Artificial Intelligence Approach for Modeling Optimal Shot Sequences in Pool

Christopher Dargan
NET ID: cdargan2 KSU ID: 000946617
cdargan2@students.kennesaw.edu
Kennesaw State University

ABSTRACT

Pool or billiards is a dynamic category of cue sport involving a blend of skill, strategy, and precision. Mastery of the game involves navigating complex physics interactions, understanding geometric principles, and strategic shot planning to outmaneuver opponents. This project addresses these challenges by harnessing artificial intelligence techniques to develop a model capable of predicting optimal shot sequences, thereby enhancing players' chances of success to win. While executing accurate and consistent shots demands significant skill, the ability to identify optimal effective shots is as equally crucial. The proposed AI model aims to assist both novice and experienced players by offering insights into optimal runout patterns. For beginners, it serves as a valuable tool for learning the game, providing guidance on strategic shot selection and fundamental skill development. Advanced players can leverage the model to analyze potential patterns, refining their tactical approach and enhancing their gameplay. Moreover, the application of this AI model extends beyond player assistance. It enhances the viewing experience for spectators and commentators during live pool matches. By predicting and projecting possible shots that players might take, it adds depth to commentary, enriching the overall viewing experience and engaging audiences in the strategic intricacies of the game. In summary, this project represents a significant advancement in the integration of artificial intelligence within the game of pool, offering benefits for players, enthusiasts, and spectators alike.

1. Introduction

Pool presents a formidable challenge to players seeking mastery since it not only involves skill but a necessity of understand strategy. Beyond the mechanics of executing shots lies the strategic puzzle of discerning which shots to take to secure victory. Within this strategic decision-making process lies the motivation behind our project. Our goal is to develop an AI model capable of predicting optimal shots, thereby providing players with invaluable insights into unseen runout patterns and enhancing their chances of success on the pool table.

As pool is a challenging sport to breach into due to the complexities of not only performing shots with precision but also knowing how to strategically navigate the sequence of shots required to achieve victory. In the traditional game of 8 ball pool, players are tasked with pocketing all balls belonging to their selected group and concluding the game by pocketing the 8 ball. Success is strongly correlated to the players ability to consecutively pocket balls, thereby denying opponents opportunities to seize control of the table.

1.1 Problem Statement

However, determining the optimal sequence of shots among thousands of possible combinations is not an easy task. While some combinations may appear favorable, the true measure of success lies in the ability to consistently execute a sequence with minimal risk of surrendering possession. This underscores the central point of the problem we seek to address: How do we analyze the state of a pool game to determine the next optimal shot?

Central to this challenge is the need to dissect and model the intricate interactions and decision-making processes inherent in achieving an optimal shot in pool. This task involves expertise domain knowledge of the game's physical dynamics and principles; it necessitates consideration of strategic factors such as positioning, shot selection, etc. By tackling this problem, our project aims to provide players, from novices to advanced players, with a powerful tool for enhancing their gameplay and strategic understanding. Through the development of an AI model capable of predicting optimal shots, we aspire to empower players to make informed decisions on the pool table, ultimately elevating their performance and enjoyment of the game.

---

2. Literature Review

Within pool, the integration of artificial intelligence (AI) techniques hold the potential for providing valuable insights, especially with the rapid growth in AI development in recent years. Researchers and enthusiasts have increasingly explored various methodologies aimed at enhancing gameplay, strategic decision-making, and overall player performance. Despite the growing interest in AI-driven analysis techniques, there remains a noticeable lack in the literature regarding the prediction of optimal shot runout patterns. This section looks to partially fill this gap by offering an overview of relevant studies within the field of AI-driven pool analysis and prediction.

