

# 第一章 緒論

## 第一節 前言

黑白棋是由英國人在 19 世紀末發明的一個棋盤遊戲，當時的名稱叫做 Reversi，由於兩位英國人 John W. Mollet 及 Lewis Waterman，都爭著做這個遊戲的發明人，這個爭執卻使這個棋盤遊戲流行起來。到了 1971 年，日本人 Goro Hasegawa 以 Reversi 為藍本再加上一些新的棋例，並借用莎士比亞四大悲劇之一 "Othello" 為這個遊戲重新命名及註冊，就變成今日的黑白棋了。Othello 是莎士比亞同名小說的男主角。他是一個黑人，妻子是個白人，因受小人挑撥離間，懷疑妻子不忠以致情海翻波，最後把妻子殺死。但其後真相大白，Othello 懊悔不已，自殺而死。黑白棋就是借用這個黑人白人鬥爭的故事來命名，頗有文化氣息。

黑白棋是由黑方和白方兩人進行的益智遊戲，一般棋盤為 8x8 綠色方格，總共使用 64 個棋子，每個棋子分正反兩面，分別是黑色和白色。遊戲開始前，在棋盤正中有 2 白 2 黑總共 4 個棋子交叉放置，如圖 1-1 (a)，黑方總是先下。而開始下棋時，雙方各使用 32 個子，棋子下在空格中間，而不是像圍棋一樣下在交叉點上。輪到另一方下棋時，必須把棋子下在與對方棋子相鄰的空位上，要求所下的棋子和原有我方棋子夾住敵方至少一個棋子(任何方向均可)，然後把被夾住的對方棋子變成我方的顏色，如圖 1-1 (b) 與 (c)，由於黑方在 F5 下子，造成原先 E5 位置的白子被翻轉成黑子。下棋過程中，任何棋子既不會從棋盤上拿

走，也不會從一個格子移到另一個格子。

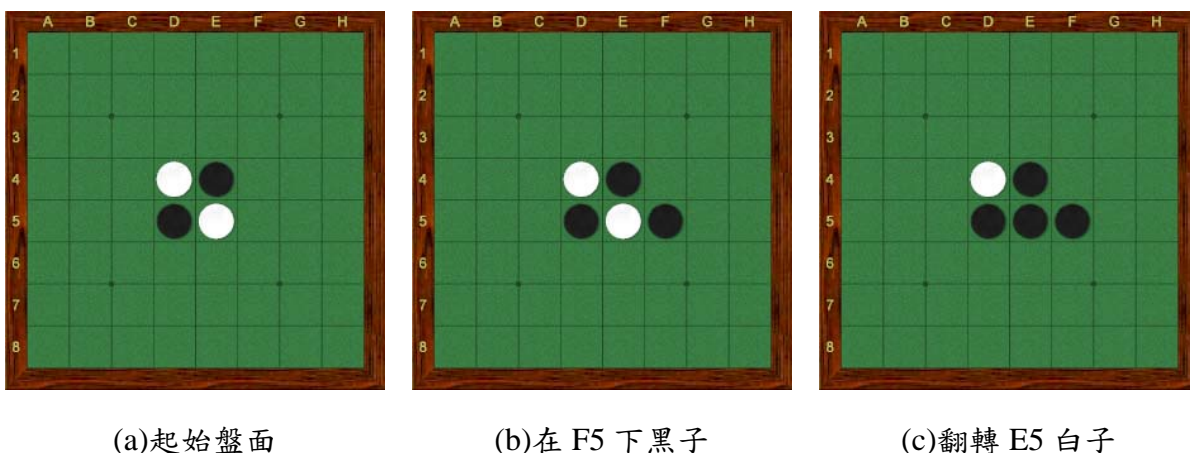


圖 1-1 棋局吃子過程

而在輪到某方下棋時，若無子可翻轉，則必須停一步，讓另一方繼續下棋，這種情況稱為 pass。如圖 1-2 (a) 的盤面，此時雖然輪黑下子但是卻無法翻轉白子，因此只好 pass 由對手白方下子，而白方在 A1 下子，使得原先在 B1、C1 及 D1 的黑子翻轉為白子，如圖 1-2(b)；而當任何一方有子可翻轉時，則不能選擇 pass。

