

# Introduction to AI

## Group Project

TEAM ID: 16

TEAM NAME: BI

TEAM MEMBERS: 0716116 吳灋晉、0616026 游騰德、0616317 游宗穎

## Implementation

### MCTS

我們這組採用的是 MCTS，主要的架構可以區分為 MCTS 與 Node 這兩個主要的 Class，而 MCTS 主要的 4 個架構分別為，Selection、Expansion、Simulation 與 Back propagation。Node 是整個 MCTS 搜尋樹裡面的每個節點，用來紀錄的是盤面，以及他的 Parent 與 Child。

---

#### Selection

我們目前的作法是從 root state，從每一個 child 裡面選擇最佳的 leave node，直到走到 leaf node 為止。

選擇最佳的 leave node 所採用的公式為  $UCB = Q / N + C \times \sqrt{\log(\text{parent}.N) / N}$ ，而 C 為常數，目前設為 1.4。

---

#### Expansion

我們目前的做法是不會將目前的 leave node 所有的 possible move 展開，而是選擇將其中一個的 possible move，然後去進行 simulation。目前我們實作兩種方法，方法 1：一種是只對第一層（root 所能走的 available move）做 Expansion，方法 2：則是比較典型的 Expansion 如果目前所在的 node 還有 available move，我們會先進行 Expansion，而非往其他的 child 去進行 select。

而在方法 1 與方法 2 的比較當中，我們發現方法 1 所能贏的機率較高。有可能是因為 simulation 的次數太少的的原因，而導致做出來的 sample 比較不能代表真正的狀況。

---

#### Simulation

這部分則是一直往下隨機的決定下一步，直到停止在 end state 的盤面，達到似於採樣的結果。而在 end state 的盤面的時候我們會進行 Back propagation。

