

從圍棋棋譜擷取棋形之探勘技術

許家平

國立東華大學資訊工程所碩士生
m9621002@em96.ndhu.edu.tw

顏士淨

國立東華大學資訊工程系副教授
sjyen@mail.ndhu.edu.tw

摘要

在棋局雙方進行短兵相接的攻防中，好的著手或定式是值得讓初學者效仿的，亦可做為電腦圍棋比對棋形判斷下一手的依據。以職業棋士來說，在他們的腦中至少有二十萬種以上棋形的知識。這種知識累積愈多，對於棋力的提昇就有愈大的幫助。在本論文中我們採用5x5 固定大小的棋形樣式作為儲存單位，藉以建立一個GoPattern資料庫。這個資料庫可做為電腦圍棋程式的局部棋形應對判斷來源，有助於電腦圍棋近戰攻防做出快速又準確的著手。

關鍵詞：圍棋、電腦圍棋、棋型比對

一、緒言

(一)棋形的概念

一個棋形(Pattern)是幾顆棋子在棋盤上之局部區域所形成的特殊形態，它指示當此棋形出現時，何處是要點。而廣義的棋形除了包括前述之特殊形態外，也包括棋形本身的意義和如何運用此情形。棋形所代表的意義是專家知識的累積，記住已出現過的棋形及其意義何在，一方面可幫助棋手減少搜尋的時間，另一方面也可幫助選擇出好的著手。事實上人類棋手對於圍棋之所以能下的好的原因是因為人有非常優秀的棋形比對能力。一個好的圍棋棋手可以很快地在實戰或從圍棋書籍學習

並累積新的棋形，快速地辨認出棋形，清楚地知道每一個棋形的目的，而靈活運用每個棋形。[3]

圍棋能力的高低與所記住的棋形的多寡有著絕對的關係，據心理學家估計，圍棋大國手林海峰九段所記住的棋形可達二十多萬種，而一般業餘高段棋士所記住的棋形也在一萬種以上[3]

例如圖 1 是一個在實戰中常出現的基本棋形，這時候輪到黑棋，以正常情況來說，在這裡黑方落點座標 C3 是一著好棋，其它的著手都不好。這對於有圍棋概念的人而言，可以說是一種棋形辨識；以專業棋手來說，在他們腦海中至少有成千上萬種棋形的知識。這種棋形懂得愈多，對於棋力的提昇就有很大的幫助。或者也可以說，想要下好圍棋，就一定要多研究棋形的相關知識，例如定石(Joseki)的學習。

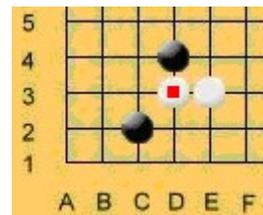
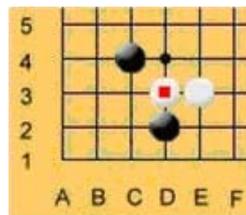


圖 1 棋形一 圖 2 棋形二

(二)棋形的表示方法

資料庫中每一個棋形分別以一個空點為中心，代表此棋形所指示之棋步，並且每一個棋形皆附有一組參數，紀錄如何使用和評估這個棋形。

棋形資料庫的設計方法有很多種 [5][6][7][8][9]，這些文獻大都是在探討如何能用最快的方式辨識棋形。一般棋形之表示方式可分為固定性大小及不固定大小兩種方式。固定大小的好處是儲存較方

便，且辨識速度較快。[3]

一般自行設計的棋形，為了讓電腦能夠輕易讀取，通常都是以符號、字串或者數字型態來表示黑子、白子、空點、無關點(don't care point)…等盤面資訊，如國立台灣大學資訊工程研究所博士「嚴初麒」論文”棋形知識庫之設計與製作”對於 5x5 格式大小棋形的表示範例如下：

```
 _ _ _ _ _ 0 代表我方棋子
_ X 0 _ _ X 代表敵方棋子
_ . . X _ . 代表空點  _ 代表無關點
_ . @ @ _ & 代表不能是我方棋子
_ _ _ _ _ @ 代表不能是敵方棋子
```

每個符號再另外使用數字編號做儲存[4]

也許這樣子的棋形表示式對人類較難以理解，但相較於電腦則是簡單又明確的辨別方式，因為電腦無法眼見所謂的視覺影像，但人的右腦主宰空間圖形的辨識，可輕易辨別出圖 1 與圖 2 的不同，但電腦只能讀 ASCII Code 碼，因此為了方便表達且能讓電腦看的懂棋形，本篇論文採用符號和字串的棋形表示式。

