

國立交通大學

資訊學院 資訊學程

碩士論文

電腦象棋棋譜分析之研究

A Study to the Analysis of Computer Chinese Chess Records

研究生：賴隆億

指導教授：吳毅成 教授

中華民國一百零一年一月

電腦象棋棋譜分析之研究

A Study to the Analysis of Computer Chinese Chess Records

研究生：賴隆億

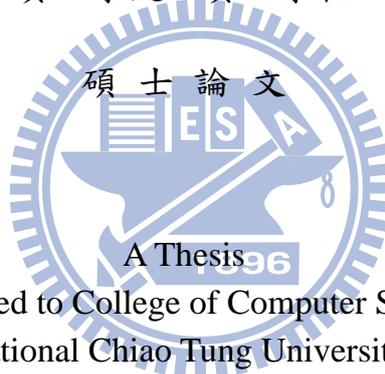
Student : Long-Yi Lai

指導教授：吳毅成

Advisor : I-Chen Wu

國立交通大學

資訊學院 資訊學程



Submitted to College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

January 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年一月

電腦象棋棋譜分析之研究

研究生：賴隆億

指導教授：吳毅成

國立交通大學

資訊學院 資訊學程碩士班



本論文的目的在於協助電腦象棋程式設計者，快速定位問題棋步或問題棋步區間。由於目前電腦象棋 AI 棋力已高於程式設計者，並且在大量對戰後，因此太多棋譜很難人工去復盤來找出問題點。本論文分析象棋棋譜，根據棋譜內的棋子位置，子力變化，審局分數變化，主要變例等資訊，來找出問題棋步。從實驗結果分析，經由此分析程式，可以協助程式設計者快速找出問題棋步，進而協助電腦象棋不斷的增進棋力。

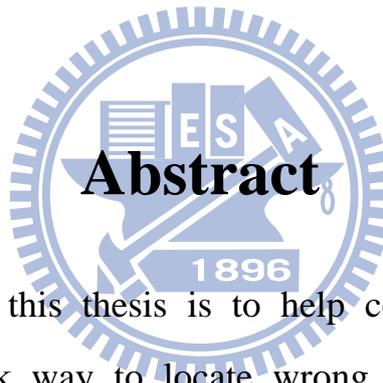
A Study to the Analysis of Computer Chinese Chess Records

Student: Long-Yi Lai

Advisor: I-Chen Wu

Degree Program of Computer Science

National Chiao Tung University



The main goal of this thesis is to help computer Chinese Chess developer find a quick way to locate wrong moves or wrong move intervals. Since Computer Chinese Chess program is already stronger the programmer in the most cases. It's hard to find the program bugs from Chinese Chess records. To improve program strength, we analyze the positions of pieces, principal variations according to games records.

誌謝

首先要感謝我的指導教授，吳毅成博士，由於他不厭其煩的細心指導，導引我走向正確的研究方向，並且經常在百忙之中抽空給我指導和討論，這篇論文才得以順利完成。同時也非常感謝各位口試委員：林志敏教授、陳隆彬教授、汪益賢博士的指正與建議，並撥時間來參與我的口試，使得本論文不完善與錯誤得以修正。

此外特別要感謝曾汶傑與陳建宇學長，在論文研究期間，分別給予電腦象棋與象棋專家知識並提供的寶貴意見和指導。也要感謝實驗室所有給與我幫助的人，特別是實驗室象棋組甘崇緯、郭青樺、林柏翰，在電腦象棋研究領域一起討論切磋。

接著要感謝我的家庭，我的父母與哥哥、姊姊們的養育與照顧，以及這段時間的加油與打氣，特別是在口試前兩週上天堂的阿母，保佑我一切順利及在心靈上給我力量。最後要感謝我的太太佳幸，我將這份榮耀獻給她，感謝她的體諒與鼓勵，讓我堅持到最後一刻，更感謝她奉獻於這個家庭的一切，負擔大部分女兒思妤與兒子元彥的教養，讓我無後顧之憂。

再一次感謝所有幫助過我及給我鼓勵的老師與親友，謝謝您們！

目錄

摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vii
表目錄.....	ix
第一章 緒論.....	1
1.1 象棋介紹.....	1
1.2 電腦象棋.....	1
1.3 研究動機.....	3
1.4 主要貢獻.....	4
1.5 論文大綱.....	4
第二章 研究背景.....	5
2.1 電腦象棋程式設計.....	5
2.2 搜尋.....	6
2.2.1 Minimax Search.....	6
2.2.2 Alpha-Beta Search.....	7
2.2.3 搜尋深度.....	8

2.2.4	Principal Variation.....	9
2.3	審局函數.....	10
第三章	電腦象棋棋譜分析與研究.....	11
3.1	問題分析.....	11
3.2	棋譜分析器.....	12
3.3	開局階段.....	13
3.3.1	吃/兌/失子分析.....	14
3.3.2	脫譜後分數分析.....	14
3.4	中殘局階段.....	14
3.4.1	前後手分數差.....	15
3.4.2	盤面最高分分析.....	16
3.4.3	特殊棋形.....	16
3.4.4	吃/兌/失子分析.....	17
3.4.5	主要變例分析.....	18
第四章	實驗結果與分析.....	20
4.1	棋譜來源.....	21
4.2	棋譜分類.....	21
4.3	關鍵棋步偵測.....	23
4.3.1	開局庫分析.....	24
4.3.2	子力變化分析.....	26

4.3.3	特殊棋形分析	28
4.4	關鍵棋步偵測	29
4.5	問題棋步棋形分析	30
4.5.1	進攻棋形	30
4.5.2	防守棋形	32
第五章	結論與未來展望	34
參考資料	35



圖目錄

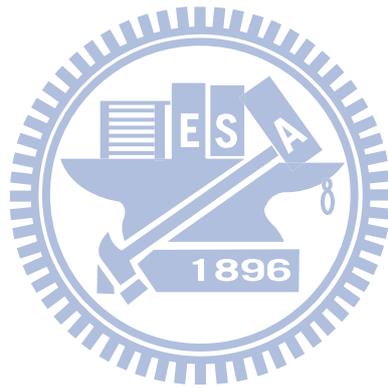
圖 1. 電腦象棋.....	6
圖 2. Minimax Search Tree.....	7
圖 3. Alpha-Beta Search.....	8
圖 4. Principal Variation.....	10
圖 5. 棋局為負之魚骨圖.....	12
圖 6. 棋譜分析器模組圖.....	13
圖 7. 窩心馬.....	17
圖 8. 車鎮窩心炮.....	17
圖 9. 俸鎮窩心馬.....	17
圖 10. 炮鎮窩心馬.....	17
圖 11. 俸鎮車馬.....	17
圖 12. 空頭炮.....	17
圖 13. PV 分析流程圖	19
圖 14. 棋譜分析器流程圖.....	20
圖 15. 問題開局庫的棋步.....	24
圖 16. 問題開局庫脫譜後.....	25
圖 17. 開局庫脫譜後與審局銜接.....	26
圖 18. 吃子步分析.....	27

圖 19. 吃子步分數曲線分析圖.....	27
圖 20. 俾鎮車馬盤面.....	28
圖 21. 分數曲線.....	28
圖 22. 關鍵棋步偵測統計.....	29
圖 23. 疊炮加雙車棋形.....	30
圖 24. 空頭炮棋形.....	31
圖 25. 當頭炮棋形.....	31
圖 26. 走子頻繁.....	31
圖 27. 出車太晚.....	32
圖 28. 未退馬吃兵.....	32
圖 29. 紅方棄兵種敗因.....	33
圖 30. 相需適時換邊.....	33
圖 31. 疊包加雙車攻勢.....	33
圖 32. 未換子求和.....	33



表目錄

表 1. 棋類整體複雜度.....	2
表 2. 子力價值分數表.....	15
表 3. 中國象棋開局編號.....	21
表 4. 棋謀為紅方的勝敗和統計.....	22
表 5. 棋謀為黑方的勝敗和統計.....	23



第一章 緒論

象棋 (Chinese Chess) 主要流行於華人與越南地區的兩人對奕棋類遊戲，與橋牌、圍棋、西洋棋、國際跳棋為 2008 年首屆世界智力運動會比賽項目[26]。另外電腦象棋[3][11][12][13]在 1989 年首屆電腦奧林匹亞至今每年亦會舉辦比賽。英國 Brain Games Network PLC 曾經預估，全世界約有兩億多人會下象棋[27]，可見象棋亦是一個相當熱門的遊戲。

1.1 象棋介紹

象棋是屬於資訊完全公開 (Perfect information) 的兩人棋盤類對奕遊戲，其棋盤是由九條直線與十條橫線相交，紅黑雙方各有七種兵種，16 顆棋子。分別有一顆帥 (將)、兩顆仕 (士)、兩顆相 (象)、兩顆俥 (車)、兩顆馮 (馬)、兩顆炮 (包)、五顆兵 (卒)，各兵種走法各有不同，目前棋規為紅方先手，雙方輪流下一合法步，目標為吃到對方的帥 (將)，或讓對方無合法步可走，即為獲勝。另外還有會因棋規來判定勝、負、和，目前國際上比賽象棋常用的是亞洲棋規[28]。

1.2 電腦象棋

電腦棋類[10]對局是人工智慧領域中非常重要的領域。IBM 團隊所開發的西洋棋程式，深藍 (Deep Blue) 在西元 1997 年，以兩勝一負三和打敗西洋棋世界冠軍 Garry Kasparov，可說是人工智慧發展的一個重要里程碑。

