

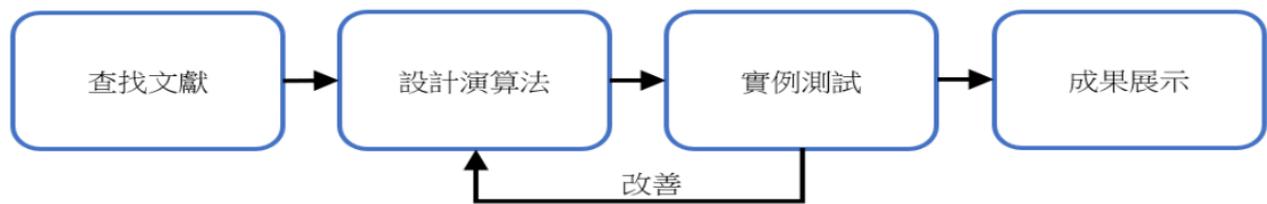
2024 年【科學探究競賽-這樣教我就懂】

普高組 成果報告表單

題目名稱： 起手無回大丈夫—以程式解決象棋殘局
一、摘要
中國象棋傳承已久，經過了無數人的潛心鑽研後產生了許多不同的殘局。而解決這些殘局也變成了許多人想要挑戰解決的目標，而我們也不例外。為了解決這些五花八門的殘局，我們用 C++ 語法設計了一個可以計算殘局解法的演算法，來達成我們的目標。而為了驗證程式的可行性，我們使用許多的殘局來驗證，並深入分析其中所遇到的問題以及未來能夠改進的方向及目標。
二、探究題目與動機
下棋有助於增進頭腦思考、策略邏輯和修身養性，對象棋規則有一定了解的我們觀察到每次下棋幾乎從來不出現同樣盤面。我們深深著迷於此遊戲的複雜，也不禁好奇這種程式背後的邏輯究竟是如何產生的，在那些「神之一手」的背後又是多少複雜運算所堆疊出來的。與其他棋類遊戲一樣，中國象棋涉及到複雜的策略和計劃。和其他類似的棋類遊戲如西洋棋和日本將棋相比，象棋的複雜性不是普通的高，即使如此，對象棋的著迷促使我們設計一個能指出最佳策略的電腦程式，在增進自己知識的同時並幫助我們提升棋藝。由此，我們決定以研究尚少、又有一定熟悉度的中國象棋作為研究主題。
三、探究目的與假設
一、提高人類玩家的策略：透過電腦程式進行深入的棋局分析，可以為人類玩家提供策略建議和不同想法，有助於找到不同解法。 二、分析步驟與產生結果：通過給出每一步之後產生的棋局變化，可以幫助判斷一個步驟是好或者壞，並提供指南或建議。 三、解決不同的殘局問題：經過演算法運算，解決人類難以一眼看出的殘局解法並獲勝。 四、累積此方面程式設計的知識：在網路可查到許多與電腦對弈的網頁，我們著迷於其能力的同時也不忘思考該如此厲害的程式背後的運作原理，親自設計一個程式剛好可讓我們有機會對於其運作邏輯有更透徹的理解。
四、探究方法與驗證步驟
一、研究方法 (一) 文獻研究方法：在進程式設計前，文獻研究是必不可少的一部，了解相關的技術和方法才能順利的編寫程式。透過各類的程式設計相關書籍，協助我們建立基礎知識，從而更好地設計和開發程式。 (二) 實驗研究方法：首先，我們將收集包含不同難度和形式的象棋殘局數據集作為研究對象。接著，透過 VS Code 將收集到的殘局輸入設計好的程式，並進行實驗，以找到程式設