(三)棋譜資料庫與棋形知識庫的差異性

以往電腦對局序盤都會使用到開局資料庫，儲存的資料幾乎都是專業棋手所下的棋譜，蒐集的目的無非就是逐一比對棋局盤面第一手至第 N 手是否與棋譜資料庫某一筆相符合。最大優點是其能夠模仿高段棋士的佈局，既準確又快速，但是隨著棋局的變化與手數的增加，會發覺所需要的棋譜資料量是倍增的，而且假設局面一開始就進入攻防戰，那麼此類棋局則無法繼續採用開局棋譜資料庫去一一做比對。

然而改採用局部的棋形做比對，不僅可以解決開局資料庫所無法面對的攻防戰問題，以 5x5 格式大小的棋形而言，在九路圍棋應用上不失為另一種的局部性開局資料庫，預估當知識庫棋形類型累積至十萬種以上的時候，大約等同於一個業餘棋士記憶的棋形量，多半可以解決前 10 手的可能著手變化。

(四)5x5 與 NxN 的迷思

要將棋形樣式透過電腦程式將它設計出來，首先要決定每個棋形樣式的大小，為了製作上的便利，棋形樣式的大小，最好是固定 nxn 的型式[4]。那很多人會有疑問，為何非要採用 5x5 不可，怎不使用 3x3 或 7x7 甚至 9x9。單就 3x3 大小來說，使用上的功用，主要是讓電腦能夠處理簡單的棋形，最常辨識的有眼位(圖 3)、鏈結點(圖 4)等應用，此類的棋形資料量較少，可讓電腦快速掃描更新盤面資訊之用。

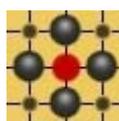


圖 3 眼位

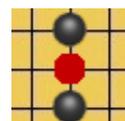


圖 4 鏈結點

至於不考量使用 7x7 或 9x9 原因，則是因為格式越大會影響搜尋比對的速度，以及比對到相同棋形的機率性相對也較低，實用性不比 5x5 來的好，況且絕大多數的圍棋手段，例如長、扳、跳、飛、衝、斷等等，都能夠包含在 5x5 的範圍中，所以就以 5x5 大小已足以適用於一般的常用棋形。而本篇論文提出的 5x5 棋形表示式將不同於[4]所定義之方法。

(五)研究目的

一般電腦對局除了開局採用資料庫之外，另外有用到資料庫的就是棋形的比對了。目前使用棋形最多的程式是曾經拿過世界冠軍的 GNU Go，對奕的過程思考時間相當之快。

要是電腦對局過程中皆用模擬的方式去思考，將會花用太多時間，又以 19 路棋盤為最，一手棋可能耗費數十秒至一分多鐘不等，對於正規比賽(9 路 30 分鐘，19 路 60 分鐘)，每一手思考太久的話，容易超時。

因此本篇研究目的是希望透過棋形的探勘採樣，讓程式可以快速比對並找到合法且有意義的著手。一但盤面資訊經過棋型資料庫比對後，搜尋結果如果沒有符合的棋形資料，此時才輪到用模擬的方式落子。然而目前最熱門且最有效的模擬方式為 Monte-Carlo Tree Search。

二、Pattern 擷取技巧

(一) 如何從棋譜採樣

如果說一個九路棋譜有五十幾手，並非所有的每一手都是可以採集的樣本，因為所要擷取出來的棋形大小為 5x5，因此必須考慮到前後手的相對應位置，皆必須同存在於範圍內才算有用的樣本，例如從圖 5 到圖 7 的變化，則是典型的問應手，因此有用的樣本端看後面一手的位置是否與前一手的位置同在於 5x5 範圍內。

