**TITLETIELTTIELTTIELTTIELTTIELTIELTIELTIELTIELTIELTEIELETIETIETIETLETIETLETIETETKE**

Spencer Chadinha & Alden Hart

**Introduction**

The game of Rock, Paper, Scissors is often frowned upon as a game of pure chance requiring no skill whatsoever. When playing human versus human, this may be a fair assessment. You and your opponent blindly choose Rock, Paper, or Scissors, and hope that you have produced the winning weapon. There is very little opportunity for strategy and most games are abandoned for a more engaging contest.

When pitting two programs against each other, however, the game becomes more interesting. The Nash Equilibrium states that playing randomly is the optimal strategy, and will yield even results of 1/3 wins, 1/3 ties, and 1/3 losses. This, however, assumes that both players play optimally (randomly). If this constraint is relaxed, it is possible to do significantly better than this even split between wins, losses, and ties. It is at this point that the game becomes a problem that can be exploited with a proper application of Artificial Intelligence. If your opponent is not playing randomly, their strategy can be exploited. In order to exploit it, however, you must open yourself up to being exploited. A clever strategy, then, will both attack and defend – that is, you will both try to predict your opponent’s moves and play in a way that is difficult for your opponent to predict.

Our strategy performed significantly better than the Nash Equilibrium when pitted against other programs from the class. Our program finished 2nd overall in the preliminary tournament, scoring a total of 36 out of 44 possible points, where 2 points are awarded for a win, 1 for a tie, and 0 for a loss.

**Background**

Our rock, paper, scissors bot implemented a famous algorithm, Iocaine Powder. Developed by Dan Egnor to compete in the First International RoShamBo Programming Competition, the bot employs a mixture of strategies to both exploit the opponent and remain unexploitable. (CITE THE THING HERE, THE WHOLE REST OF THE SECTION IS BASED ON IT) The algorithm employs six meta-strategies and three strategies, playing the combination that historically has performed the best. The meta-strategies are broken down as follows:

* P.0 plays naively. It predicts with prediction algorithm P what the opponent will play, and plays a move to beat that prediction.
* P.1 assumes the opponent thinks you’re playing P.0. If P predicts your opponent will play rock, your opponent is expecting you to play paper to cover their rock, so your opponent will play scissors so cut your paper. Thus P.1 dictates that you play rock to crush their scissors.
* P.2 assumes the opponent thinks you’re playing P.1. If P predicts your opponent will play rock, P.1 says you should play rock, so your opponent will play paper. Thus you will play scissors. It is at this point that the cyclical nature of the game means that you need not continue this cycle. These three meta-strategies cover all cases of second-guessing.
* P’.0 assumes your opponent is predicting you with P. Thus you predict yourself as if you were your opponent. If P predicts that you will play rock, your opponent will play paper, so you play scissors to beat their paper.
* P’.1 and P’.2 continue the cycle of second guessing. P’.1 assumes your opponent is playing P.1, and P’.2 assumes your opponent is playing P.2.

In order to choose which meta-strategy to use, the algorithm plays all six at every stage of the game and tracks whether each would have won or lost. It does this for each of the three strategies (random, frequency analysis, and history matching), and plays the meta-strategy/strategy combination that would have won the most times in the past.

The meta-strategies predict the highest level of what your opponent is thinking, but they rely on the actual strategies to predict the move they will play. As a fail-safe, the algorithm considers a random strategy. In case your opponent is outsmarting you, the bot will detect that it is losing and start playing randomly in hopes of coming out with no worse than a tie.

Frequency analysis is the first layer of prediction used by our bot. The program tracks the move history of the opponent, and at each stage predicts that the opponent will play its most frequent move next turn.

History matching is the most involved and by far most effective predictor employed by the algorithm. The bot looks for patterns in the opponent’s move history. If one or more matches are found, the strategy predicts that the opponent will play the move that most frequently follows that pattern.

By combining these strategies and meta-strategies, the algorithm predicts several scenarios, playing the one that historically has performed the best. This allows it to adapt to changes in opponent strategy while also staying unpredictable should the opponent start winning.

**Experiment**

**Results**

**Conclusion**