象棋與西洋棋的規則，走法類似，而電腦西洋棋發展甚早，不少發展

經驗都可從中學習，且從棋類的整體複雜度比較（如表 1），象棋是介於將棋（Shogi）與西洋棋（Chess）之間[4][16]，象棋將是下一個電腦程式可能擊敗人類世界冠軍的棋類遊戲[3]。

表 1. 棋類整體複雜度

棋盤類遊戲	Game Tree Complexity
Go（圍棋）	10^{360}
Shogi（將棋）	10^{226}
Chinese Chess（象棋）	10^{150}
Connect6（六子棋）	$10^{140-188}$
Chess（西洋棋）	10^{123}
Go-Moku（五子棋）	10^{796}

國內電腦象棋的研究從 1980 年以後開始起步[3]，在國內，宏碁電腦與象棋協會於 1988 年首次舉辦程式對程式比賽，在國際上，則從 1989 年開始舉辦第一屆電腦奧林匹亞，即有電腦象棋項目，一直到今。茲就目前定期舉辦的電腦象棋重要比賽，列舉如下：

1. International Computer Games Association（ICGA）從 1988 年起，每年舉辦 ICGA 電腦奧林匹亞。
2. 中華民國人工智慧學會與國內大學自 2004 年起，每三年舉辦一次世界電腦象棋爭霸賽。

3. 中國人工智慧學會自 2006 年起，每年舉辦中國機器博弈錦標賽
4. 台灣電腦對局學會自 2011 年起，舉辦 TCGA 電腦對局比賽。

而人腦對電腦的象棋比賽，自 1999 年起開始舉辦，在 2010 第 11 屆人腦對電腦象棋大賽由國手劉國華（八段）、馬仲威（七段）與象棋程式象棋世家（鄭明政開發）、鷹眼（曾汶傑開發）對奕，結果為一勝兩和一負，勢均力敵以平手收場。

另外電腦程式棋天大聖（2006, 2007 年電腦奧林匹亞金牌）在 2007 年與象棋特級大師吳貴臨對弈兩局，皆以和局收場，顯示在不久的將來，電腦象棋程式打敗人類的象棋世界冠軍將是指日可待。

1.3 研究動機

本實驗室所開發的電腦象棋程式「棋謀」，積極的參與國內外的電腦象棋競賽。雖然在 2009、2010 年 ICGA 拿下銅牌，但在網路上與玩家大量對戰，以在 2011 年三到六月的數千局對戰中，勝敗和的比率為 75.7%、12.3%、12.0%，勝率雖高，但也顯示有許多盤輸掉的棋局可以分析做改進。

另外在程式每次修改後的不同版本，在做程式自行對戰時亦會產生大量的棋譜，也需對輸掉的棋局做分析。面對如此大量的棋譜，對於程式設計者所面臨的問題有二，其一為目前電腦象棋程式棋力已高於程式設計者，不易快速發現問題棋步所在，另一問題為太多棋譜由人工復盤去找出問題點是費時費力的方法。因此本論文主要探討如何快速找出棋譜中不佳的棋步，提供程式設計者快速分析問題。

1.4 主要貢獻

此篇論文的主要貢獻為分析棋譜，根據棋譜內的棋子位置，子力變化，審局[15][20]分數變化，主要變例等資訊，來找出問題棋步。從實驗結果分析，經由此分析程式，可以協助程式設計者快速找出問題棋步，進而協助電腦象棋不斷的增進棋力。

1.5 論文大綱

本論文第一章緒論介紹象棋及電腦象棋，並說明研究動機，主要貢獻。第二章介紹研究背景，包含電腦象棋程式設計、Minimax、alpha-beta、Principal Variation（主要變例）演算法與象棋審局函數[2]。第三章說明電腦象棋棋譜分析器如何找出問題棋步。第四章將就本論文實驗結果與分析作討論。最後，第五章提出總結以及未來可發展、研究的方向。

第二章 研究背景

2.1 電腦象棋程式設計

電腦象棋設計與人下象棋大致相同，一般將棋局分為開局、中局、殘局三個主要階段。在這不同階段中，可用特定策略來下棋，如圖 1。大多數電腦象棋程式的開局階段，會以開局庫[6][7]來下棋，除了可以將思考時間，運用到中、殘局，若是開局譜是收集自特級大師或是象棋高手所下過的，是可以增強攻擊或是防守陣式的佈局。當然，開局階段也是可以直接讓電腦象棋程式思考下棋棋步，而程式思考主要是靠搜尋(Search)[5][8]和審局(Evaluation)，審局通常使用審局函數，之後會介紹電腦象棋主要使用的搜尋演算法與審局函數。

在中局階段，也常是棋局的決勝階段。對電腦象棋決勝點是搜尋與審局。電腦象棋常用 Alpha-beta 演算法來搜尋最佳步，隨著電腦運算能力的提升，搜尋已不再是瓶頸點，甚至可以多搜尋一、兩層。相對的，審局函數才是主要左右電腦象棋勝負的關鍵，因審局函數常需具備象棋的專家知識，然棋局千變萬化，一來很難將各種狀況變化實作到審局函數，二來電腦象棋程式設計者不易具有特級大師或象棋高手的專家知識。

在殘局階段，不一定每盤面都會走到，棋局有可能在中局以分勝負。若使用殘局庫[24]對搜尋將是overhead，且目前電腦運算能力已大幅提升，直接用搜尋與審局也是一種做法。

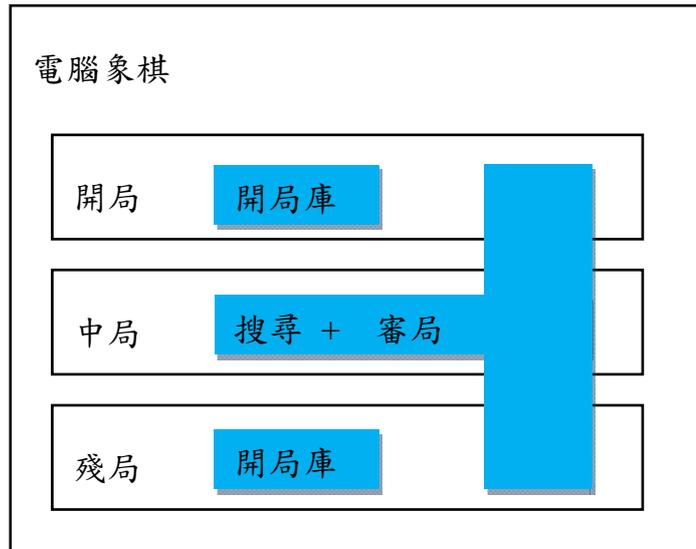


圖 1. 電腦象棋

2.2 搜尋

就人類下象棋而言，若能比別人往後多想幾手，其獲勝的機會也大幅提高。對電腦象棋而言，就是在有限的時間內，對局樹（Game Tree）要能多搜尋幾層。要加快搜尋速度或加深搜尋深度，除了使用高階的硬體設備外，更重要的是需要優異的搜尋技巧。主要使用 minimax、alpha-beta，以及逐層加深（Iterative Deepening）、殺手法則（Killer Heuristic）、同形表（Transposition Table）、主要變例（Principle Variation Search）、平靜搜尋（Quiescent Search）等[9][18]。

2.2.1 Minimax Search

將雙方對奕的問題轉換成搜尋問題，在雙方都是會找對自己最有利著法的前提下，這就是以 Minimax Search。圖 2 是一個 minimax search tree

的例子，以方形節點表示 Max Node 也就是我方的節點，以圓形節點表示 Min Node 也就是對方的節點，方形圓形節點內的數字皆表示我方的分數，分數是從葉節點 (Leaf node) 向上更新，而分數是從評估函數取得的。

在 Max Node 也就是我方觀點，將會選擇對自己最有利的子節點 (Children Node)，也就是分數最高的子節點，相反的，在 Min Node 也就是對方觀點，一樣會選擇對方最有利的子節點，也就是我方分數最低的子節點。假設在雙方都不會犯錯的前提下，雙方皆會選擇對自己最有利的子節點，在更新完的每個節點分數代表遊戲進行的該節點的分數。如圖 1 中的紅色路徑，代表雙方皆選擇對自己最有利的走法。

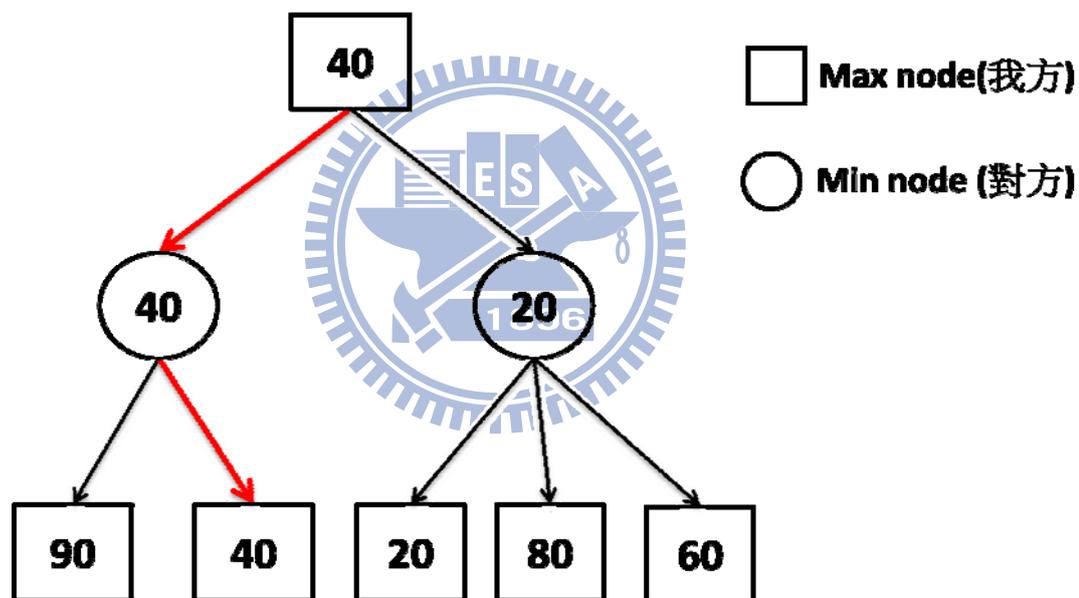


圖 2. Minimax Search Tree

2.2.2 Alpha-Beta Search

Alpha Beta Search 是個有效增進 Mini-Max Search 的演算法，以圖 3 為例，第二層的圓形 min node 要取最小的，已經找完最左邊第一條路徑是

