**Moves Performance Optimisation for Large Search Space using Decision Tree Learning Technique on Minimax Algorithm with Alpha-Beta Pruning**

**Khong Wai Foo Andrew1,** **Sun Yick William2, Samuel Winterton3, Kim Tae Luke4**

School of Computer Science & Software Engineering

The University of Western Australia

35 Stirling Highway, Crawley WA 6009

[khongw01@student.uwa.edu.au1](mailto:khongw01@student.uwa.edu.au1), [suny05@student.uwa.edu.au2](mailto:suny05@student.uwa.edu.au2), [wintes02@student.uwa.edu.au3](mailto:wintes02@student.uwa.edu.au3), [kimt01@student.uwa.edu.au4](mailto:kimt01@student.uwa.edu.au4)

**Abstract – The Minimax algorithm with alpha-beta pruning (or simply alpha-beta pruning) is commonly used in machine playing of two-player games such as Tic-tac-toe and chess.[1]**

**In this paper, we show that an agent using the alpha-beta pruning algorithm will not win the game or wins the game with longer time duration than it should take with a large moves space in the MixMeta4 environment. A decision tree learning technique will then be used to decide the move that will lean towards winning the game in the fastest possible way. Finally data gathered from the decision tree will be compared with just the alpha-beta pruning algorithm to determine whether there is an improvement in the agent’s play.**

*Keywords:* Minimax, Alpha-Beta Pruning, MixMeta4, Decision Tree

**1 Introduction**

The Minimax algorithm with alpha-beta pruning or simply just alpha-beta pruning algorithm is more commonly used in machine playing games than the naïve Minimax algorithm as it searches faster. The reason is that it prunes away search paths if the value of the target node is worse than the alpha value.

In the MixMeta4 environment, an agent with just the alpha-beta pruning algorithm seems to win all the games against an agent that chooses random moves in the game such as Arnie. However, when it is played against more intelligent agent such as Hal, it moves backwards or away from the opposition towards the end of the game, thus loses the game.

In this paper, we investigate the effect of allowing the agent to learn the moves via Decision Tree on the probability of winning a chess game against more intelligent agent such as Hal in the MixMeta4 environment.

**H0:** Agent that uses Decision Tree learning technique will make beneficial moves that will lean towards winning the game in a shorter time than agent that does not use the Decision Tree learning technique

**H1:** Agent that uses Decision Tree learning technique will not make beneficial moves that will lean towards winning the game in a shorter time than agent that does not use the Decision Tree learning technique

**2 Method**