計之不足。透過一次次的實驗試錯比對，成功設計出最後的程式。

二、研究流程圖



圖一：研究流程圖

三、SWOT 分析圖



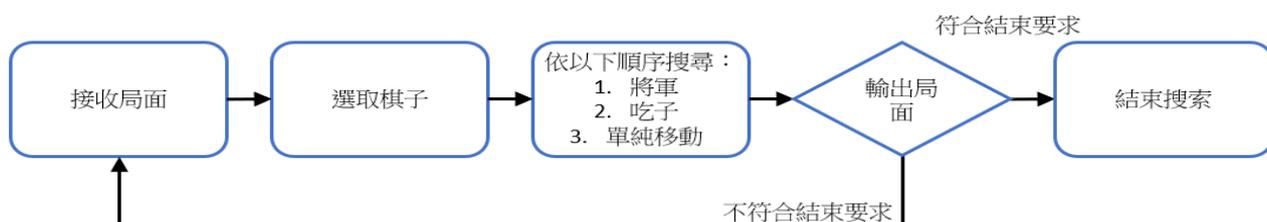
圖二：SWOT 分析圖

四、演算法邏輯設計

本程式的核心依然是搜索演算法背後的運作邏輯，雖然我們尚無能力做出高難度的人工智慧分析，但我們參考 MiniMax 演算法，設計出了一個能夠對簡單殘局棋面進行搜索的演算法，以下是其邏輯設計（以紅方先走為例）：

- （一）對於我們接收進來的局面，搜尋盤面上所有紅棋的所有移動方法，並將之作為子局面。
- （二）對於任何一個子局面，再以所有黑棋的所有移動方法作為它的子局面，重複直到建立完我們想搜索的最大層數。
- （三）對建立起來的樹進行一個一個節點進行訪問，找出必勝的下法。

即使是殘局，中國象棋的複雜度也十分高，若單純用以上的演算法搜索的話勢必在空間和時間上都占用太多，另外，我們也無法對一個移動方式給出非常好的分數分配，經過思考後我們將上述演算法進行下列改良，如下圖所示：