20 分，也就代表這個 node M 分數至少 20 分，而第二條路徑 min node N 的子節點其分數為 12 分，換句話說，node N 的分數最多 12 分，但是 max node M 分數至少 20 分，所以 node M 不會選擇 node N，所以 node N 的其他子節點分數是多少，不會影響 node M 的選擇，也就不需要做搜尋，故可以做修剪 (prune)。

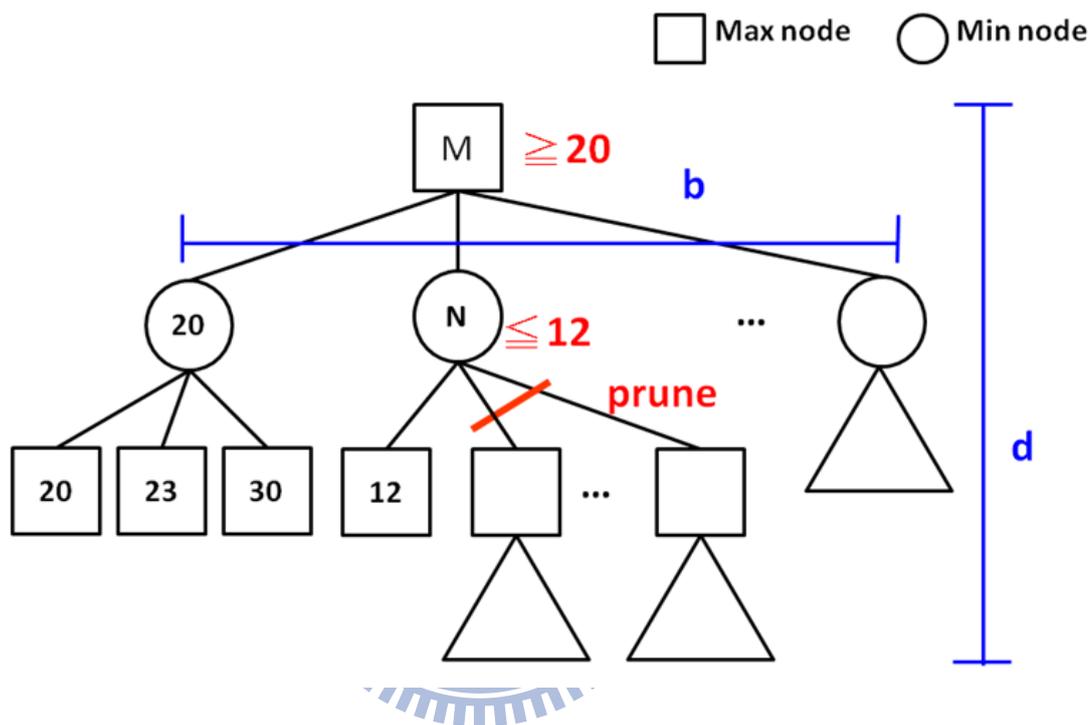


圖 3. Alpha-Beta Search

此方法可以修剪許多子樹 (sub-tree)，進而節省搜尋的時間。若 b 為搜尋寬度 (branching factor)， d 為搜尋深度 (search depth)，在最佳情況下，最多可以將原本的時間複雜度從 $O(b^d)$ 改良至 $O(b^{d/2})$ [4]。

2.2.3 搜尋深度

象棋每手棋的合法步介於 32 到 42 步之間，每局平均有 95 手，從表 1

可知整體複雜度約為 $10^{150}(38^{95})$ [25]，要完全解此對局樹是不可能的。假設目前 AI 可以搜尋到第 20 層，另外一個問題為有些的著手可能要搜尋到第 20 層後，才可找到最佳著手。假設搜尋到第 20 層所需時間為 60 秒，由前一節可知時間複雜度為 $O(38^{20/2})$ ，第 21 層時間複雜度變為 $O(38^{21/2})$ 或 $O(38^{1/2} \times 38^{20/2})$ ，所需時間即為 $38^{1/2} \times 60(6.16 \times 60)$ 秒，為 20 層的 6 倍，再往深層搜尋，搜尋所需時間更是等比級數成長。但常礙於比賽時間與象棋整體複雜度，搜尋程式不僅無法展開整棵對局樹，連多搜尋 1 到 2 層，即為在搜尋方法、電腦效能或平行處理上的大進展，而搜尋深度不足的問題，將會導致以下的幾種現象[18]：

- (1) 走出無關痛癢的緩著，導致失子。
- (2) 水平效應，胡亂棄子。
- (3) 不敢棄子攻殺，坐失得勝良機。
- (4) 將軍效應，降低棋力。

而 (1) 與 (2) 常是敗著的原因所在，在之後的棋譜分析程式中會實作偵測失子/棄子的著手，並判斷是否為關鍵棋步。

2.2.4 Principal Variation

程式判斷對雙方最好且預期玩家會走的路徑，紅色路徑為此 game tree 的 Principal Variation (主要變例)，如圖 4 所示。之後在棋譜分析時，將會以主要變例資訊來得知，AI 在每一手的想法為何。

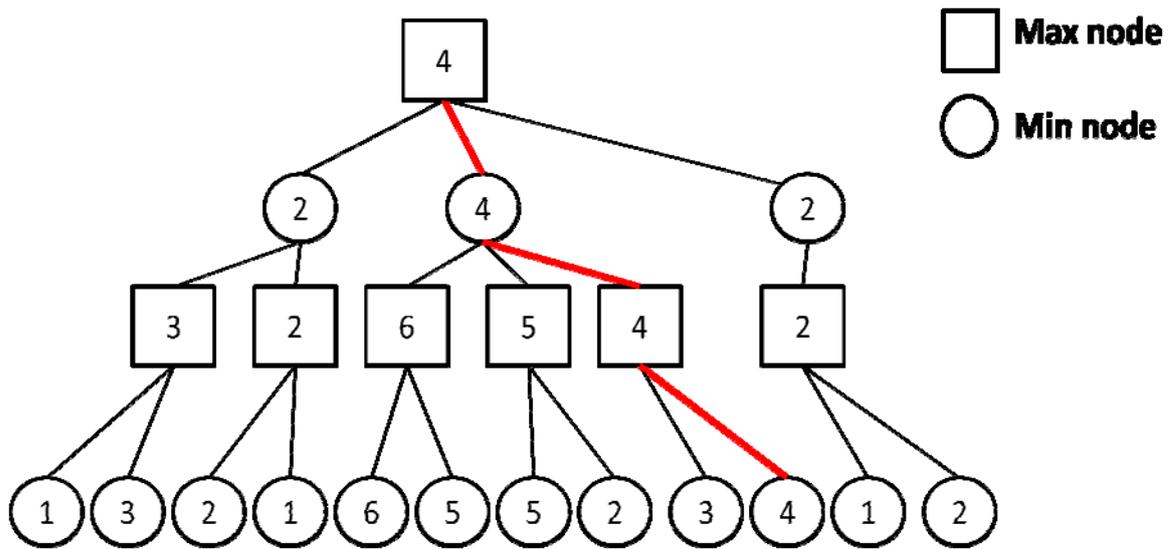


圖 4. Principal Variation

2.3 審局函數

審局是由審局函數 (Evaluation Function) 轉化成分數來評定盤面形勢子力的優劣。審局函數做法有下列幾種：

1. 找象棋高手，由象棋高手給定審局函數的各種參數值或特徵盤面。
2. 統計方式，如鄭明政[21]以統計的方式，用程式建立子力位置分數表。
3. 基因演算法，如高暉倫[19]以基因演算法的方式，使用選擇(selection)、交換(crossover)和突變(mutation)程序。經由對戰後，留下最佳子代，以建立審局參數表，如子力位置分數表、子力基本分、子力靈活度分、連環馬分、窩心馬分等等。
4. 增強式學習法，大多用 temporal-difference (TD) learning，如陳漢鴻[22]，選定 pattern 盤面來做學習，也是經由不斷對戰來修正審局參數表，進而提升棋力。

但主要是以做法 1、3、4 來不斷的改進審局函數。作法 1 的問題為費時費工且不一定很精確的轉換成各種參數值。故較常見的是用基因演算法或 TD learning 離線學習的方式，經由不斷對戰後，修正出最佳審局參數表，供 AI 在比賽時使用。而本論文所提的方法為，離線分析 AI 在對戰時的棋譜，也就是對奕時的各種資訊，試著找出棋局中不佳的著手，讓程式設計者可以快速的去檢視審局函數是否有問題。

第三章 電腦象棋棋譜分析與研究

在這一章節，先以魚骨圖（Fishbone Diagram）或稱石川圖（Ishikawa Diagram）[30]，來列出造成棋局為負的原因，並試著將這些原因實作到棋譜分析器中，分析棋謀在群想網站與玩家對奕棋譜，在 2011 年三月到六月數千盤棋譜，但主要分析棋謀為負的盤面，從中找出不佳的著手，節省程式設計者分析棋譜的時間，進而不斷的提升棋謀的棋力。