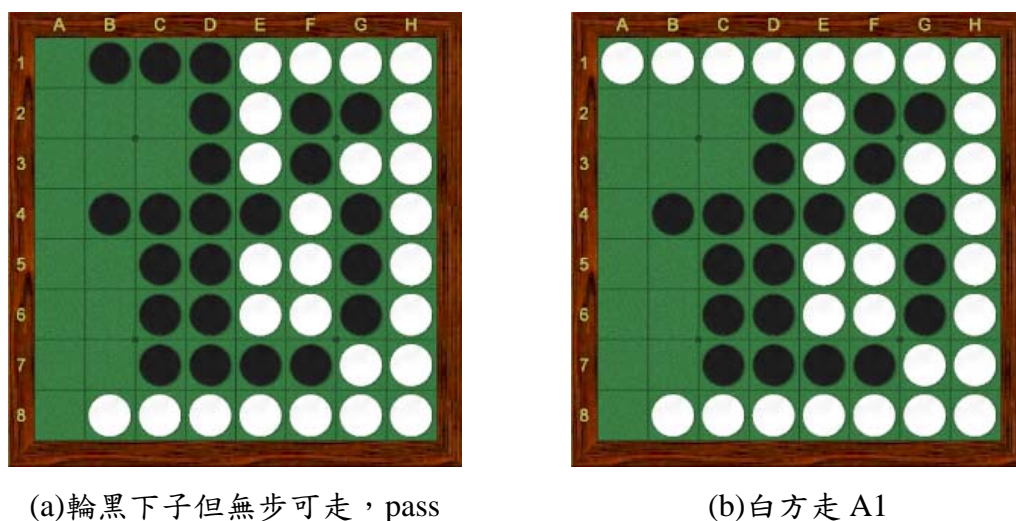
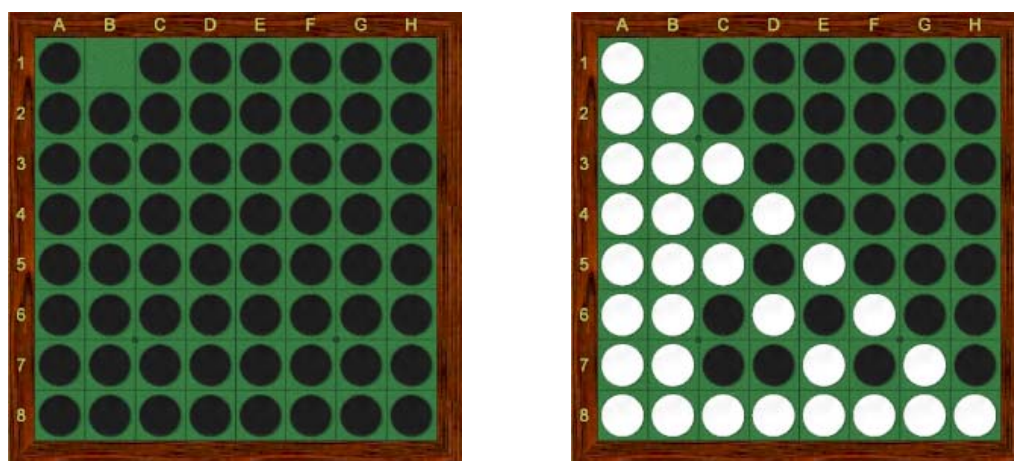


圖 1-2 棋局 pass 過程

當雙方都無棋可下時，棋局便結束了，一般情況是所有 64 個方格都被佔滿，但是也有這樣的情況，棋盤上雖然還有空位，但雙方都不能下。其中一種情況是

棋盤上全部是黑子或白子，如圖 1-3 (a)；另一種情況如圖 1-3 (b)，因為雙方在 B1 下子都不能翻轉對方的棋子，所以都不能在 B1 下棋，因此棋局結束，而棋盤上有 34 個黑子和 29 個白子，空位判給目數較多的一方，最終結果是黑方 35 子贏白方 29 子。



(a)盤面上全為黑子

(b)雙方無步可走

圖 1-3 棋局結束狀況

目前最強的黑白棋電腦程式可以輕鬆的擊敗最強的人類棋手。早在 1997 年時，由 Michael Buro 所發展的 Logistello 更以 6：0 懸殊的比數打敗當時人類的冠軍 Takeshi Murakami[6]。當時，像這樣的優勢在其它的棋類是空前未有的，以西洋棋為例，最好的西洋棋程式棋力大約只等於當時的人類冠軍；而在圍棋，相反地，大多數的人則可以輕鬆地擊敗目前最強的圍棋程式。

