# Intelligent Strategies for Playing Connection Games with Large Search Space

Casey Beaird
Virginia Commonwealth University
School of Engineering Dept Computer Science
Richmond, Virginia
beairdcm@vcu.edu

Chris Everitt
Virginia Commonwealth University
School of Engineering Dept Computer Science
Richmond, Virginia
everittcj@vcu.edu

**Abstract – Connect four is a member of a collection of games known as connection games. Connection games are played by players who place game pieces in turn with the goal of creating a series of connected game pieces of a desired length. We know from Nash that for any two player finite perfect information game there exists a pure strategy. The accepted method for analyzing dynamic games such as connect for is by creating a game tree and working from the bottom of the tree up to evaluate the value of playing any particular move. In theory this allows for any dynamic perfect information game to be solved. However, in practice even with significant modern computing power many games remain unsolved due to the immense size of their game tree. We present a survey of strategies and their effectiveness as played for the game connect four. We implement and play a generic connection game which allows us the direct ability to manipulate the size of the game tree and therefor the effectiveness of the different strategies with respect to the size of the game.**

## 1. Introduction

Connect Four is a game that was popularized by the Milton Bradly company in 1974. The origin of the game is not truly know but many have claimed to have invented it. Connect four has many names Captains Mistress, Plot Four, Fourplay and Gravitrips to name a few. The game in its 6x7 configuration is strongly solved and is a first player win game. Connect Four was independently solved by two people. The first James Dow Allen who solved the game on October 1, 1988. The second Victor Allis solved the game only 15 days later. When originally solved in 1988 a brute force approach was not considered due to the complexity and size of the game and the limits of computational resources. However, in 1995 John Tromp solved the game using a brute force method by compiling an 8-ply database.

The game in it's traditional form is played on a six by seven board with the winning connection length being four. The number of possible legal game states is 4,531,985,219,092 the number is listed in the on-line encyclopedia of integer sequences a A212693. This should provide some insight into the complexity of solving the game using the traditional tree search. Expanding the game board by even a single row or column expands the search space by orders of magnitude. It is this reason that heuristic playing strategies are required to play intelligently within a reasonable time frame.

The two main strategies we employ to deal with the size of the game tree are first the minimax strategy. We use a parameter to decide how deep into the game tree to explore using this strategy. Second is the Monte Carlo Tree Search (MCTS) which expands the game tree where possible and then plays a random game from that point. MCTS uses a play clock parameter to limit the amount to time the strategy is allowed determine it's next move. We test these strategies against each other and against several others to show how each strategy fairs against the others. We also change the size of the game board to see the impact this has on both the effectiveness of the strategy as well as how it changes the computational time required to determine the next move to play.

The structure of this paper is laid out as follows. Section 1 contains a brief overview of the game of connect four as well as our goals. Section 2 covers in depth the strategies we used to conduct our experiments, the details of how the strategy works as well as the advantages and disadvantages associated with each given strategy. Section 3 covers the experiments we used to asses our hypothesis, and the reasoning behind it. Section 4 covers the results of our experiments, and finally section 5 holds our conclusions.

## 2. Strategies

A player's strategy for the game Connect 4 is responsible for choosing the next move the player will make. Connect 4 has a finite strategy set, more specifically to assign priorities to actions from the set of legal moves that the player can make. The priorities or maybe more aptly probabilities are calculated from the current game state. We introduce in the beginning of this section several unintelligent and often irrational strategies, mainly Random Play and Antagonistic Play. We then introduce perfect play as implemented by a Brute force strategy. Finally, we introduce two more interesting and useful strategies. First the classic game theory strategy, and second the more recently introduced Monte Carlo Tree Search. Each of these strategies benefits and short comings are explained in detail which we use to justify our experimental outcomes.

### 2.1 Random Play

A random strategy, from its name chooses a move from a uniformly random distribution over the set of legal moves. The random strategy does not consider the state of the game board when choosing the next move. The random strategy is a good place to start when evaluating the effectiveness amongst a set of strategies. It serves as a baseline for playing a game, which requires little time to implement nor does the algorithm have heavy computational cost even on large game trees.

### 2.2 Antagonistic Random Play

The second strategy we discuss will be the Antagonistic Random strategy. It is an incremental step forward in intelligence from the purely random strategy. This strategy considers the most basic of game state evaluation algorithms for a connection game. The antagonistic random strategy iterates over all the legal moves from the current board state, playing each move to see if that move results in a win. If the strategy can win by playing a move it will choose that move and terminate the game. However, if no winning move can be played the same process is repeated for the opponent. Each legal move from the current board state is played in turn and at each move the board is evaluated to see if the opponent can win the game by playing that move. Again from the name of the strategy, the current player will play the first move blocking the opponent from winning from that set of connected game pieces. Finally, if no blocking move exists from the set of legal moves then as with the random strategy a move is selected from the uniformly random distribution over the set of legal moves given the board state.