One noteworthy research explored a similar approach to the one undertaken in this project by employing reinforcement learning techniques to analyze and predict optimal shot sequences in pool games. The study delved into the integration of a robot equipped with the ability to play pool [2]. Central to their approach was the implementation of planning algorithms for shot selection, considering factors such as shot angle, tolerance, stroke power, and more [2]. While its critical to our project to predict the sequence of shots it was also pertinent to be able to predict the movement of the cueball to simulate the path the cueball with take

In the task of simulating cue ball movement, researchers have developed comprehensive models to predict the projected path of the cue ball. For instance, paper [1] calculated the cue ball's trajectory by accounting for various factors such as initial velocity, post-collision velocity, elastic collision equations, force, speed, restitution factor, friction, attack angle, and deflection angle. This research provides valuable insights into the physical interactions involved in cue ball movement, informing our approach to simulating shot outcomes in the simulated pool game environment.

2.1 Proposed Algorithm

Building upon the understanding from previous research, our project proposes a methodology to simulate the physical interactions of shooting a pool ball, thereby generating possible states of a pool game position. From these simulated based states we aim to utilize a reinforcement learning approach.

We intend to score these states based on weighted criteria informed by domain expertise in pool principles. Subsequently, we will employ a search tree algorithm, augmented by heuristic-based search techniques, to explore the search space of potential shot sequences and identify optimal runout patterns.

Through the application of AI-driven analysis and simulation techniques, our proposed algorithm aims to offer players invaluable insights into strategic shot selection, enhancing their gameplay experience and strategic proficiency on the pool table.

3. Methodology

This project's methodology consists of three major phases: derive possible shots, shot simulation, and optimal pattern search. Figure 1 depicts a snapshot of our pipeline for modeling optimal shot sequences. The model generates a pool game environment where the cueball and object balls can be defined. In this environment we determine all possible shots within our scope. The generated possible shots are then passed into our simulator where the shot is taken, and a new position is generated. Finally, this generated state is given an evaluation score which is used as a heuristic in our optimal pattern search algorithm.

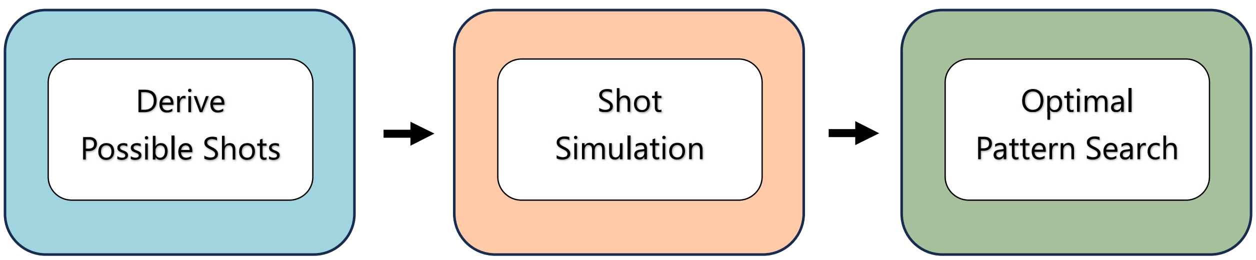


Figure 1

4. Implementation

Within this section we will cover the implementation of our project's methodology pipeline which consists of the three phases derive possible shots, shot simulation, and optimal pattern search.

4.1 Derive Possible Shots

In order to derive all the shots that are possible from the position of the cueball, we establish a pool game environment where the cueball, object balls, pockets, and table can be defined and visualized in our GUI. To begin analyzing possible shots, we iterated through all the appropriate object balls in respect to the cueball. For each object ball we consider the possibility of shooting that ball in each pocket. For each pocket we calculate the contact point needed for the cueball to pocket the object ball in the respective pocket. Once a contact point is established, we determine if the path from the cueball to the object ball is obstructed. To avoid unnecessary calculations, we devised an algorithm to determine all the potential obstacle balls within the boundary of the contact point and the cueball. A boundary box can be seen in Figure 2 regarding the potential balls in regards to the object ball and pocket. Next, we cycle through the potential obstacle balls determining if the cueball has enough tolerance to pass by each ball to make a successful contact. This is done by defining the shot line from the cueball to the contact point and for each obstacle ball we find the distance from the obstacle ball's perpendicular intersection point of the shot line.