一般來講，人類與黑白棋電腦程式對奕是不可能獲勝，這是因為電腦能比人類往前看更多的走法變化，從一開始下棋起，便先設想到許多步棋以後的棋局狀況，而在第一步棋就規劃無懈可擊的策略。但是由於可能的組合實在太多了，黑

白棋合法位置的數目組合大約是  $10^{28}$ ，遊戲樹的複雜度大約是  $10^{58}$ ，目前要完全由電腦自動展開整個遊戲樹，仍是力有未逮。

我們知道在  $n$ -by- $n$  盤面上推論是否先手的玩家必可贏得比賽是 PSPACE complete 問題[12]。而截至目前為止，4-by-4 及 6-by-6 盤面均已被證明為後手必勝，其中 6-by-6 盤面的完全解由 Joel Feinstein 在 1993 年獲得[13]，而 8-by-8 盤面仍然未求得完全解，而據分析數千個高品質的盤面（大部分由電腦對奕產生）所得的結論，8-by-8 盤面最後的結果很可能為和[14]。

## 第二節 研究動機

1997 年由 IBM 公司研製的超級計算機「深藍」(Deep Blue)在 6 局比賽中擊敗了當時的西洋棋世界冠軍卡斯帕洛夫 (Kasparov) 轟動全球，讓人們見識到電腦運算能力的可怕，電腦再一次終結了人類在另一個棋類遊戲上的統治地位。而目前仍有圍棋尚未被電腦超越，人類總是能夠輕鬆擊敗電腦。但最近電腦圍棋界出現了一種新的演算法，使得電腦圍棋程式能戰勝高水平的對手。

我們知道圍棋的棋盤由兩組數量相同、互相正交的平行線構成，分為 9 路及 19 路圍棋。對弈雙方分執黑白兩色棋子，通過在棋盤的交叉點上落子，棋手要儘可能擴張自己的領地並包圍對方棋子，而每一步可採取的策略數量都非常多。在對局中期時，平均每一步可採取的策略可多達 200 種，相較之下，西洋棋中每一步數十種的可選策略就顯得微不足道了。由於圍棋的複雜度極高，因此目前電腦圍棋程式所設計出各種演算方法，仍無法在 19 路圍棋的比賽中戰勝實力稍強的

人類棋手。

在 2006 年秋季，由兩位匈牙利研究人員發明了一種新的演算法，它的勝率比現有的最佳演算方法提高了 5%，使之能夠在 9 路圍棋的比賽中與人類職業棋手抗衡。這種被稱為 UCT(Upper Confidence bounds applied to Trees)的算法，是匈牙利國家科學院計算機與自動化研究所 Levente Kocsis 與加拿大阿爾伯塔大學 Csaba Szepesvari 合作提出的[3]，是著名的蒙地卡羅方法(Monte Carlo method)的擴展應用。70 年代，蒙地卡羅方法首次被運用於圍棋程式，這種方法以隨機走步的方式來模擬大量的比賽，採用統計的方式來預測大規模群體的表现與特性，藉以對候選走步加以評估並排序。然而即使每一步都採用評估值最高的走步，仍不能保證獲得比賽的勝利。

2007 年，法國南巴黎大學的數學家 Sylvain Gelly 與巴黎技術學校的 Yizao Wang，將 UCT 實作電腦圍棋程式 MoGo 中，獲得的勝率比先前最先進的蒙特卡羅擴展算法幾乎高出了一倍[15]。2007 年春季，MoGo 在 9 路圍棋的比賽中擊敗了實力強勁的業餘棋手，在 19 路圍棋的比賽中也擊敗了實力稍弱的業餘棋手，充分展示 UCT 演算方法的能力。

有鑑於 UCT 演算法在圍棋獲得的成果，目前有部分專家學者將 UCT 演算法應用於其他棋類遊戲中，如 Amazon[5]，但用於黑白棋的研究仍付諸闕如。因此本論文將運用 UCT 演算法於黑白棋上，除探討 UCT 演算法在黑白棋上的的適用性及優缺點外，並進一步提出改進策略。

### 第三節 論文架構

本論文共分為五章。第一章為緒論，簡述前言、研究動機及論文架構。第二章回顧電腦黑白棋發展之歷程，並敘述其使用的基礎理論，如對局樹演算法與電腦黑白棋設計技巧等。第三章為基本架構，概述實作黑白棋程式時，使用的資料結構、演算法與設計理念等。第四章為 UCT 演算法之實驗結果分析，除探討 UCT 演算法在黑白棋上的適用性外，並提出演算法之改進策略。第五章為結論及未來研究方向。