3.1 問題分析

魚骨圖為一個簡單結果與成因圖形表示法，是日本品管大師石川馨（Kaoru Ishikawa）所創。如圖 5 所示，魚頭向右表示結果，而組成魚身的大骨，即為造成此結果的主要原因，是應用在問題解決思考流程上。因為造成一盤棋局結果為負的原因，可能有一個到數個，故可運用此手法將可能的原因列出。

先在 2011 年一、二月棋謀為負的棋譜中，以人工找出大部分造成棋局結果為負的原因，再與五段象棋專家討論，參考其意見，釐清出造成棋局結果為負的主要要因，以魚骨圖列出真因圖，如圖 5 所示。此魚骨圖魚頭表示棋局為負，魚骨表示主要原因。下一節將會根據這些原因，實作到棋譜分析器，試著去找出造成這些原因的不佳著手。

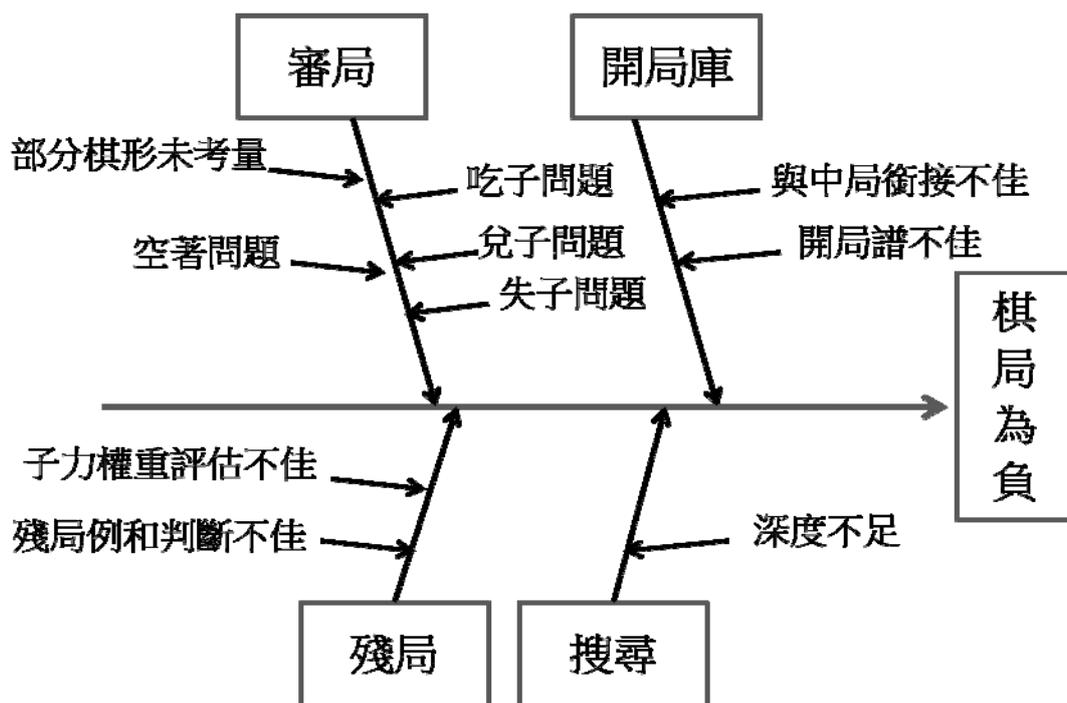


圖 5. 棋局為負之魚骨圖

3.2 棋譜分析器

根據第三章第一節已知的原因，實作了一個棋譜分析器，如圖 6 所示，離線讀入棋譜後，經由各模組判斷是否有問題棋步的存在，各模組作法將於本章下一節起說明。棋譜分析器主要分析棋謀與玩家對戰結果為負的棋譜，藉由開局階段與中殘局階段走子棋步分析，另外在中殘局針對棋譜內的資訊，例如分數變化、主要變例、特殊棋形以及吃子、兌子、失子做分析。

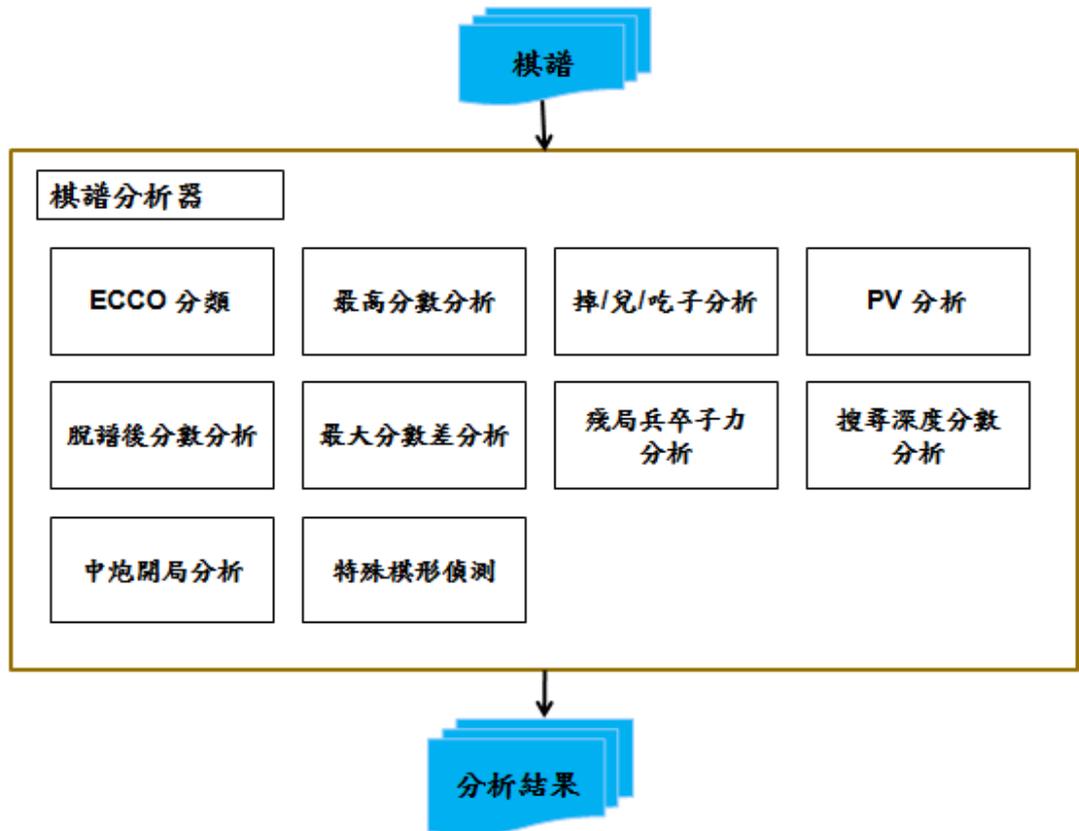


圖 6. 棋譜分析器模組圖

3.3 開局階段

棋書[1]，常提到的一些開局概念，列舉如下：

1. 出動大子，占據要位。
2. 均衡出子，子勿多動。
3. 馬勿輕進，炮勿輕發。
4. 講究協調，注意聯絡。
5. 得子失先，得不償失。

故開局的分析主要會判斷大子的俸（車）有盡早開出、或是同一顆子是否頻繁的走動、脫譜（out of book）後的分數是否即居於劣勢或是不同

開局走法（如中炮開局或是仙人指路開局）對勝率是否影響。

3.3.1 吃/兌/失子分析

由於棋譜使用開局庫的來走子的開局階段，沒有紀錄審局分數，所以開局階段的子力變化，吃、兌、失子分析主要針對是否有子力的損失。

3.3.2 脫譜後分數分析

由於從象棋高手下過的棋譜收集而成的開局譜，其走子佈局有其目的、計畫，以便中局的攻殺與防守陣勢。但是 AI 程式並未被告知其計畫為何，因此當脫譜時，無法得知開局譜的下一手走法，此時由中局程式接手，當審局函數無法精準的評估出最佳著手，會給予對手可趁之機[6]。

偵測電腦象棋在脫譜後（即結束開局庫棋步），其分數是否脫譜後即落後很多或是越走分數愈小。在脫譜後，分數越來越低的情形有二，一是開局庫有問題，此開局譜的敗率可能大於勝率，卻被收錄到開局庫，另一原因為 AI 程式與開局庫的銜接，此開局譜脫譜後，審局函數無法找到最佳的棋步。

3.4 中殘局階段

中殘局主要針對子力變化，特殊棋型，分數差來作分析。棋謀的子力表如表 2，由於帥（將）被對方吃掉棋局結果為敗，因此帥（將）的價值，大於其他子力的總和。

但隨著棋局變化，子力價值的分數亦會有所變化，例如開局炮的子力價值分數是比馬高，但在殘局時，馬的子力價值分數反而比炮高。另外表 2 得知，一馬換兩相是不一定划算，雖說「一馬換二象，其勢必英雄」，但這是在中殘局時在作攻殺時對整個盤面的「勢」很有幫助。

表 2. 子力價值分數表

兵種	價值分數
帥/將	20000
仕/士	200
相/象	200
俥/車	1000
馮/馬	450
炮/包	450
兵/卒	100

3.4.1 前後手分數差

從子力價值分數表，可以訂定分析前後手分數差的 Threshold 高低，一開始先訂前後手差異超過 500 分以上的棋步來分析，也就是差異超過一顆馬（馮）或包（炮）的子力價值分。等將問題較嚴重的審局解掉後，之後再將 Threshold 往下修訂，目前是將 Threshold 定為 300 分，若前後手

