

第五章 結論與未來研究方向



第一節 結論

本研究首度運用 UCT 演算法在黑白棋上，在目前黑白棋相關文獻中，尚未有人提出相關的研究。而我們主要目標是在探討 UCT 演算法在黑白棋上的適用性。雖然 UCT 演算法在 9 路圍棋的比賽中擊敗了實力強勁的業餘棋手，取得相當不錯的成果，但根據我們的實驗結果發現 UCT 演算法並不太適合使用於黑白棋中，這可能是跟黑白棋的特性有關。在圍棋中，我方走錯一、二子，在局勢上我方仍可保持優勢，但在黑白棋中走錯一子，因為黑白棋翻轉的特性，極易造成局勢瞬間逆轉。而在雙方勢均力敵的對戰盤面，如 Wzebra 自我對戰盤面，其獲勝的 PVS 路徑很少，因此贏率會被非 PVS (Principal Variation Search) 的路徑贏率給稀釋掉，造成贏率不高的狀況出現。由我們的研究顯示，UCT 演算法較不適用在極易翻盤的棋類，較適合用在走錯子後仍可繼續保持優勢的棋類，如圍棋、Amazon 等。

在研究的過程中，我們也針對黑白棋特性，提出了多項改進 UCT 演算法的策略，這些改進策略實作完成之後，在執行時的確能改進效率，但和 Wzebra 對戰，仍無得勝之機會。實際上我們拿實作的 UCT 演算法黑白棋程式與眾多的黑白棋程式進行對戰（如附錄），比較後發現 UCT 演算法仍具有一定水準的棋力，但仍無法與發展多年的程式匹敵。然而，若是在欠缺黑白棋相關 domain knowledge

的前提下，想要在短時間內撰寫出一個具有不錯棋力的黑白棋程式，其實是可以考慮採用 UCT 演算法來達成。希望這些經驗能作為未來 UCT 演算法相關研究的參考。

第二節 未來研究方向

本研究探討了 UCT 演算法在黑白棋上的適用性問題，由於黑白棋的特性，使得造成原本可獲勝的路徑在大量模擬時不一定贏率較高，使得 UCT 演算法無法找出最佳走步。這使我們了解 UCT 演算法較適用在盤面獲勝路徑較多的棋類，如圍棋、Amazon 等。而針對 UCT 演算法的精神，未來可進行的研究方向如下：

1. 改善 UCT 演算法無法判別大輸小輸、大贏小贏的狀況：

例如根據盤面勝負棋子數量，來決定獲得分數大小差距，使 UCT 演算法較能區別出大輸小輸、大贏小贏的狀況。而目前 UCT 演算法獲勝得分一律為 1 分、和為 0.5、輸為 0 分。

2. 找出 UCT 演算法計算索引值中修補分數內常數 C 的最佳值：

該常數會影響到 UCT 建立決策樹的行為模式。該常數值若很大，會使 UCT 演算法很頻繁地選擇拜訪次數少的分枝；該常數值若很小，則會使得 UCT 演算法過於集中拜訪贏率高的分枝，忽略拜訪次數少的分枝。因此必須在這二種行為模式中取得一個平衡點，找出一個最佳值，能使得 UCT 演算法能有最好的行為表現。而目前我們則是採用 MoGo 使用的常數值 $\sqrt{2}$ ，該常數是目前 UCT 演算法使用在圍棋中的最佳常數。

3. 結合 UCT 演算法與 α - β pruning 優點：

我們知道 UCT 演算法的整體精神，是根據贏率以類似廣度優先搜尋的方式來建立決策樹，而 α - β pruning 則是以深度優先搜尋的方式來搜尋整棵樹，若能結合二者的優點，或許可以改善目前 UCT 演算法在黑白棋中所遇到的瓶頸。