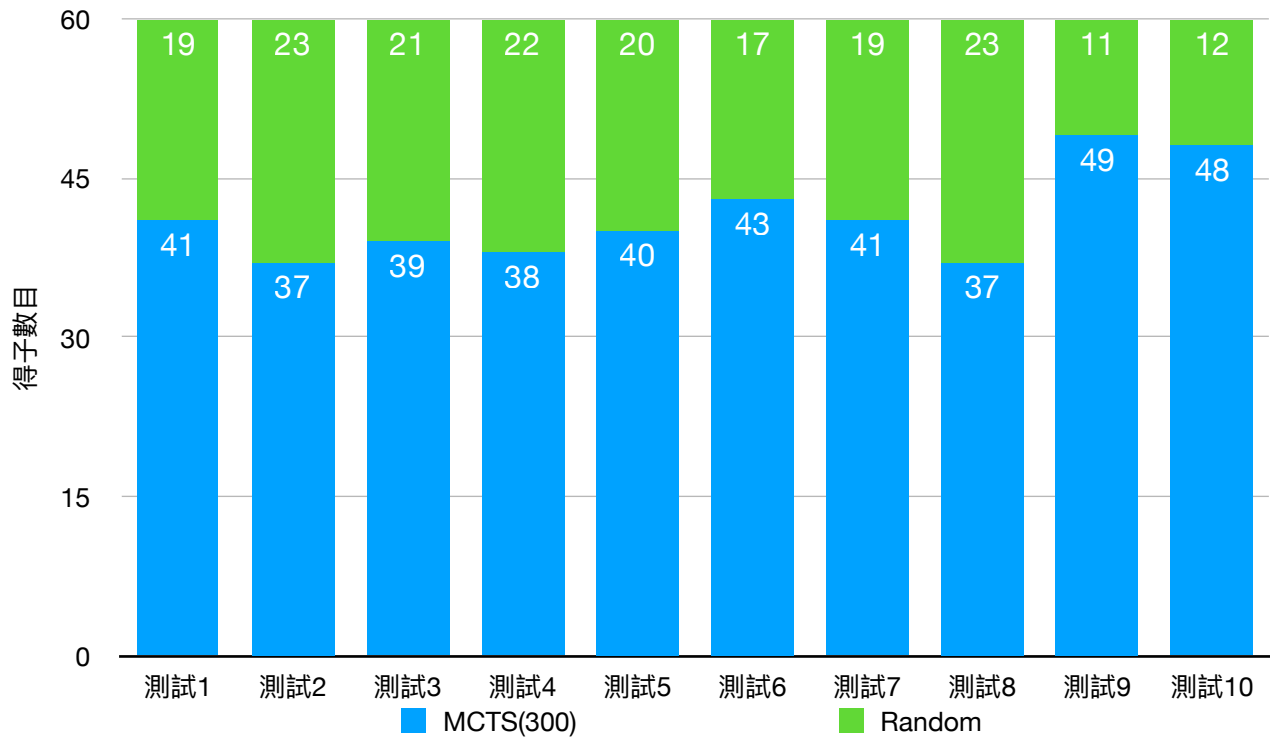
---

#### Back Propagation

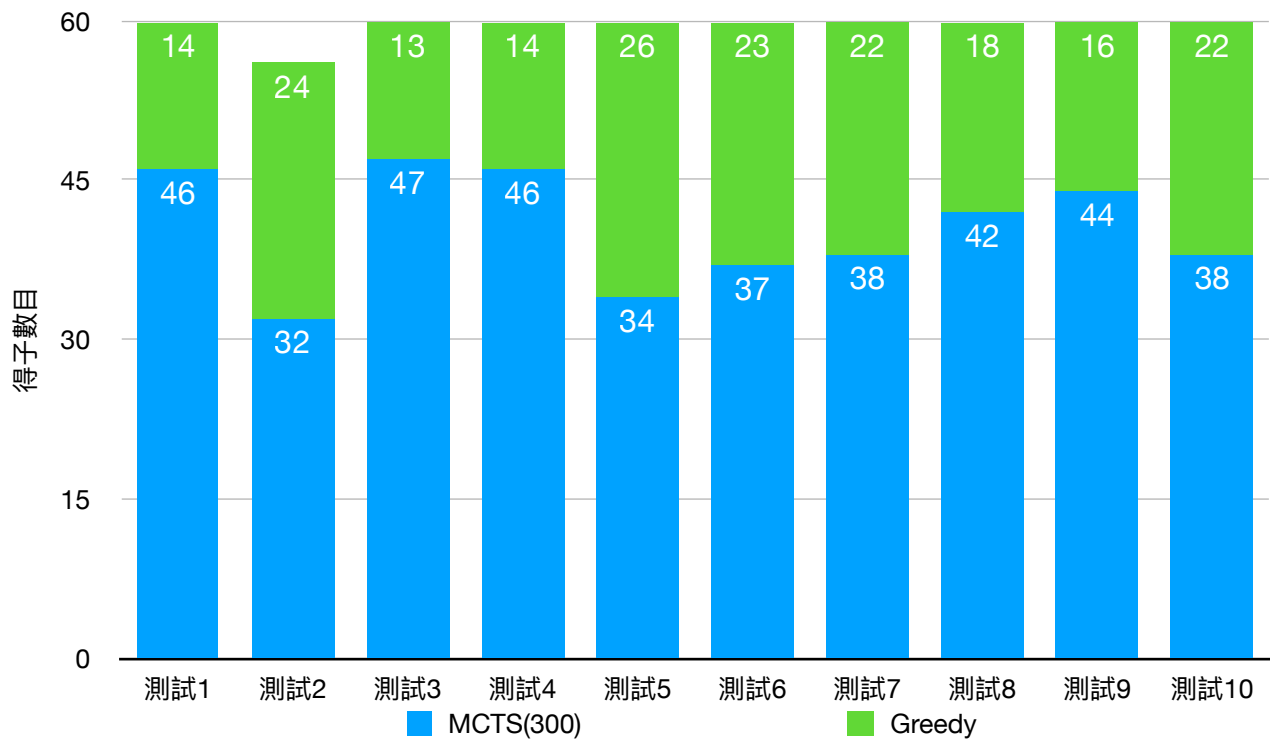
更新 N 與 Q 的值，贏的話就將 Q 得值 +1，否則 -1，一直從跟節點往上遞迴，直到 root 為止。

