# Chowka Bhara: A Game of Dice

## CS 6601 Mini Project Proposal

### I. Problem Addressed

To implement an optimal strategy for a computer player in this partly stochastic, partly strategic "Game of Dice". The game, designed for 2-4 players, can be played on a square 5x5, 7x7 and 9x9 board, 'X' marking safe houses, with each player having a fixed set of pawns and different starting points (marked in colors). A player wins if he moves all of his pawns to the center square along a predefined path (Fig. 1).

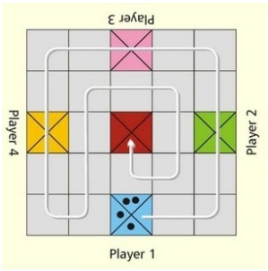


Fig. 1: Predefined path for Player Blue

The game has several intricate rules which we have listed out on a Wikipedia page[1] for the game.

The game poses interesting challenges that need to be addressed through multiple heuristics. There is a considerable amount of work done in analyzing popular partially stochastic games like Backgammon. The objective of our project is to build upon such related work, bring it under the purview of AI scientists and revive this ancient Indian game.

### II. Related Work

There are several variations of the game without an AI component; implemented as Java applets[2] and in nodejs[3]. The only existing computer player implementation[4] uses a weighted tree mechanism as an evaluation strategy to rank each of the moves. However, the factors considered in weighing each move are relatively few and do not vary with time (we address it in the next section).

CHANCEPROBCUT [5] is a general strategy for forward pruning in chance nodes in a game like *Stratego*. We plan to combine this with an improvized alpha-beta pruning approach [6] in a way that will satisfy the rules of our game as well as have good performance. Both the previous techniques have only been tested on popular games and so they might perform poorly if implemented "as-is" in our game.

### III. Implementation

Our computer player will be designed to play on 5x5, 7x7 and 9x9 boards. We plan to tackle this objective in 3 ways: (1) Implementing a variation of the two player A* search [7], where each player looks for the minimum number of steps to reach the goal than to try and block the opponent; (2) A randomized version of the previous approach, allowing some non-optimal or unpredictable moves; (3) Using the CHANCEPROBCUT algorithm and improvized alpha-beta pruning, with a suitable evaluation function. The evaluation function will be a linear combination of the factors below, each of which is assigned a particular weight, varying with time (some factors may outweigh others when the game is close to the end).

1) Will moving of this pawn "hit" an opponent's pawn?
2) Will the pawn move onto a safe house?
3) Will the pawn be "out-of-reach" of an opponent?
4) Will it result in the formation of a double?
5) The closeness of the pawn in reaching the goal.
6) Will it restrict the opponent's pawn(s) to reach the goal?

Our implementation uses three distinct approaches of playing the game, revealing the strengths and weaknesses of each and thereby identifying the most optimal strategy.

### IV. Result Evaluation

1) Compute the winning rate of the three approaches and compare them with each other.
2) Compare the winning rate of approach (3) with that of CHANCEPROBCUT in *Stratego* (51.1%) and examine why it is better (or poorer!)
3) Determine how the time taken for each move is affected by the size of the board.
4) Determine the initial branching factor of the game and the branching factor after applying improved alpha-beta and compare it with similar metrics of Backgammon.

The evaluation metrics mentioned above are very measurable and does not leave any room for ambiguity. From the results of the second metric, it may be possible to evolve better strategies for other similar partially observable games (like Backgammon).

### References


[1] Arvind Krishnaa Jagannathan (2012) *Chowka Bhara* [Online].Available: http://en.wikipedia.org/wiki/Chowka_bhara

[2] KtSoft (2012) Java applet of *Chowka Bhara* [Online]. Available: http://pagdi-boardgame.appspot.com/

[3] S4RK (2012) Nodejs implementation of *Chowka Bhara* [Online]. Available: https://github.com/S4RK/chowka-bhara

[4] Dhanan Sekhar Edathara (2012) Flash implementation of a variation of *Chowka Bhara* [Online]. Available: http://kavidikali.com/making/

[5] Schadd, M.P.D.; Winands, M.H.M.; Uiterwijk, J.W.H.M.; , "CHANCEPROBCUT: Forward pruning in chance nodes," Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on , vol., no., pp.178-185, 7-10 Sept. 2009.

[6] Zhang Congpin; Cui-Jinling; , "Improved Alpha-Beta Pruning of Heuristic Search in Game-Playing Tree," Computer Science and Information Engineering, 2009 WRI World Congress on , vol.2, no., pp.672-674, March 31 2009-April 2 2009

[7] Igor Roizen and Judea Pearl. 1983. "A minimax algorithm better than alpha-beta? Yes and No." Artif. Intell. 21, 1-2 (March 1983), 199-220.