

Final Project

R09922188 曾泓硯

January 5, 2023

1 How to compile your code into an agent

1. `> $ make`

The agent will be compiled into file "negascout".

2 What algorithms and heuristics you've implemented?

在這一次的專案中，我實作了 NegaScout, Star1 algorithm, transposition table, IDAS, time control module 以及 heuristic 的審局函數。

3 Experiment results and findings of your implementation

在實作的過程中，我發現自己的 IDAS 搜尋五層的結果，與直接搜尋五層的結果不同，在這樣的前提之下我比較了 IDAS 搜尋不同層數與直接搜五層的差別。

Agent	IDAS depth 6	IDAS depth 5	IDAS depth 4	direct depth 5
IDAS depth 6	X	10W10L	12W8L	8W12L
IDAS depth 5	10W10L	X	11W9L	7W13L
IDAS depth 4	8W12L	9W11L	X	6W14L
direct depth 5	12W8L	13W7L	14W6L	X

Table 1: Win rate under different agent

從表格中，我們可以發現直接搜五層的結果會比 IDAS 搜尋不同層數還要好，但是還是可以看出相差兩層時，搜尋越深的會比較有利。而 IDAS 的結果會出現這個的原因可能跟 transposition table 有關，因此我把骰子的資訊也一起加到 transposition table，來增加正確性。

4 Some detail about your implementation

1. NegaScout - 這一次的作業中規定要實作 NegaScout，我在實作的過程中原本是使用 Nega 版本的，並且也成功的作出了 Star0 版本，但是到了 Star0.5 版本時發現剪枝後的結果與 Star0 不同，才發現說 chance node 也要寫成 Nega 版本的，也因為改不太出來，所以最後就寫成 Min-max 版本。另外，NegaScout 演算法中有一行當 $depth < 3$ 時就不用 research，這件事情因為在 chance node 計算時可能因為 pruning 導致結果不精準，所以不符合原本可以 pruning 的假設，因此我把它改成 $depth < 1$ 時才能夠剪枝。

2. Chance Node - 這一次我實作了 Star1 的演算法，與 Star0 相比速度有感提升，原本 star0 搜尋五層可能需要花費 5 秒左右，但現在可以穩定限制在 2 秒內。基本上大致程式碼與講義上相同，但在實作的過程中，因為計算過程中會有誤差，會造成小數點後 7 位數字可能會不同，因此我將算出的數字先乘上 10^6 取 floor 以後再除回去，如此一來就可以得到完全相同的結果。

3. Transposition table - 我實作的方法基本上與講義上相同，只是在 Min node 的時候 Upper bound 的 flag 要跟 Lower bound 交換，並且也利用 XOR 的特性在 $O(1)$ 的時間內計算出下一步的 hash value。此外，參考講義上的公式，我把 table 的數量設定在 2^{17} ，hash 的數值範圍則是 2^{25} ，以此來降低 hash miss。另外，我在經歷第一輪比賽後，就把骰子的資訊加入到 hash 的計算中，也讓 transposition table 在整場比賽中共用，不會每一步就刷新一次。

4. Heuristic evaluation function - 在經過測試以後，我發現用整個盤面的數據來進行統計其實是沒有太大的意義的，因此我利用雙方離勝利最近的棋子來進行統計，結果發現對各棋子而言，最近的距離以及它可以移動的機率會對結果影響最大。

5. Time control - 由於目前的演算法在搜尋五層的時間花費不超過 1 秒鐘，因此我的目標是讓演算法搜尋得更深，或者退而求其次，在關鍵步的時候搜尋久一點。搜尋 6 層的時間花費大概介於 15 40 秒之間，因此，我利用 這一步的時間 \times 這一步的時間 / 上一步的時間來估計下一步所需的時間，如果有機會做完的話，就會再搜尋一層。另外，我也設定了一個時間的硬上限（剩餘時間的 $/3$ ），如果超出的話會直接回傳到根結點。

6. IDAS - 基本上按照講義上去實作，threshold 的設定上就是憑感覺而已。

5 Discuss benefits of various enhancements

Week1 - 在第一週的時候，我把 NegaScout, Star1, transposition table, heuristic evaluation function 都實作出來，發現從 Star0 - Star1 的過程中搜尋的時間會有 2 倍左右的進步，而 transposition table 也還會再有 1.5 倍左右的加速。至於 Evaluation function 我一開始是只使用距離當作參考，也讓我可以穩定贏過 random baseline。

Week2 - 經過第一週的比賽後，我將 transposition table 引入了骰子點數的資訊，並且每一步結束後不清空，再來加上了時間控制的參數，讓 IDAS 可以在指定的時間內盡可能走搜尋越深越好。此外，我也補上當只有一種走步方式時，就直接選擇沒有必要再搜尋的方法。加上了這些方法後，與第一週的程式對打大概 12 勝 8 敗。

6 Reference

- 課程講義