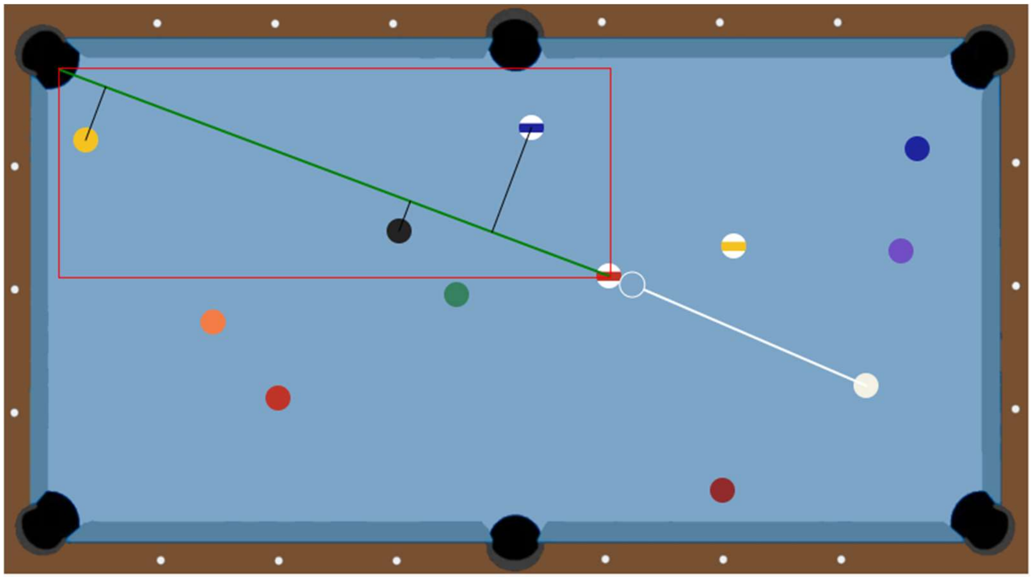


Figure 2

A similar process is done for the path of the object ball to the pocket. This process can be seen in Figure 2 for the object ball and pocket. If both paths are found to be unobstructed, we continue by determining if the shot is possible by considering the tangent line of the contact point and the cut angle of the shot. If all three requirements are met the shot is determined to be possible.

4.2 Shot Simulation

The purpose of the simulator is to approximate the interactions of the cueball and object balls after a shot is made. To start the simulator, we need to initialize the cueballs velocity towards the desired contact point determined in the previous phase. The formulas below show how we calculate the initial cueball velocity.

$$\vec{X} = x_{contact\ point} - x_{cueball}$$

$$\vec{Y} = y_{contact\ point} - y_{cueball}$$

$$||V|| = \sqrt{(\vec{X})^2 + (\vec{Y})^2}$$

$$|\vec{X}| = \frac{\vec{X}}{||V||}$$

$$|\vec{Y}| = \frac{\vec{Y}}{||V||}$$

$$V_x = |\vec{X}| * power$$

$$V_y = |\vec{Y}| * power$$

With the cueball's velocity established we run the simulator. The simulator runs for 6000 ticks. To emulate time passing between ticks we took the average time elapsed of the framerate for 6000 frames which resulted in approximately 0.00416 seconds based on the device the simulation was ran on. For each iteration of the simulator, we update all the balls position based on their respective velocities, collisions with other balls, collision with the table's rails, and friction of the balls against the table. At the end of the simulation, we get the state of the pool table after the shot is made, this can be seen in Figure 3 showing the trajectory path took by the cueball during the simulation.