分數差，略超過半顆馬（馬）或包（炮）的子力價值分，就可能有問題棋步存在。

3.4.2 盤面最高分分析

Threshold 先訂為 500 分，若盤面最高分大於 Threshold，代表曾經比對手多一顆馬（馬）或包（炮）的子力價值分，很明顯在這盤面最高分的前後幾手，可能有下不好的地方，讓對手有逆轉的機會。

3.4.3 特殊棋形

選擇已知的特殊棋形來做分析，是否對後續局勢有影響。例如棋諺有云：「馬入宮，不死也帶傷」、「馬入宮心，老將發昏」，說明窩心馬（圖 7）為馬入己方九宮中心點，將會帶來相當不利的後果。窩心馬演變的棋形有俥鎮窩心馬（圖 9）或炮鎮窩心馬（圖 10），更是受制於對方子力，己方九宮子力無法發揮防守功能。「空頭炮，兇惡難當」，空頭炮（圖 12），因讓對方的士象無法到中路防守，若在配合俥或馬，攻勢更凌厲，常可抽子。而圖 11 所示的俥鎮車馬將使黑方的車馬陷於位置不佳，靈活度奇差的困境。

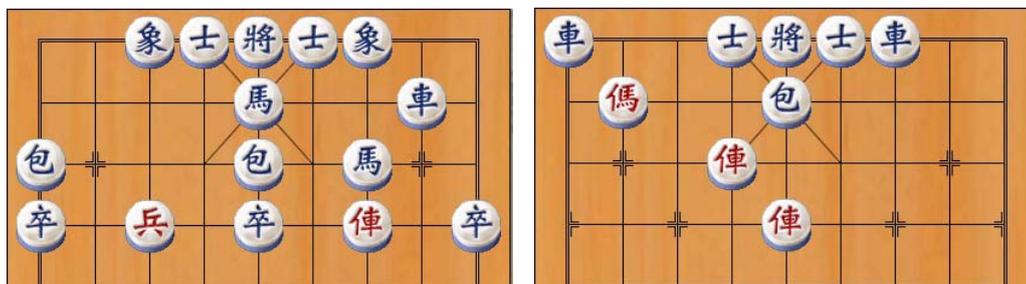


圖 7. 窩心馬



圖 8. 車鎮窩心炮



圖 9. 俾鎮窩心馬



圖 10. 炮鎮窩心馬

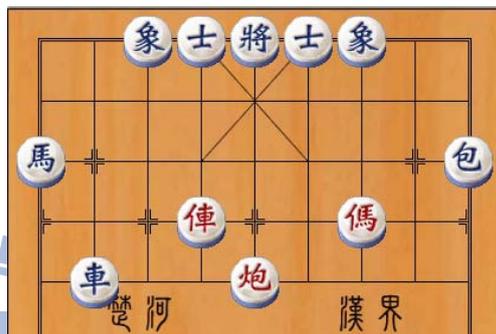


圖 11. 俾鎮車馬

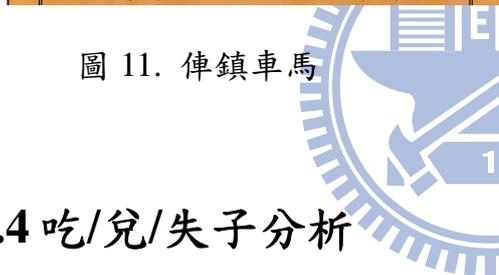


圖 12. 空頭炮



3.4.4 吃/兌/失子分析

子力變化如吃子、兌子、失子是很明顯的有形的變化。但象棋重要的優先順序應為「殺勢先子」。「殺」為殺著，象棋走子的目標，即為吃掉對方的帥（將）。「勢」是為「殺」作準備的，有「先」的優勢，才能形成局面的「勢」，而「寧失一子，不失一先」。

單只從棋譜中分析子力變化，吃子不一定有利，失子不一定不好。因象棋上有抽象的「殺勢先」概念在其中。故吃子、兌子、失子還需搭配前後數手（ply）的分數變化來分析，

就吃子步來說，一般而言，吃子是對己方有利的著手，但是有三種情況下是不利的，

1. 開局時貪吃兵卒，開局時的兵卒價值是較低的，花太多著手吃兵卒會影響大子佈局的時效性。
2. 「死子勿急吃」，將棋步更有效的用在佈局或取「勢」上
3. 對方的棄子搶攻，搶先，也是對方採取「寧失一子，不失一先」的策略。

關於失子棋步分析，若不是己方棄子爭先的策略，通常都是被迫失子的不佳的著手。兒子棋步分析，除了判斷相互兒子的價值是否對等外，若對等但是之後得幾手，審局分數下降，是被對方「兒子消勢」不佳的著手，宜避免。

3.4.5 主要變例分析



棋謀每一步搜尋結果，都會列出其主要變例(Principal Variation, PV)，也就是在該手 AI 當時認為之後雙方會走的最佳步。根據此原理，PV 的資訊將是用來找 AI 問題的重要根據。棋譜中的 PV 分析可以用來分析問題棋步的做法有二：

第一在有特殊事件的 ply P_{i+1} (例如吃,掉,失子)，且分數之後一路往下掉，可做 PV 分析，如圖 13，在棋步 P_i 每次回推兩步，若在 P_{i-2} 的 PV 前幾步與棋步 P_{i-2} 到 P_i 相同，則繼續往前回推兩步。以圖 13 為例，可回推到 P_{i-2} 的 PV 前四步棋步為 C、D、E 與 F，與棋譜棋步相同，故推測在棋步 P_{i-2} 即有問題而造成在 P_{i+1} 可能有掉子的情形。

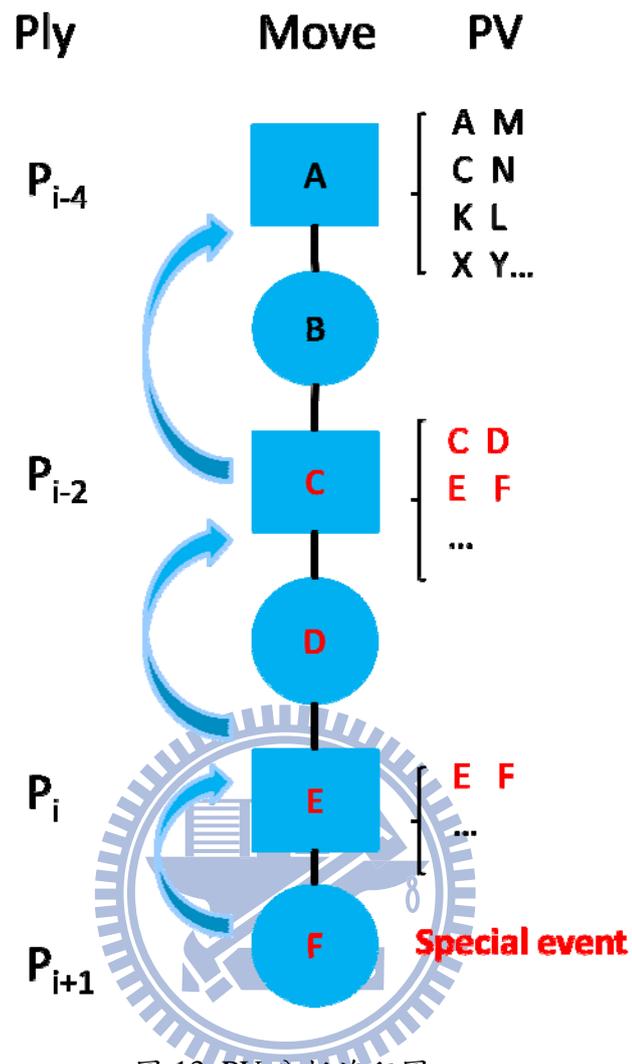


圖 13. PV 分析流程圖

第二個分析方法為模擬 PV 走完的結果，再以圖 13 為例，在 ply P_{i-4} 走 A 其步，若走完此步的 PV（棋步為 A、M、C、N、K、L、X、Y）預估有吃子步，但是在實際棋譜中沒有吃子步，反而在 ply P_{i+1} 時有掉子情形，此亦可偵測在此步的審局是有問題的。

第四章 實驗結果與分析

本章透過實作的棋譜分析器，如圖 14 的分析流程，將之前棋謀與玩家對奕結果為敗的棋譜做分析，從中找出問題棋步。同時五段的象棋高手也去分析棋謀為敗的棋譜，與棋譜分析器做比較，驗證棋譜分析器找出的問題棋步與實際的問題棋步差異有幾手。

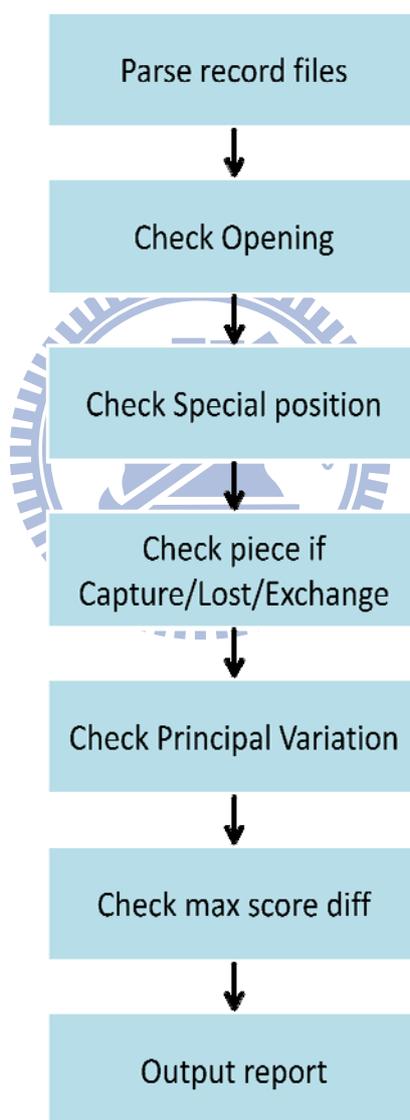


圖 14. 棋譜分析器流程圖

4.1 棋譜來源

收集 2011 年三月到六月，棋謀在群想網站與玩家對奕棋譜共數千筆棋譜，因要找出錯誤棋步，進而修正改善棋力，本實驗將主要從分析棋謀為負的盤面著手。

4.2 棋譜分類

中國象棋開局編號（Encyclopedia of Chinese Chess Openings，ECCO）是中國象棋開局類型的編號體系。反映了現代中國象棋的開局體系[27]。如表 3。每個分類可再從 00 編到 99，例如分類 C，中炮對屏風馬可從 C00 編到 C99。