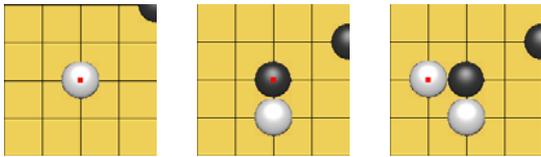


圖 5 採樣一 圖 6 採樣二 圖 7 採樣三

(二) 對採集的樣本加上認證

何以要對採集的樣本加上認證？因為 5x5 格式大小的棋形，如果棋形表示僅用黑子、白子、空點表示的話，複雜度則是 3 的 25 次方，雖然看似很大，但盤面倘若很空的話，像是僅有少數的一兩顆棋子在範圍內如圖 5，擷取棋譜時候難免不會遇到相同的樣本，這種情況最容易發生在佈局序盤的時候，此乃擷取棋形最容易發生的狀況一；而且單就以往[4]在定義 5x5 格式棋形方法來看，無法顧及到 5x5 範圍之外的任何子棋對範圍內的影響力，例如圖 8 的 B 點是高段棋士運用厚勢去攻擊白 A 的範例，擷取樣本則為框起來的範圍，同樣的取樣如果為圖 9 的話，B 點則不太可能會是高段棋士所下的位置，也許落在其它位置，像這種擷取的棋形雖為相同，但是下一手的着手卻大不相同，也就是擷取棋形範圍外的子棋會影響範圍內着手的不一樣，此為狀況二。

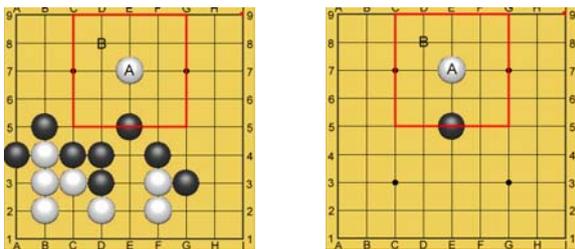


圖 8 棋形附帶外勢 圖 9 棋形無外勢

所以本篇論文考慮以上兩點狀況加以改良，利用對棋形的加權勢力分析[1]，宛如對棋形加上認證，確保同一個棋形，在 5x5 周圍外有著不同的棋子分布的情況下，有不一樣的定義方式。

以九路棋譜範例來說，圖 10、圖 12、圖 14 分別為 1968 年 吳清源 vs 宮本直毅 的棋譜第二、三、四手，而圖 11、圖 13、圖 15 則分別為加權勢力變化圖，分析方法可察看[1]。

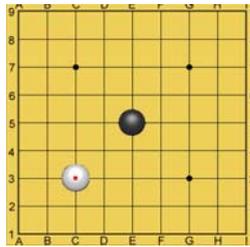


圖 10 圖 5 之原圖

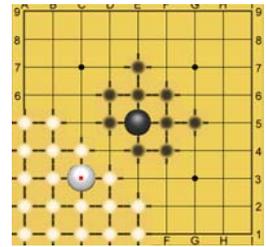


圖 11 對圖 10 加權

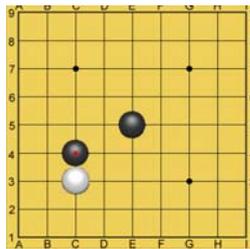


圖 12 圖 6 之原圖

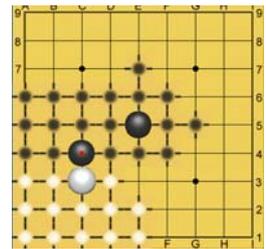


圖 13 對圖 12 加權

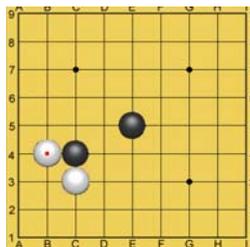


圖 14 圖 7 之原圖

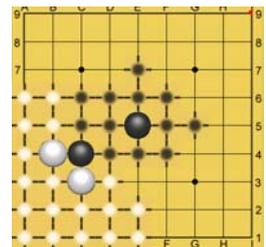


圖 15 對圖 14 加權

(三) 5x5GoPattern 建構模式

搜尋高手棋譜從第一手到最後一手，依據當前敵方所下之子(如圖 16 中棋標所在位置 D3)的區塊當中，擷取 5x5 的格式大小製作棋形，如果棋譜的下一手為我方著手落在此 5x5 區塊當中(圖 16 中“#”字位置 D2)，即為所需要儲存的 5x5 格式之棋形，但當下一手棋譜著手超出此 5x5 範圍則不列入所需棋形考量。

探勘多個棋譜過後，如果遇到相同盤面，擷取出的棋形在資料庫中已有的話，則不重複儲存。

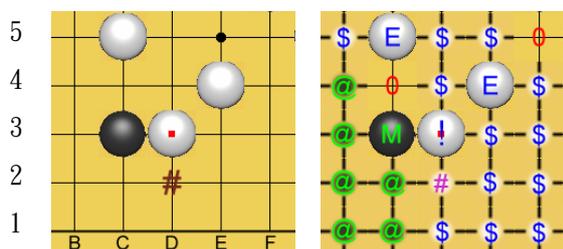
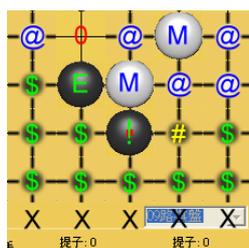