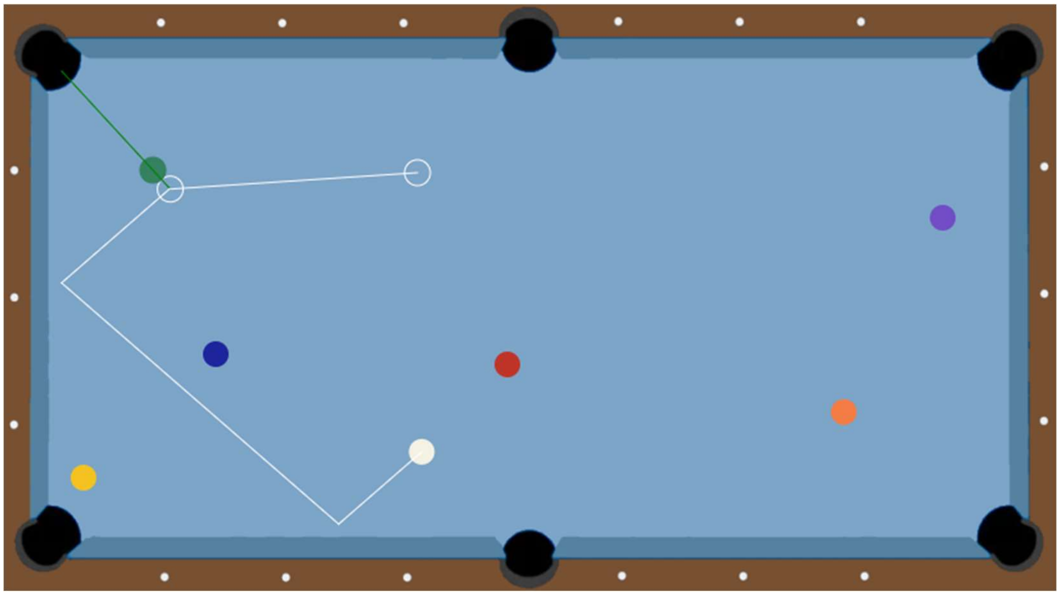


Figure 3

4.3 Optimal Pattern Search

Once we have our generated pool state, we need to evaluate the position to give it a score to measure how good or bad the position is. This is where the domain knowledge of pool principles come into perspective. The score is determined by the sum of metrics measured multiplied by the metrics weight respectively. With this approach the model can evaluate a position by the predefined metrics and how much impact that metric has on towards the position. Some of the metrics consider is the cut angle of the shot, shot distance, pocket distance, what object balls are available after the shot is made, was the object successfully pocketed, etc. Below is a breakdown of the score for evaluated for the position of Figure 3 in Table 1.

Metrics	Evaluated Score
Cueball Centeredness	0.740
Missed Shot	0.000
Scratch Cueball	0.000
Pocket Distance	0.509
Shot Distance	0.650
Cut Angle	0.720
Available Pockets Next Shot	0.089
Post Collisions	0.000
Total Score	2.708

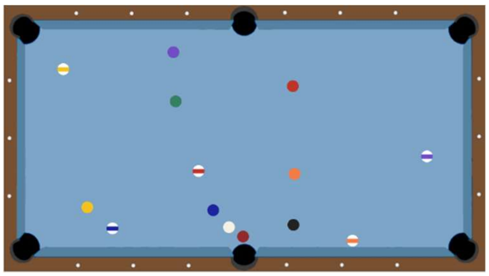
Table 1

Now that we can evaluate a position, we implemented a heuristic based search algorithm to find a cost optimal path resulting in an optimal sequence of shots leading to the goal state positions which is the pocketing of the 8 ball. Below is pseudocode for the proposed search algorithm used.

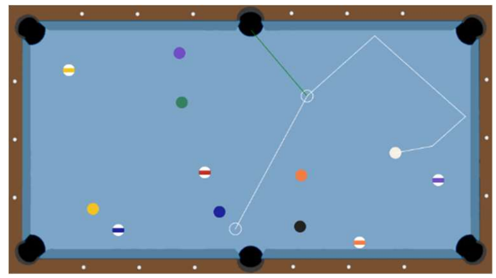
```
ALGORITHM    findOptimalRunOut( Node root )
    // Initialize data structures
    frontier = PriorityQueue()
    closed_set = HashSet()
    paths = HashMap()
    edge_cost = HashMap()

    // Add root to queue
     frontier.push(root)

         // Search for goal state
    while frontier is not empty:
        // Get the highest priority node in frontier
        current = frontier.pop()

        // Check if node is the goal state
         if current is the goal state:
            // Retrace path from goal state to root
            return reconstructPath(current, paths)

        // Add visited node to closed set
        closed_set.add(current)

        // Expand current node
        current.expand()

        // Iterate through child nodes
        for each child_node in current.child_nodes:
            // Skip if child node is already in closed set
            if child_node is in closed_set:
                continue

            // Calculate total edge cost
            total_edge_cost = current.cost + edge_cost(current, child_node)

            // Update path and cost if necessary
            if child_node not in frontier or total_edge_cost > child_node.cost:
             paths[child_node] = current
             child_node.cost = total_edge_cost
                frontier.push(child_node)

   // No path was found
   return null
```