表 3. 中國象棋開局編號

分類	開局分類名稱
A	非中炮類開局（非仙人指路局）
B	中炮對反宮馬及其他
C	中炮對屏風馬
D	順炮局和列炮局
E	仙人指路局

將所有棋譜以 ECCO，做分類，棋謀分別為紅、黑方的勝負和比例統計如表 4、表 5。由以下數據發現，棋謀為紅方在中炮對屏風馬的開局，

其敗率為 24.7% 將近平均敗率 12.4% 的兩倍。雖說以屏風馬抗衡當頭炮的開局佈局，自康熙年間的「梅花譜」開始至今，有其定式在，但從脫譜後（開局階段後）的分數分析，最高分為 129 分，最低分為負 91 分，顯示中炮對屏風馬局的開局庫沒有大問題，問題在脫譜後的中局程式思考是否夠精準銜接開局佈局。

另外棋謀為紅方在中炮開局中，有 30% 因三路或七路相被吃，導致半邊防守空虛而落敗，這亦是棋謀在中炮開局敗率較高原因之一。

表 4. 棋謀為紅方的勝敗和統計

分類	勝率	敗率	和率
A 非中炮類開局 (非仙人指路局)	75.3%	12.4%	12.3%
B 中炮對反宮馬及 其他	82.4%	9.1%	8.4%
C 中炮對屏風馬	63.6%	24.7%	11.7%
D 順炮局和列炮局	82.8%	2.5%	14.6%
E 仙人指路局	73.7%	14.0%	12.2%
全部不分類	75.7%	12.4%	12.0%

就棋謀為黑方的各分類開局中，無較明顯的差異，只有分類 D 仙人指路開局的敗率 14.4%，略高於棋謀為黑方的敗率 12.1%。統計數據符合紅方有先手優勢，且仙人指路開局特性為，紅方可根據黑方不同的行棋，伺機行事。實作棋譜分析器所找出的問題盤面列舉如下。

表 5. 棋謀為黑方的勝敗和統計

分類	勝率	敗率	和率
A 非中炮類開局 (非仙人指路局)	76.3%	13.6%	13.6%
B 中炮對反宮馬及 其他	80.0%	8.0%	8.0%
C 中炮對屏風馬	76.2%	12.6%	12.6%
D 順炮局和列炮局	78.7%	9.9%	9.9%
E 仙人指路局	73.6%	14.4%	14.4%
全部不分類	75.8%	12.1%	12.1%

4.3 關鍵棋步偵測

就實作棋譜分析器所找出的問題盤面列舉如下。

4.3.1 開局庫分析

分析出脫離開局庫後，分數是否已大幅落後，或分數是一直往下掉的。分數大幅落後的為用到有問題的開局庫，如圖 15 選到送俸的開局庫，到圖 16 脫譜後的分數已是負 726，之後分數更是一路往下，可以說勝負已定。



圖 15. 問題開局庫的棋步



圖 16. 問題開局庫脫譜後

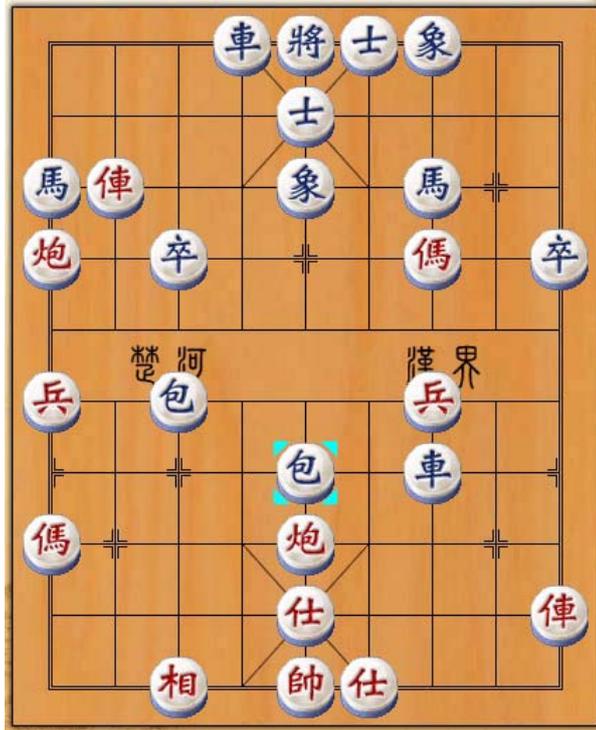
另一種情形為脫譜後分數上未負很多分，約半顆馬的子力分，但脫譜後分數是一路往下掉。此為審局函數無法精準的銜接開局庫，例如在脫譜後盤面如圖 17，此時棋謀為黑方略居劣勢，少一中兵，若策略為收拾兵或兌兵至少為和局，相反的黑方的卒被收光且對方上有三顆兵，此時勝負立判。



圖 17. 開局庫脫譜後與審局銜接

4.3.2 子力變化分析

吃子後分數卻一路往下掉，棋諺有云「寧失一子，勿失一先」。如圖 18 為例，雖然俥八平九可吃一隻馬，但是沒有考慮到包三平五，疊包加雙車的攻勢，局勢立即失控，可從圖 19 分數曲線看出，種下敗因。



思考 主要變例		棋譜
30. 包 9 平 5		
31. 俥 8 平 9		
32. 包 3 平 5		
33. 俥 9 平 8		
34. 前包 進 2		
35. 俥 8 退 3		
36. 車 4 進 7		
37. 馬 3 進 5		
38. 象 7 進 5		
39. 俥 9 平 8		
40. 包 2 平 3		
41. 俥 8 平 9		
42. 包 3 平 2		
43. 俥 9 平 8		
44. 包 2 平 3		
45. 俥 8 平 9		
46. 包 3 平 2		
47. 俥 9 平 8		
48. 包 2 平 3		
49. 俥 8 平 9		
50. 包 3 平 2		
51. 俥 9 平 8		
52. 包 2 平 3		
53. 俥 8 平 9		
54. 包 3 平 2		
55. 俥 9 平 8		
56. 包 2 平 3		
57. 俥 8 平 9		