## Experiment

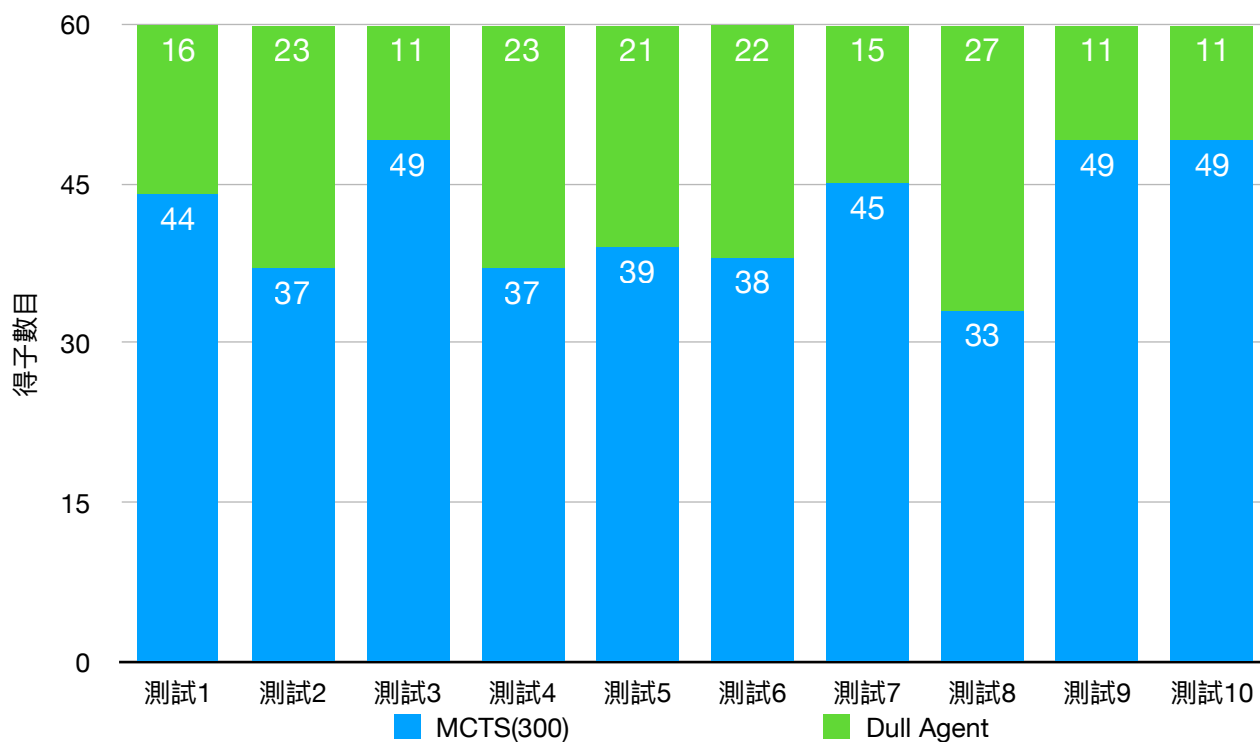
### MCTS vs Random



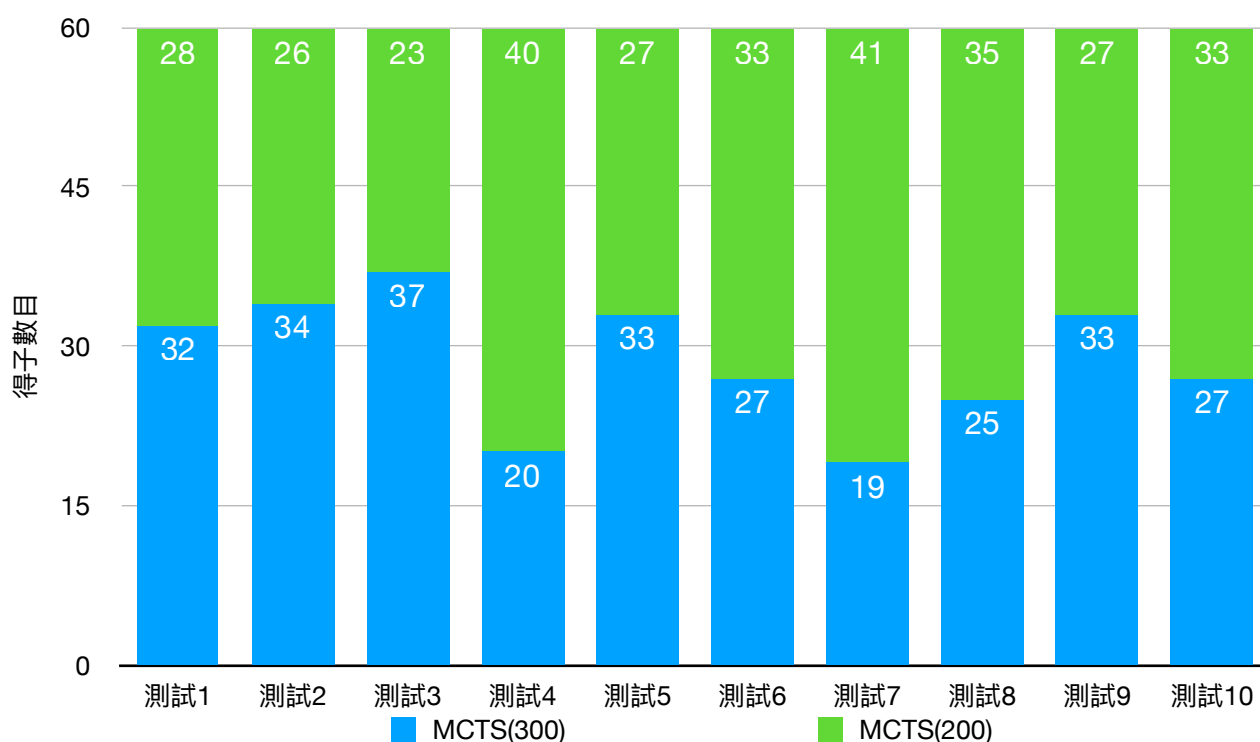
### MCTS vs Greedy



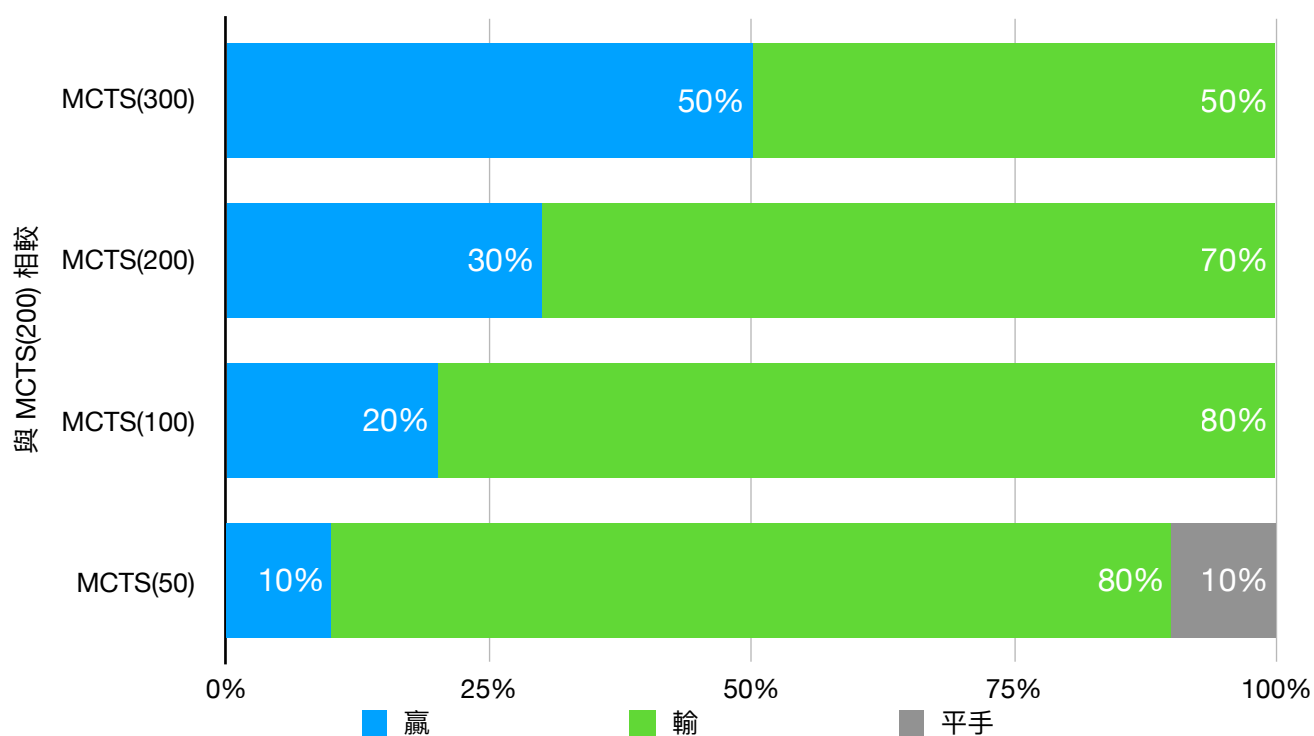
## MCTS vs Dull



## Stronger MCTS vs Weaker