# Bulls and Cows: Strategies and Techniques*

Team Alpha†
Praveen S V (170010025), Deepak H R (170010026),
Anudeep Tubati(170010039), Rohan Shrothrium (170020031)
Computer Science and Engineering, IIT Dharwad

November 16, 2018

**Abstract**

This paper describes the strategies and techniques used by Team Alpha in their program for the *Bulls and Cows* code-breaking game conducted as part of the course CS 211 Data Structures and Algorithms Lab. Although, many strategies were explored, this paper highlights the three main techniques considered by Team Alpha namely the Naive Elimination method, Minimax Algorithm and Randomized-minimax algorithm with $\alpha$ improvisation.

# 1 Naive Approach

## 1.1 Heuristic

A rudimentary strategy to win the game employed simple elimination. First, a list of all possible numbers (4-digit numbers with unique digits) that could be the final guess was generated. For each round the first number was played from the playing set and the score was retrieved. Only the numbers matching the score, when played against our guess, were retained in the playing set. In the worst case, a single number is left in the playing set which is assured to be the opponent's secret number.

## 1.2 Statistics

$\mu_{guesses} = 5.37$
Max number of guesses = 8

*This is a report on the course project for the course CS 211 Data Structures and Algorithms Lab
†Email IDs of team members:[roll-no]@iitdh.ac.in

# 2 Game theory and Minimax

A paper by Donald E. Knuth [1] shows us how to employ the famous Minimax algorithm in Game Theory in the game of Mastermind which is a similar version of the Bulls and Cows game. It was observed that the number of elements remaining in the playing set after elimination was dependent on the previously played guess. Another paper, by G. Ville [2], provided a good description of the implementation of the algorithm.

## 2.1 Heuristic

Similar to the naive approach a list of all possible numbers was generated. Once again, the numbers that produced different scores were eliminated. Instead of playing the first number from the set of possible numbers, the Minimax algorithm gives us the optimal guess number to play.
First an empty score table for each of the 14 possible score combinations is created. A number is chosen from the playing set and played against all the numbers in the list of possible playing numbers. The corresponding score count for each game is then updated into the score table. A number from the set which corresponds to the maximum score count is stored. Similarly, the score table is generated for all the other numbers in the playing set and a number from each maximum score count list is chosen. Finally, the number from the minimum of these maximum score count lists is chosen as the optimal number to be played as the next guess.

## 2.2 Statistics

$\mu_{guesses} = 5.01$
Max number of guesses = 7

# 3 The Alpha Strategy

The performance of Minimax algorithm was improved by randomizing the first guess. Whenever possible, the opponent's first guess was played as Alpha's first guess. This was implemented because there was generally a close proximity between the guess number and the secret number while using C++ random library. After winning two rounds, if it is determined that the opponent's secret guess is constant then Alpha's first guess is the constant secret number.

## 3.1 Statistics

$\mu_{guesses} = 4.95$
Max number of guesses = 7

# 4 Acknowledgements

We would like to express our sincere gratitude to Professor Sandeep R.B. for the project suggestion and his inspiring guidance towards the completion of this project. We would also like to thank our teaching assistant, Ms. Charu Agarwal for the conduction of the league matches and rounds in an unbiased manner and for providing a referee evaulator.

# References

[1] D. E. Knuth. The computer as master mind. *J. Recreational Mathematics*, 9(1), 1976-77.

[2] G. Ville. An optimal mastermind (4, 7) strategy and more results in the expected case. *arXiv preprint arXiv:1305.1010*, 2013.