig. 6 represents the number of games on the x-axis and the corresponding winning percent of both the QL Player and the Random Player on the y-axis. The number of games the players are trained using starts at 1,000 games and increases in increments of 1,000 up to 10,000 games each. It is observed that QL Player consistently outperforms the Random Player irrespective of the number of training games.

Fig. 6. Winning percentage of the QL Player against the Random Player

C. QL Player vs. Balanced Player

Fig. 7 represents the number of games on the x-axis and the corresponding winning percent of both the QL Player and the Balanced Player on the y-axis. The number of games the players are trained using starts at 100 games and increases in increments of 100 up to 1000 games each. It is observed that when the players are trained using a sufficient number of games (greater than 5,000 training games), the QL Player gradually outperforms the Balanced Player.

Fig. 7. Winning percentage of the QL Player against the Balanced Player

VI. CONCLUSION

By incorporating artificial intelligence in ancient games, the work proposed in this work brings the past closer to the present. The work focuses on three Strategic Players viz. Random Player, Fast Player, and Balanced Player for the traditional Indian game *Chowka Bhara*. Further, a Q-Learning Player based on the methodologies and strategies of the developed Strategic Players is proposed. It is observed from the experimental results that, when trained sufficiently, the Q-Learning Player consistently outperforms each of the three Strategic Players. Our objective of designing an RL agent that can successfully play *Chowka Bhara* can be significant in future implementations of many more Indian games like Indian Ludo. It can solve problems involving stochastic components and decision-making under uncertainty.

FUTURE WORKS

This work explores only Q-Learning algorithms for the development of the RL agent. However, future scope of this work can include the design of an agent using algorithms such as the Temporal Difference Lambda (TD- λ). Further, to enhance the performance of the agent, planning algorithms such as Monte Carlo Tree Search can be employed.

ACKNOWLEDGMENT

This research was supported and funded by the M.S. Ramaiah Institute of Technology Alumni Association. The authors would like to thank Ramaiah Institute of Technology for providing the facilities and means to enable the research.

REFERENCES

- [1] Roll the Dice. 2021. Chowka Bara builds energy, strategic thinking and family bonding. [online] Available at: <<https://www.rollthedice.in/chowka-bara/>> [Accessed 1 July 2021].
- [2] Yildirim, Sule, and Sindre Berg Stene. "A survey on the need and use of AI in game agents." InTech, 2010.
- [3] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.
- [4] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8, no. 3-4 (1992): 279-292.

- [5] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).
- [6] Lample, Guillaume, and Devendra Singh Chaplot. "Playing FPS games with deep reinforcement learning." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1. 2017.
- [7] Dedieu, Antoine, and Jonathan Amar. "Deep reinforcement learning for 2048." In Conference on Neural Information Processing Systems (NIPS), 31st, Long Beach, CA, USA. 2017.
- [8] Block, Marco, Maro Bader, Ernesto Tapia, Marte Ramírez, Ketill Gunnarsson, Erik Cuevas, Daniel Zaldivar, and Ral Rojas. "Using reinforcement learning in chess engines." Research in Computing Science 35 (2008): 31-40.
- [9] Shao, Kun, Zhentao Tang, Yuanheng Zhu, Nannan Li, and Dongbin Zhao. "A survey of deep reinforcement learning in video games." arXiv preprint arXiv:1912.10944 (2019).
- [10] Penn.museum. 2021. Expedition Magazine - Penn Museum. [online] Available at: <<https://www.penn.museum/sites/expedition/the-indian-games-of-pachisi-chaupar-and-chausar/>> [Accessed 19 July 2021].
- [11] Live History India. 2021. Ludo's Indian Roots. [online] Available at: <<https://www.livehistoryindia.com/story/forgotten-treasures/ludos-indian-roots/>> [Accessed 19 June 2021].
- [12] Arulkumaran, Kai, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. "A brief survey of deep reinforcement learning." arXiv preprint arXiv:1708.05866 (2017).
- [13] Wang, Jane X., Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Remi Munos, Charles Blundell, Dhharshan Kumaran, and Matt Botvinick. "Learning to reinforcement learn." arXiv preprint arXiv:1611.05763 (2016).
- [14] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1. 2016.
- [15] Alhajry, Majed, Faisal Alvi, and Moataz Ahmed. "TD (λ) and Q-learning based Ludo players." In 2012 IEEE Conference on Computational Intelligence and Games (CIG), pp. 83-90. IEEE, 2012.
- [16] Davoudian, Pouyan & Nagabhushan, P.. (2019). "Machine as One Player in Indian Cowry Board Game: Basic Playing Strategies." 10. 174-183. 10.34218/IJCET.10.1.2019.019.