圖三：程式運行流程圖

(四) 將棋子的移動分為將軍、吃子和單純移動，以此做為搜索的順位，除非沒有上一順位的移動才搜索下一順位。

(五) 當輪到紅方移動且將黑方將軍時結束搜索，並將此步加入必勝的節點。

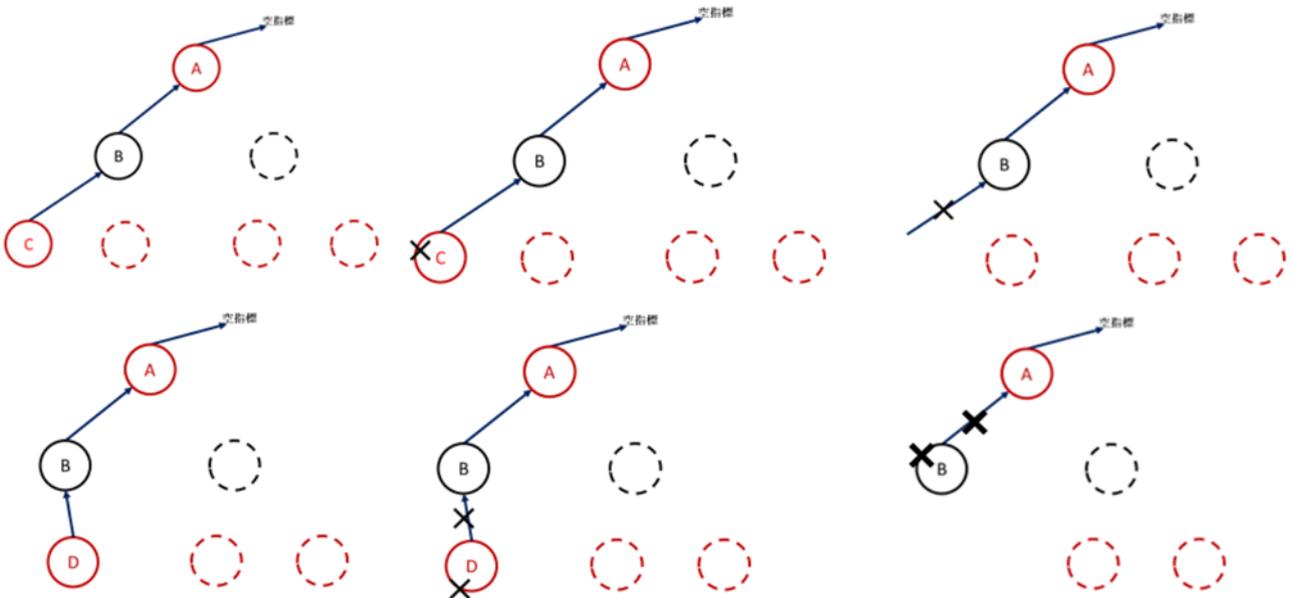
(六) 當輪到黑方移動且將紅方將軍時直接結束搜索。

(七) 將節點之間的連接的指標以智慧指標取代，所以當一個節點不是必勝的且不再搜索時，就會被自動摧毀且釋放出記憶體。

當一次完整的搜索進行完後，那些「必勝法」會用智慧指標指向，使得這些節點跟他們的祖先節點都不會被刪除，至於其他節點都會被刪除並釋放出它們的記憶體空間。

五、程式原理

從零開始實作一個程式並不容易，我們也歷經了多次修改，雖然可能不是最理想，但也最終撰寫出了一個可以成功執行的版本，結合了 C++ 物件導向的特性以及我們設計的演算法，以下是我們程式的幾個重要特性。我們將所有棋子實作成一個 class，並且它們皆繼承自同一個父類別以做到迴圈的統一處理，而一個棋盤則含有 9×10 個棋子 (包含沒有棋子的空格)，對局樹的節點存放一個棋盤存放當前的局面和一個智慧指標用以指向父節點。我們的演算法能夠一邊建立對局樹一邊進行搜索，以此來彌補中國象棋複雜度使得記憶體消耗過多的缺點，如同上述說明的，只要找到必勝的節點就將其記錄，只要搜索完一個節點所有的子節點，其就會被自動摧毀並釋放出記憶體，以下為示意圖：

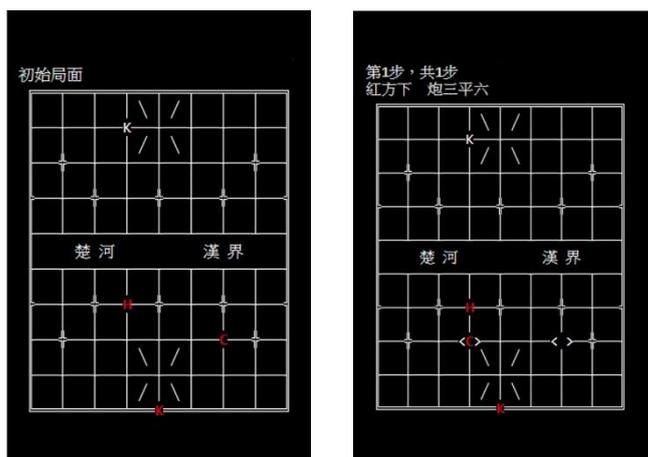


圖四~九：智慧指標示意圖

參、測試執行

在上網查找正式的棋譜之前，我們透過一些簡單的、人類目測便可看出必勝下法的棋盤進行驗證，下面就是我們使用的最簡單的例子。在接下來的圖片中使用到的代號如下：K 代表將、帥(King)；A 代表士、仕(Assistant)；E 代表象、相(Elephant)；R 代表俥、車；H 代

