Abstract

This project investigated the development of an Artificial Intelligence (AI) player for the multiplayer Chess-like game Duchess. Adjustments and extensions are made to traditional Chess-playing techniques to develop the player, and experiments are conducted on game tree search optimisations to improve its learning potential. These experiments determine that the history heuristic best suits the game of Duchess, and other optimisations such as lookup tables and incremental evaluation are also helpful in increasing the depth of the game search tree.

The TreeStrap machine learning algorithm is used to train the AI player to a competent skill level via self-play. The player is trained in various stages, and each of these stages are competed against each other to evaluate the utility of the feature sets they use. Various techniques are additionally evaluated to reduce the frequency of draw results in an effort to improve future training, including stochastic game-play and new sets of features.

How to play Duchess

Duchess is an adaptation of Chess that is played by two teams of three players. Each player controls 15 pieces; the standard Chess pieces as well as a Fortress (which moves like a Rook and a Bishop), and a Wizard, which enables adjacent friendly pieces to teleport to any squares adjacent to any Wizard on the team. The game is won when all opponents are in Checkmate.

How the AI works

The AI plays the game by conducting an Alpha-Beta search to determine the which move results in the most favourable board state. To determine how favourable a given board state is, it examines multiple sets of Features, which track information like which pieces each player still has (Material), whether certain pieces are attacking or defending one another (Attacking/Defending), and which square pieces are occupying (Piece-Square).

Planning Further Ahead

Helping the AI plan further ahead is crucial to its ability to learn via the TreeStrap learning algorithm, and to help it choose better moves when playing real games. To enable it to traverse the Alpha-Beta search tree faster, we experimented with several optimisation strategies.

Improving Board State Evaluation Accuracy

The other key factor that affects the quality of the AI is how accurately it can estimate its chance of winning on a given board state. If it can reliably determine which board state results in the most favourable board state. To determine how favourable a given board state is, it examines multiple sets of Features, which track information like which pieces each player still has (Material), whether certain pieces are attacking or defending one another (Attacking/Defending), and which square pieces are occupying (Piece-Square).