This strategy suffers from being short sighted. The strategy only performs a single move look ahead and will in some instances play a move that provides it's opponent with the move needed to win the game.

We believe this strategy bares a similarity with how we would expect an average human opponent to play. While human players would generally not suffer quite as much from the short sightedness as this strategy. It is easy to imagine that if we extended the look ahead ability of the strategy to even two or three moves then we would quickly approach the typical ability of the average novice player given a large enough game board.

### 2.3 Brute Force

The Brute Force strategy is the first of our strategies that inspect the extensive form game tree for an optimal solution. Brute Force inspects the tree down to the game's end states and assigns the utility of a cumulative utility of the node's descendants, and ultimately to the end game leaf nodes where winning, losing, and drawing at the game's end return high, low, and neutral utility for the player. While conceptually simple this approach is computationally expensive even for relatively small game boards.

### 2.4 Minimax

### 2.5 Monte Carlo Tree Search (MCTS)

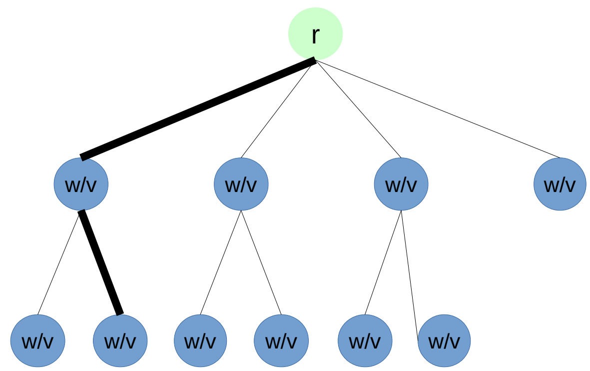
The Monte Carlo Tree Search method is a fairly recent addition the the list of game strategies. The technique was first introduced in 2006 [*]. The strategy is quite simple in it's design the algorithm consists of four simple steps selection, expansion, simulation and back propagation.

The selection step starts from the root of the tree and is recursively applied until an unexplored node is reached. For the most classic definition of MCTS if all nodes of the tree at a given level have been explored then a node is selected at random. However, most implementations of MCTS look to balance exploration and exploitation of promising nodes. There are many techniques to perform node selection on fully explored levels of the tree among the more interesting are simulated annealing and upper confidence bound applied to trees.

For our implementation and experimentation we use the Upper Confidence Bound Applied to Trees (UCT). The technique was developed by [1] and applies the formula below to each node at that level and selects the node with the highest value.

$$\frac{w_i}{n_i} + c\sqrt{\frac{\ln t}{n_i}}$$

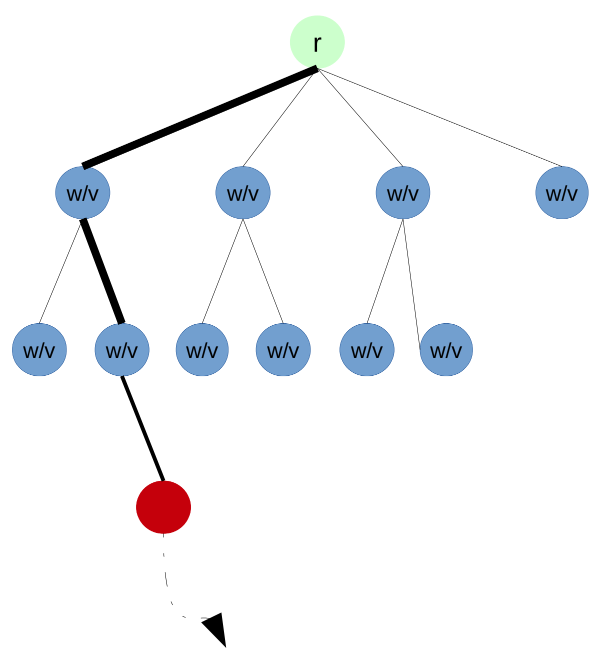
Here $w_i$ is the number of wins for node $i$ and $n_i$ is the number of times node $i$ has been visited. $t$ is the number of times the parent of all $i$ nodes has been visited. $c$ is a tunable constant that can be established either from expert knowledge or through experimentation, $c$ generally falls $\sim\sqrt{2}$ .




Expansion is simply adding the selected unexplored node to the tree and initializing the values for the node.

The simulation step performs the real work of the algorithm. While a simple step, it is the step where a single random game is played from the current node to a terminal node. Unlike a brute force strategy where we play each branch to it's terminal state MCTS selectively plays a single random game from the best unexplored section of the tree.

The back propagation step simply updates the statistics held by each node along the path from the root to the terminal node in the newly explored branch of the tree.




## 3. Experimentation

## 4. Results

## 5. Conclusion

## 6. References

1. Kocsis, Levente, and Csaba Szepesvári. "Bandit based monte-carlo planning." *Machine Learning: ECML 2006*. Springer Berlin Heidelberg, 2006. 282-293.