圖 16 棋形原圖 圖 17 棋形定義圖

圖 17 為 5x5 格式之 GoPattern 定義方式：“！”代表目前敵方所落子的位置，“#”是下一手預測我方應對的著手位置，“0”代表空點(非雙方勢力點)，“M”代表我方棋子，“E”代表敵方棋子，“@”代表空點(我方勢力點)，“\$”代表空點(敵方勢力點)。



當 5x5 格式之棋形超出盤面所容許的範圍(19 路為 A1~S19，9 路為 A1~I9)，則超出的部份以“X”表示。

圖 18 擷取 5x5 棋形超出棋盤範圍

(四)儲存棋形的讀取方向

每個棋形與實際的盤面作棋形辨識比對時，除了原形之外，通常需要再作 90 度、180 度、270 度等共 4 種角度的旋轉；另外每種角度還要作鏡射(對稱)處理，因此總共需要比對 8 次[4]。

從盤面儲存棋形之讀取方向共有八種方位，意指一個原始棋形樣式將會產生八筆 Pattern 資料。

- 1：先 Down 往 Up，再 Left 往 Right。
- 2：先 Up 往 Down，再 Left 往 Right。
- 3：先 Down 往 Up，再 Right 往 Left。
- 4：先 Up 往 Down，再 Right 往 Left。

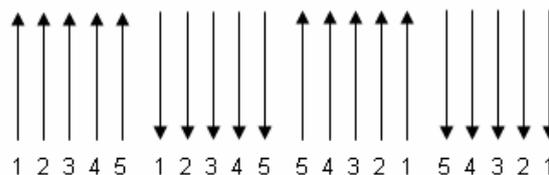


圖 19 擷取棋形讀取方向前四種

- 5：先 Left 往 Right，再 Down 往 Up。
- 6：先 Left 往 Right，再 Up 往 Down。
- 7：先 Right 往 Left，再 Down 往 Up。
- 8：先 Right 往 Left，再 Up 往 Down。

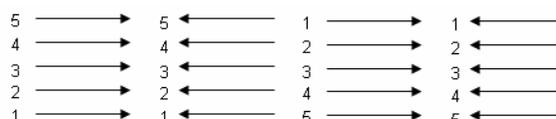


圖 20 擷取棋形讀取方向後四種

以下為一組棋形樣式經過程式讀取並儲存至資料庫內的同類棋型之八筆資料範例，圖 21 為其採樣的原始圖，圖中的“#”為下一手著手位置，圖 22 為探勘後的棋形表示式。

- 1：00\$000\$\$\$0\$\$!\$\$0\$\$\$000#00
- 2：00#000\$\$\$0\$\$!\$\$0\$\$\$000\$00
- 3：00\$000\$\$\$0\$\$!\$\$0\$\$\$000#00
- 4：00#000\$\$\$0\$\$!\$\$0\$\$\$000\$00
- 5：00\$000\$\$\$0\$\$!\$#0\$\$\$000\$00
- 6：00\$000\$\$\$0#\$\$!\$\$0\$\$\$000\$00
- 7：00\$000\$\$\$0\$\$!\$#0\$\$\$000\$00
- 8：00\$000\$\$\$0#\$\$!\$\$0\$\$\$000\$00

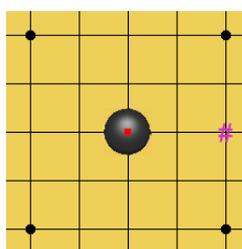


圖 21 原始圖形

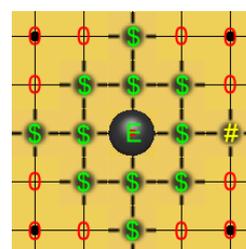


圖 22 GoPattern

三、實作測試

使用的資料庫為 Microsoft SQL Server 2005，程式則是自行開發的 CPS Go

電腦圍棋程式[1]，作業系統為 Windows XP SP2，硬體設施 CPU 為 Inter Pentium 4 3.0GHz，RAM 為 2.00GB。



圖 23 CPS Go 電腦圍棋程式

目前測試以九路圍棋為主要優先，收集的九路棋譜為日本高段職業棋士之比賽棋譜，年限最早從 1968 的吳清源開始到 2007 年的日本理光杯男女混雙賽棋譜，截至目前為止，一共探勘了 660 個棋譜，採集了 13097 個棋形樣式，乘以 8 等於 104776 筆資料。