圖 18. 吃子步分析

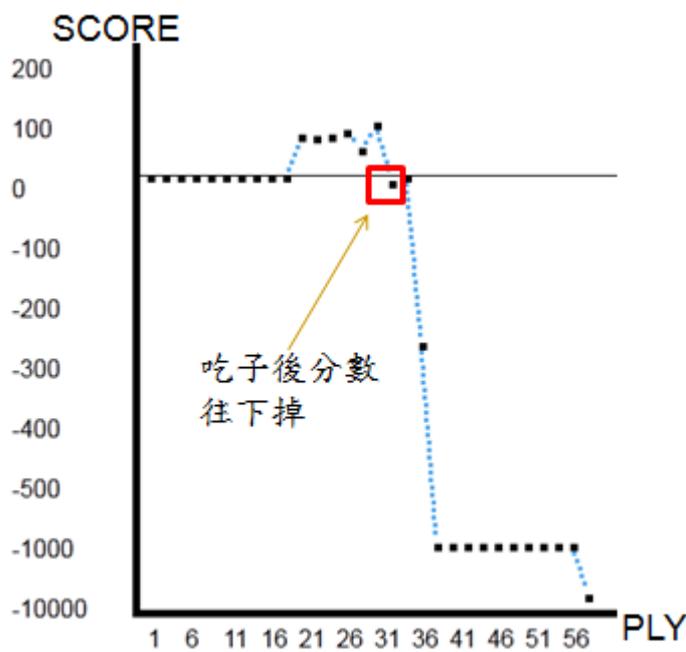


圖 19. 吃子步分數曲線分析圖

4.3.3 特殊棋形分析

如圖 20 為例，棋謀為黑方，因為未考慮到會被俾鎮車馬，導致車馬的靈活度相當低，盤面漸漸的失去優勢，圖 21 為這盤面的分數曲線。

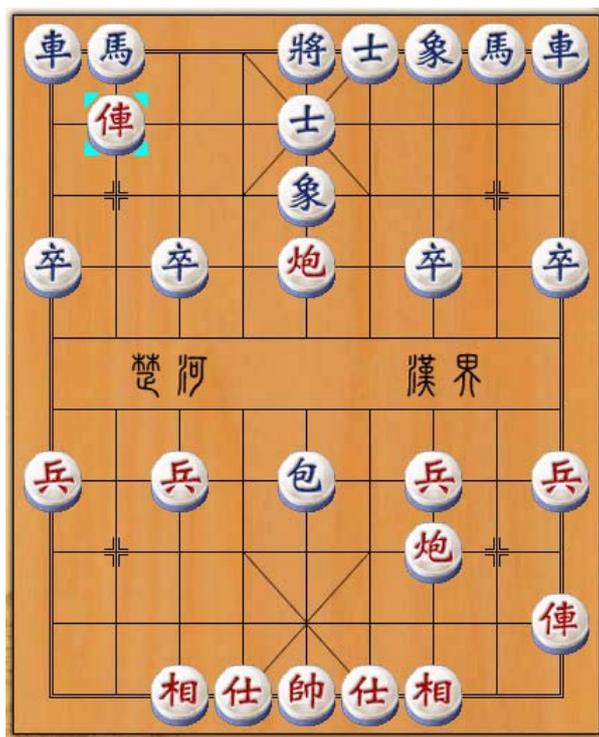


圖 20. 俾鎮車馬盤面

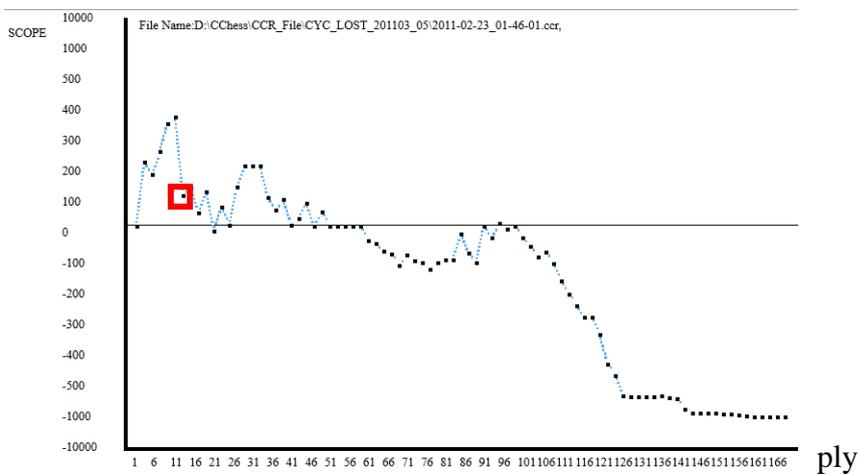


圖 21. 分數曲線

4.4 關鍵棋步偵測

本節將會討論棋譜分析器是否有真正找出的關鍵問題棋步。由五段的象棋專家分析 26 盤 AI 為負的棋譜，找出 AI 的問題棋步，再與棋譜分析器比較，我們以象棋專家分析結果當作答案，來統計棋譜分析器判斷的棋步與答案差異幾回合。分析結果如下圖 22：

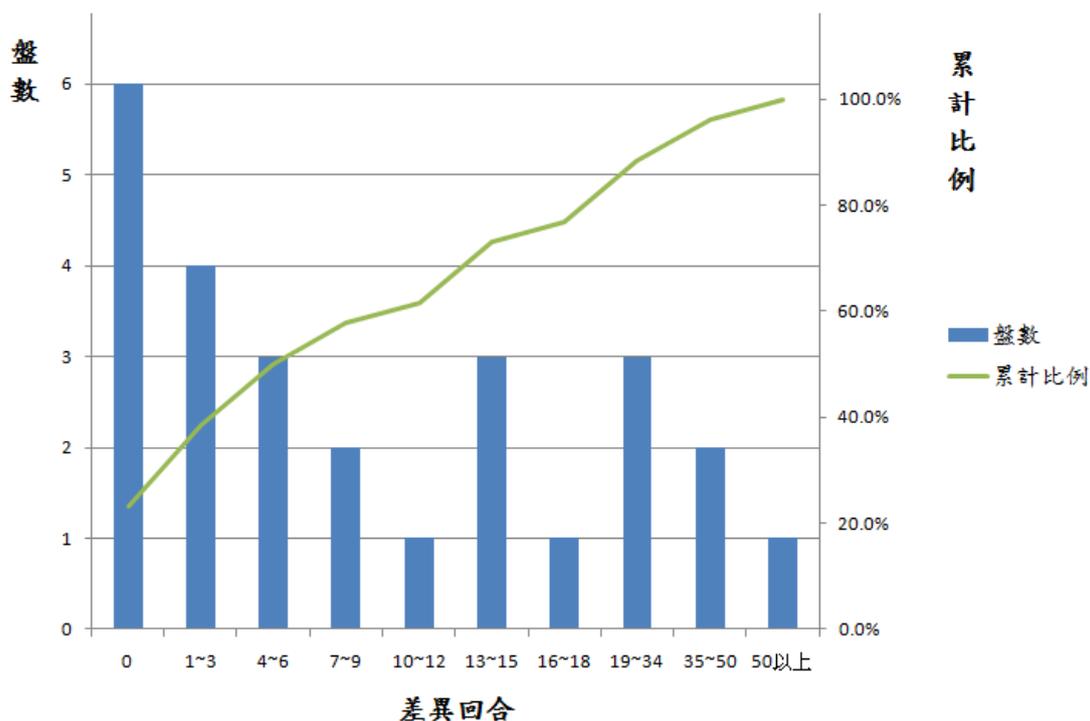


圖 22. 關鍵棋步偵測統計

以紅黑雙方各下一手表示一回合，完全與象棋專家分析的問題棋步相同的比例為 23.1%，在差異 10 回合內的目標下，準確度為 57.7%。預估差異超過 10 回合有 42.3%，原因為象棋 Game Tree 得知其走法千變萬化，目前第三章所描述的規則尚無法涵蓋所有問題棋步的盤面，尚需不斷實作已知盤面特徵至棋譜分析器來提升判斷準確率。例如目前沒實作疊包加雙車棋形，且棋謀亦未警覺到黑方這種棋形破仕後，對我方將造成嚴重的殺

傷力，如圖 23。持續加入新發現的棋形，優點為不斷的提升棋譜分析器判斷準確率，並持續用來檢驗棋謀是否有不斷的修正問題，缺點為需持續的費力費時實作新發現的問題棋形。

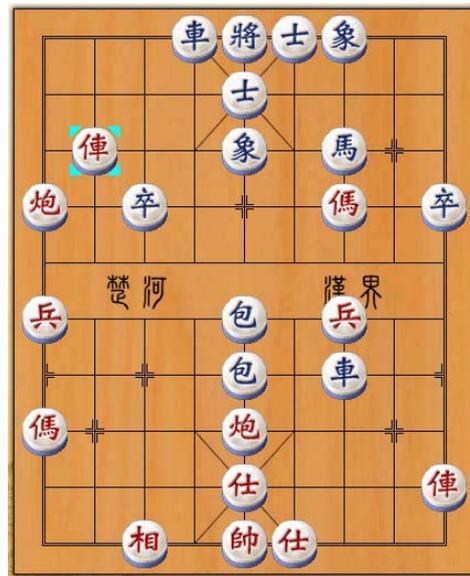


圖 23. 疊炮加雙車棋形

4.5 問題棋步棋形分析

本節將列出棋譜分析器發現的問題棋步棋形，供棋謀之後改善審局，另外這些 feature/pattern 盤面，將可作為未來開發 TD Learning 的資料，

4.5.1 進攻棋形

以下實例為棋譜分析器發現目前棋謀在進攻時的問題，列舉如下：

1. 空頭炮優勢未把握，如圖 24，未協調俥馬進攻，將優勢擴大，相反的，優勢漸失，敗局收場。
2. 當頭炮開局的相容易被攻擊，如圖 25，造成單邊空虛，導致敗局。



圖 24. 空頭炮棋形

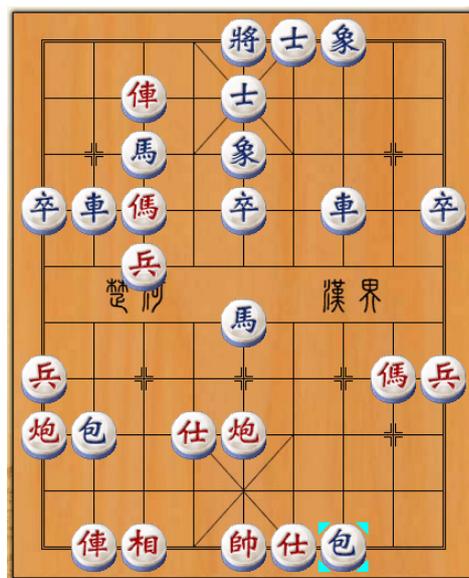


圖 25. 當頭炮棋形

3. 開局走子頻繁，如圖 26，紅俥在第 11 至 23 手走了五次，其他大子未即時佈局，敗局收場。

思考 主要變例		棋譜
1. 兵七進一	2. 卒3進1	
3. 炮八平五	4. 卒3進1	
5. 炮五進四	6. 馬3進3	
7. 炮五退一	8. 馬3進4	
9. 炮二平五	10. 馬8進9	
11. 俥九進一	12. 包2平3	
13. 俥九平八	14. 車9平8	
15. 仕六進五	16. 車1進2	
17. 俥八平六	18. 包8進2	
19. 馬八進九	20. 包3進7	
21. 俥六平七	22. 包3平1	
23. 俥七進三	24. 車1平2	
25. 馬二進一	26. 車2進7	
27. 仕五退六	28. 車2退2	
29. 仕六進五	30. 車2平1	
31. 俥七進二	32. 包8退3	
33. 後炮平七	34. 象3進1	
35. 炮七進二	36. 馬4進3	
37. 炮七平六	38. 包8平5	
39. 俥七平五	40. 包2平5	
41. 炮六平五	42. 馬3退5	
43. 兵五進一	44. 車8進7	
45. 俥一平二	46. 車8平3	
47. 俥五平六	48. 包5進4	
49. 仕五進四	50. 車3進2	
51. 帥五進一	52. 車3退1	
53. 帥五退一	54. 象1進3	
55. 俥二進七	56. 包1平8	
57. 俥二平六	58. 車1進2	
59. 後俥退六	60. 包8平4	