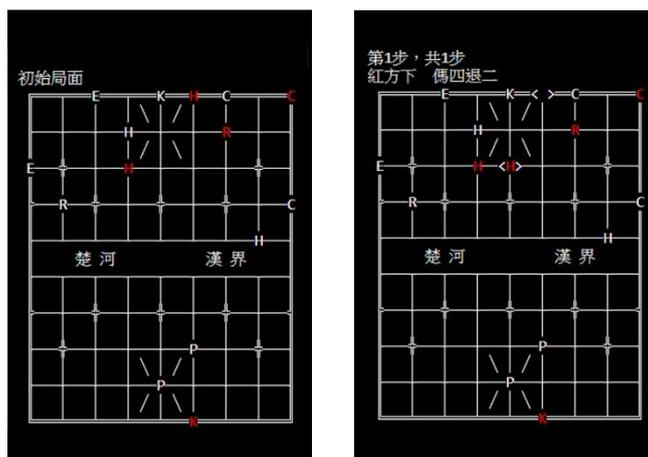
表馬、馮(Horse)；C 代表砲、炮(Canon)；P 代表兵、卒(Pawn)，白色字為黑棋；紅色字為紅棋，棋盤下方為紅方，如下圖十、十一。



圖十、十一：測試用例一

了解象棋規則的話很容易可以看出紅方如何移動可取得勝利，而我們的程式也確實給出了此下法。

單純倚靠一個測試就判定程式沒問題略顯草率，所以以下是我們的第二個測試用的例子。

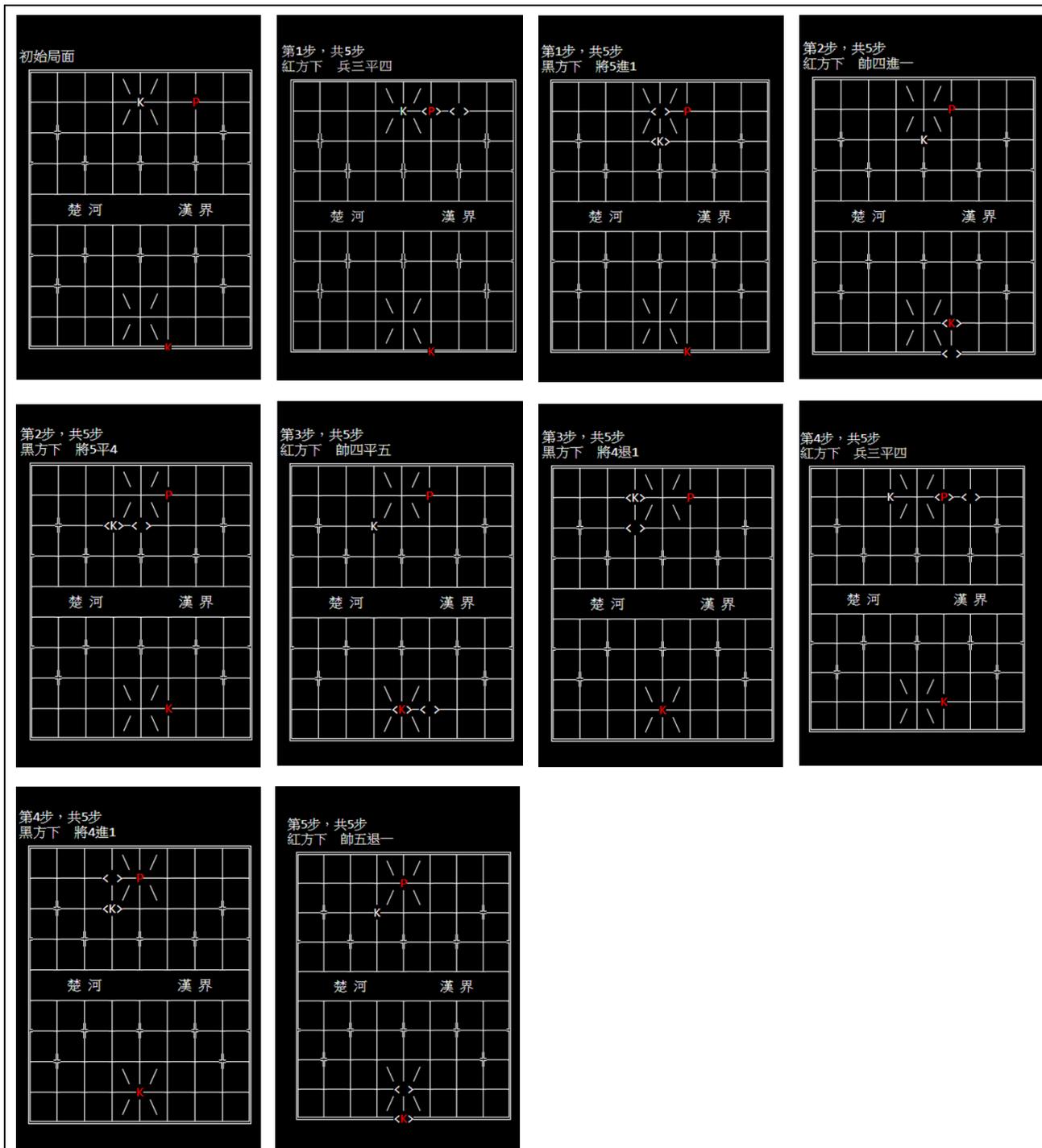


圖十二、十三：測試用例二

六、真實殘局測試

經過第二節的兩個測試例子，我們確定程式是可以正常執行的，但僅使用兩個例子就宣稱程式是成功的確實過早，所以我們也在網路上搜尋真實的殘局，並測試程式的能耐以及是否有不足之處。接下來，我們在網路（取自永樂象棋棋譜網 <https://reurl.cc/RWep9D>）上搜索真實的殘局，試圖確認程式是否真的可行，以及其可能存在的漏洞。我們選取的殘局叫做單兵例勝孤將之一，以下是程式給出的六個走法中的一個，與其網站上的走法相同。

從以下圖十二~二十一這些殘局的執行結果中，我們可以看出其確實可以給出正確的走法。



圖十四 ~ 二十三：殘局走法圖

五、結論與生活應用

一、結論：

(一) 對於大多數殘局可以搜索出正確的路

透過上面一些執行結果，我們看到程式在處理大部分殘局時都可給出一條必勝的路，但其缺陷確實存在。如同上面提到過的，我們設計的程式會給出所有它認為可能的必勝法，但有些結果看上去就不那麼的必勝，例如黑方在某步可能有更好的移動方法，但出於某些原因，程