5. Experimental Results

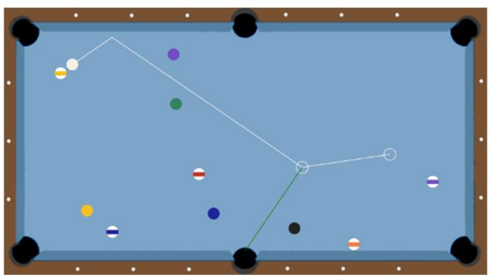
Based on our implemented methodology and pipeline we found that our model could successfully predict optimal shot sequences resulting in a runout pattern. The model outputs the trajectory path of the cueball, the order of which object balls to pocket, which pocket to aim respectively, and how much power to use. Below shows an example of the model running out the pool table given an initial pool table position.



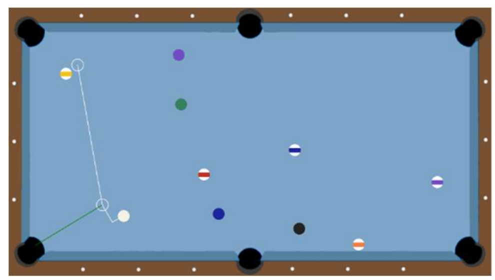
[Initial Position]



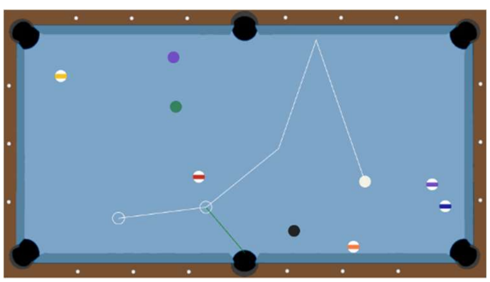
[3 Ball, pocket-2, 0.75%]



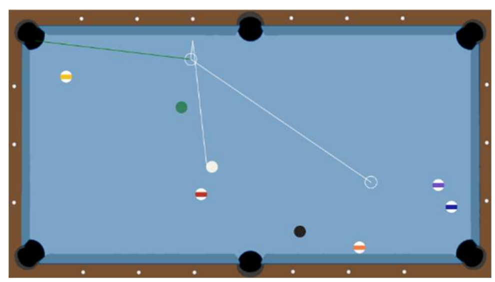
[5 Ball, pocket-5, 0.75%]



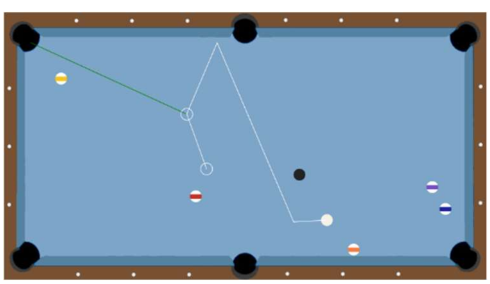
[1 Ball, pocket-4, 100%]



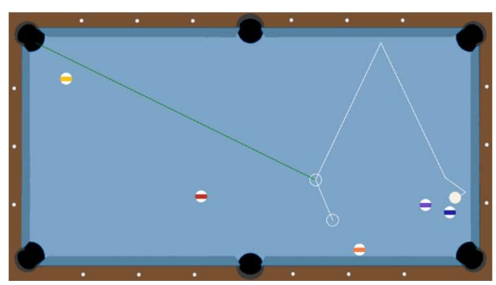
[2 Ball, pocket-5, 100%]