圖 26. 走子頻繁

4. 開局車在不開出，如圖 27，在第 41 手才出車，敗局收場。
 5. 殘局未把握吃對方兵機會，如圖 28，未退馬吃兵求和，導致敗局。

以上實例將可回饋給棋謀程式設計者作為改善審局的參考。

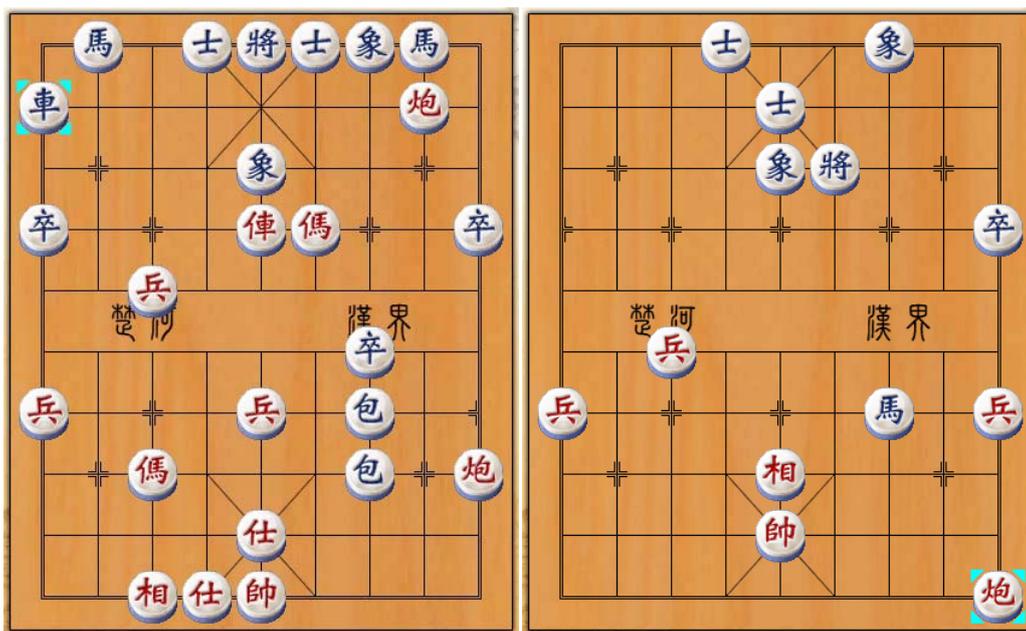


圖 27. 出車太晚

圖 28. 未退馬吃兵

4.5.2 防守棋形



以下實例為棋譜分析器發現目前棋謀在防守時的問題，列舉如下：

1. 俥鎮車馬棋形未避免，如圖 11。
 2. 中殘局主動送兵卒種敗因，如圖 29。
 3. 相需適時換邊防守，如圖 30，先鞏固防守陣形，再伺機進攻。
 4. 疊炮加上雙俥棋形，如圖 31。破仕後可用鐵門門或雙車殺著，因「缺士怕雙車」。
 5. 殘局未換子求和，如圖 32。紅俥未兌車求和導致敗局。
- 以上實例亦回饋給棋謀程式設計者作為改善審局的參考。

第五章 結論與未來展望

本論文所實作之棋譜分析器可以快速的找出部分問題棋步，因此在需檢討棋譜時，程式開發者可以先找到可能的問題棋步再去分析是否審局函數無法對特定的盤面做出正確的判斷。

如在第四章第五節問題棋步棋形分析列舉的實例，棋譜分析器確實偵測到棋謀審局函數在進攻方面空頭炮棋形未把握優勢，中炮開局敗率較高，且三路或七路相容易受對手攻擊而種敗因，中殘局兵卒價值計算過低而棄子，導致進入殘局纏鬥而落敗等等的問題。在防守方面，未偵測到受制對方俥鎮車馬於角落的影響，中殘局輕忽對方兵卒影響而未吃子，以及該兒子求和確沒兒子而落敗等等的問題。

將分析結果改進本實驗室研發的「棋謀」程式，改進棋謀部分審局缺失，增強程式棋力，在 2011 年的第十六屆國際電腦奧林匹亞（Computer Olympiad）中，獲得電腦象棋競賽之銀牌（在 2009、2010 年為銅牌）。

除了找出問題棋步外，這些問題棋步的棋譜將可作為將 TD（Temporal Difference）Learning 的 feature，協助 AI 提升棋力，朝向與人類特級大師棋力邁進。

參考資料

- [1] 金海英，象棋基礎我在北大講課，第一版，人民體育出版社，北京，2006。
- [2] 吳身潤，電腦象棋人工智慧程式設計，第一版，旗標出版股份有限公司，台北市，1996。
- [3] S.J. Yen, J. C. Chen, T. N. Yang and S.C. Hsu. "Computer Chinese Chess". ICGA Journal, Vol. 27, No. 1. pp. 3-18, 2004.
- [4] H. J. van den Herik; J. W. H. M. Uiterwijk; J. van Rijswijk. "Games solved: Now and in the future". Artificial Intelligence 134 (1-2): 277-311, 2002.
- [5] Kuang-che Wu, Shun-Chin Hsu, Tsan-sheng Hsu. "The Graph-History Interaction Problem in Chinese Chess". Advances in Computer Games, H.J. van den Herik et al. (eds.), pp. 165-179, Vol. 4250 of Lecture Notes in Computer Science. Springer-Verlag, Berlin. ISBN 978-3-.540-48887-3, 2006.
- [6] Michael Buro. "Toward Opening Book Learning". ICGA Journal, Vol.22, No.2, pp.98-102, 1999.
- [7] Chrilly Donninger, Ulf Lorenz. "Innovative Opening-Book Handling". Advances in Computer Games, ACG 11, H.J. van den Herik et al.(eds.), pp. 1-10, Vol. 4250 of Lecture Notes in Computer Science. Springer-Verlag, Berlin. ISBN 978-3-.540-78887-3, 2006.
- [8] Shi-Jim Yen, Tai-Ning Yang, Jr-Chang Chan, Shun-Chin Hsu." Study on Loop Problem in Opening Database". New Trends in Applied Artificial Intelligence, H.G. Okuno and M. Ali(eds.), pp. 1118-1126, Vol. 4570 of Lecture Notes in Computer Science. Springer-Verlag, Berlin. ISBN 978-3-.540-73322-5, 2007.
- [9] 許舜欽，「電腦對局的搜尋技巧」，台灣大學工程學刊，第五十一期，第 17-31 頁，1991 年。
- [10] 許舜欽，「電腦西洋棋和電腦象棋的回顧與前瞻」，電腦學刊，第二卷，第二期，第 1-8 頁，1990 年。
- [11] 許舜欽，「電腦象棋的設計與製作」，電腦學刊，第二卷，第四期，第 1-11 頁。
- [12] 許舜欽，林益興，「電腦象棋的盲點解析」，電腦學刊，第三卷，第四期，第 1-6 頁，1991。

- [13] 許舜欽，陳志昌，李天佑，「網路應用於電腦象棋學習系統之設計與製作」，高速計算世界，第七卷，第四期，pp.12-16，1999。
- [14] 徐心和，王驕，「中國象棋計算機博弈關鍵技術分析」，小型微型計算機系統，第二十七卷，第六期，pp.961-969，2006。
- [15] Bo-Nian Chan, Pangfang Liu, Shun-Chin Hsu, Tsan-sheng Hsu. “Abstracting Knowledge from Annotated Chinese-Chess Game Records”. Computers and Games Conference 2006.
- [16] I.-C. Wu and D.-Y. Huang, “A New Family of K-in-a-row Games,” Advances in Computer Games Conference (ACG2005), Taipei, Taiwan, 2005.
- [17] Haw-ren Fang, Tsan-sheng Hsu, and Shun-chin Hsu . “Construction of Chinese Chess Endgame Databases by Retrograde Analysis”. Computers and Games, Second International Conference, CG 2000 (eds. T.A. Marsland and I. Frank), pp. 96-114, Vol. 2063 of Lecture Notes in Computer Search. Springer-Verlag, Berlin. ISBN 3-540-43080-6, 2001.
- [18] 陳志昌，「電腦象棋知識庫系統之研製」，國立台灣大學，博士論文，民國 94 年。
- [19] 高暉倫，「電腦象棋審局評分自動調整系統」，國立交通大學，碩士論文，民國 98 年。
- [20] 許俊彬，「象棋棋形辨識之研究」，國立交通大學，碩士論文，民國 95 年。
- [21] 鄭明政，「電腦象棋程式位置評分表之研究」，國立成功大學，碩士論文，民國 95。
- [22] 陳漢鴻，「電腦象棋的自我學習」，國立雲林科技大學，碩士論文，民國 95。
- [23] 蔡旭程，「高效率電腦象棋停著殺程式之設計與製作」，台灣大學資訊工程學研究所，碩士論文，民國 94。
- [24] 曾汶傑，「象棋殘局庫之研究」. 交通大學資訊科學與工程研究所，碩士論文，民國 97。
- [25] 曹國明，「智慧型中國象棋程式的設計與製作」，台灣大學資訊工程學研究所，碩士論文，民國 77。
- [26] WIKIPEDIA, Chinese Chess, available at http://en.wikipedia.org/wiki/Chinese_Chess
- [27] WIKIPEDIA, 中國象棋開局編號, available at <http://zh.wikipedia.org/wiki/>
- [28] 亞洲象棋聯合會裁判組 (2003). 象棋比賽規則
- [29] MREM, Media Relations & Event Management, available at

<http://mrem.bernama.com/viewsm.php?idm=1768>.

[30] WIKIPEDIA , Ishikawa diagram, available at
http://en.wikipedia.org/wiki/Fishbone_diagram,