式將其視為最佳策略而呈現於最終結果中，又或者是圖十九至二十八展示出來的，紅方的移動並非最佳，其解決會隨著搜索深度增加而難度遞增，但這確實是我們應解決的問題。

(二) 演算過程過於繁複

在我們測試執行某些殘局的時候，程式的缺點也很明顯，即偶爾會發生運算過久導致程式卡頓的情況，由於演算法的計算複雜度以及電腦性能所導致，這是我們接下來所必須面對以及改善的地方。

二、生活應用：

(一) 學習電腦下棋的步驟

誠如研究目標所言，我們渴望在這個研究中學習電腦的下法。從上面許多的範例都可以看到，我們的程式在此方面可一步一步地展示走法，雖然我們沒有設計出優秀的圖形介面，但至少可利於使用者快速看出移動方法。

(二) 設計下棋機器人的可能性

根據前面多種不同的殘局實驗過後，我們可以顯而易見地看出程式是可以順利運行的。在這個基礎上，若是再加上生活科技的應用，或許有希望設計出一款可以解決殘局的機器人。

參考資料

柳大華 (2013)。象棋殘局寶典。成都時代出版社。

王嘉良、李中健、王國順 (編) (1993)。象棋殘局大全。成都時代出版社。

程式人雜誌 (2014 年 7 月)。電腦下棋的關鍵：Min-Max 對局搜尋與 Alpha-Beta 修剪算法。

象棋教室 (2022 年 2 月 16 日)。舉例說明：特級大師也會上當的江湖殘局【象棋教室】 [影片]。Youtube。

教育部國語辭典簡編版 (2021)。<殘局> 辭典檢視。

中國象棋實用殘局 (2017 年 6 月 5 日)。單兵例勝孤將。

賴隆億 (2012)。電腦象棋棋譜分析之研究。國立交通大學資訊學院：碩士論文