[4 Ball, pocket-1, 100%]



[6 Ball, pocket-1, 100%]



[8 Ball, pocket-1, 100%]

6. Advantages & Drawbacks

The first advantage to consider is the approach of simulation, this is beneficial because it allows use to model a more accurate representation of the physical interactions within the game of pool compared to other techniques. Another benefit of our proposed methods is the adaptability of the model's output sequences. Since the model decisions are justified by the metrics passed, the model's decision making can be adjusted and tailored to certain play styles and skill ranges. Lastly, our model has a fast execution time for deriving possible shots, simulations all shots, and pattern searching. This allows us for applications to utilize our predicted shot sequences in near real time.

Some drawbacks of our approach is the simplification of the game of pool we took. For instance, a big simplification was cueball and object ball collisions and interactions. In our simulation the collisions were calculated with perfect elastic collisions which allows use to calculate the trajectory easier since it would be based on the collision tangent line. However, in reality this interaction is more complex. This complexing allows the cueball to travel in more than just the tangent line. By adding spin to the cueball, this unlocks a tremendous amount of possibilities regarding positions of the cueball. In addition, our model does not take into account kick shots, bank shots, and combination shots. While our model does not consider these shots our range of possible runout sequences are shrunken since they are still valid options within the real game of pool.

---

7. Summary & Conclusion

In summary, this project presents an unique approach to predicting optimal runout patterns in the game of pool through the integration of artificial intelligence (AI) techniques. By utilizing reinforcement learning, simulation methods, and heuristic-based search algorithms, the proposed model offers players invaluable insights into strategic shot selection and gameplay optimization. Throughout the literature review, we explored the growing interest in AI-driven pool analysis and the potential applications of advanced computational techniques in enhancing player performance and strategic decision-making. Despite a notable gap in research specifically addressing the prediction of optimal shot runout patterns, existing studies provided valuable insights and foundational knowledge for the development of our approach. Through our proposed algorithm, outlined by the three major methodology phases: derive possible shots, shot simulation, and optimal pattern search, players gain a competitive edge on the pool table, allowing them to make informed decisions and optimize their chances of success.

Looking ahead, the potential for real-time application of shot sequences during live pool matches or training is a useful tool for beginners, advanced players, and spectators alike. In addition, ongoing innovation and refinement hold promise for further enhancements towards the algorithm's capabilities and utility. As pool continues to evolve, AI based analysis stands promising potential into strategic gameplay and player performance in the world of pool.

In conclusion, the development of the proposed algorithm represents an advancement in the integration of AI techniques within the game of pool. By providing players with more indepth analytical tools and strategic insights, the algorithm not only enhances gameplay experience but also contributes to the ongoing evolution and innovation of the sport. As players embrace AI-driven analysis as a valuable resource for strategic decision-making, the future of pool promises to be both exciting and transformative.

References

[1] Shih, Chihhsiong. (2014). Analyzing and Comparing Shot Planning Strategies and Their Effects on the Performance of an Augment Reality Based Billiard Training System. International Journal of Information Technology & Decision Making. 13. 10.1142/S0219622014500278.

[2] T. Nierhoff, K. Heunisch and S. Hirche, "Strategic play for a pool-playing robot," 2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), Munich, Germany, 2012, pp. 72-78, doi: 10.1109/ARSO.2012.6213402. keywords: {Friction;Planning;Optimization;Robot sensing systems;Cameras;Games},

[3] Park J, Park J. Intelligent Carom Billiards Assistive System for automatic solution path generation and actual path prediction with principal component analysisbased ball motion detection. International Journal of Advanced Robotic Systems. 2022;19(4). doi:10.1177/17298806221118865\

[4] Samuel Bauza, Brian Choi, Chuanquiao Huang and Olga Souverneva, “POOL AID: Detection and Identification of Pool Balls for the purpose of recommending shots to beginner players” (2006) Embedded Systems Design Project University of California San Diego