MCTS

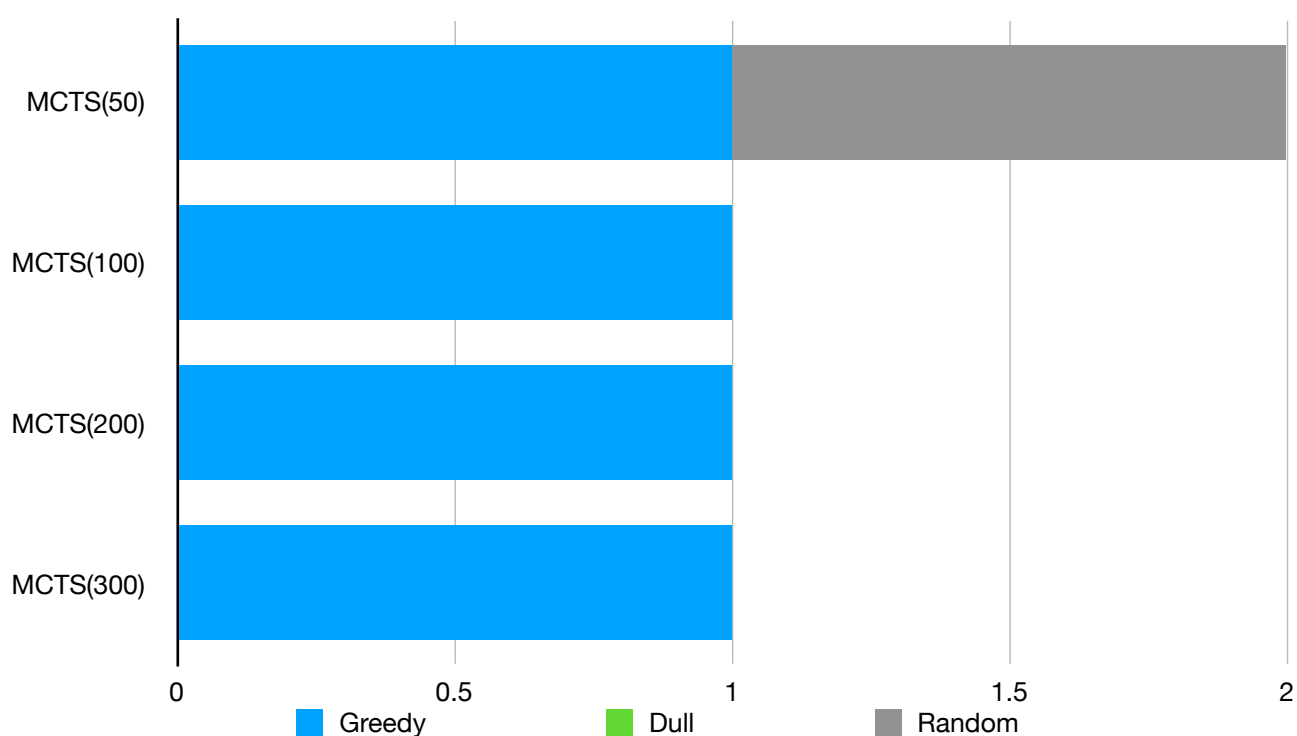


從上面的數據表現，可以得知基本上 MCTS 取樣 300 次的表現優於隨機下，貪婪（盡可能奪最多的點），Dull Agent（與看到第一個能下點就下），但是在 MCTS(300) 與 MCTS(200) 的項目上，事實上勝負各半，也因此接下來繼續探討，MCTS 取樣上的差別。



根據上表可以得知 MCTS(200) 與 MCTS(300) 的方法在勝率表現是差不多的，但在 MCTS(100) 以下就出現了明顯的差距，但因為是抽樣，不代表 MCTS(50) 不會有贏的機會，即便是 MCTS(50) 也可能出現抽樣表現比 MCTS(300) 的狀態好的狀態。

而對付其他的 Agents，MCTS 的表現會不會也因此有表現上的差距呢？



由此可知，在抽樣的條件狀況次數（> 50）的狀況底下，基本上這些 MCTS 的勝率都表現得相當好，但是如果對手變強 MCTS 抽樣次數則是越高的表現會越好。