ID	No	Name	Pattern
104752	13094	2007年_日本理...	\$#0M@#EEM\$#...
104753	13095	2007年_日本理...	@MEE\$#@#MEE...
104754	13095	2007年_日本理...	EE0M@MOE0M...
104755	13095	2007年_日本理...	\$EEM@EEM#@...
104756	13095	2007年_日本理...	@MOEEMOEOE...
104757	13095	2007年_日本理...	@@MMEM#00E...
104758	13095	2007年_日本理...	EMM@#E00#M...
104759	13095	2007年_日本理...	\$EMM@EEMOM...
104760	13095	2007年_日本理...	@MME\$MOME...
104761	13096	2007年_日本理...	XXXXX@#\$@#...
104762	13096	2007年_日本理...	MEEMEE\$E\$#...
104763	13096	2007年_日本理...	XXXXX@#\$@#...
104764	13096	2007年_日本理...	EMEE\$E\$E0...
104765	13096	2007年_日本理...	X@#EMX\$EEX...
104766	13096	2007年_日本理...	ME#@XEE\$E\$E...
104767	13096	2007年_日本理...	X@0\$EX@MEM...
104768	13096	2007年_日本理...	E\$0@XMEM@X...
104769	13097	2007年_日本理...	XXXXX\$E\$@#...
104770	13097	2007年_日本理...	\$EEMM\$EEM@...
104771	13097	2007年_日本理...	XXXXX@#\$E\$M...
104772	13097	2007年_日本理...	MMEE\$@MME\$...
104773	13097	2007年_日本理...	X\$E\$E\$E\$E\$E\$1...
104774	13097	2007年_日本理...	\$E\$E\$E\$E\$E\$EM...
104775	13097	2007年_日本理...	X@M@MX@#M...
104776	13097	2007年_日本理...	M@M@XMM#@...

圖 24 GoPatternDatabase 之 NinePattern

搜尋的方式為程式剛開啟時便載入 GoPattern 資料庫內所有棋形暫存於記憶體，以 104776 筆棋形資料，所消耗記憶體大小約為 385MB。實戰測試的過程，每下一手比對一次，比對模式首先經由掃描盤面資訊一次之後(分析棋串、棋塊、雙方勢力點)，根據當前敵方所落的位置，擷取範圍上下左右+2 的加權勢力分析過後的棋形(亦即 5x5 格式棋形)，再與所有棋形資料比對過一遍，過程中如遇單筆資料其中有一個字元與棋形 25 個字元任何一個不符合則跳過，依序比對到最後一筆棋形結束，一次比對完所花費的時間 1 秒內可完成。

測試的結果，假設棋局開始序盤不低於 3 路，如以正常的開局手法來論，運用專家的佈局思維落子，逐一搜尋比對到的棋形符合率，第一手固然是 100%，但就前 10 手而言，將近 3 成至 5 成機率會比對到棋形，其中又以第一手下天元以後比對到的機率為最高，爾後甚至到二十至四十手以後也都還會比對到相同棋形，只不過機率顯得相當微小，約為 1 至 3 成左右。

研究發現，序盤大概前 10 手內，最常出現同一個棋形如圖 21，比對之後的結果，卻有好幾個著手產生，一但遇到這種情況時，改善的方法有兩種：第一、程式將挑選出最有利於增加愈多己方勢力點的候選棋步為著手，也就是說以得到愈多實地的著手為優先考量，但也因此讓比對所花時間多增加了兩秒。第二、棋譜探勘的過程中紀錄每一個棋形重複的次數，當遇到上述情況發生時，以重複次數最多的棋形為優先考量，如此省去第一種方式評估的時間。

四、結論

(一) 5x5GoPattern 優缺點

優點 1. 網路上收集高手棋譜很簡易，並可利用電腦大量擷取出可用的棋形。

優點 2. 比對棋形資料庫相當快速，有助於增加電腦對局落子速度。

優點 3. 有利於 5x5 區域內近戰攻防產生有效率的應對手。

優點 4. 不同於一般 5x5 格式，引入加權勢力分析(對棋形加上認證)，有助於格式外的子棋亦可影響格式內的棋形變化，增加棋形複雜度，也就是 5 個變數，25 個位置減去 2 個固定變數(!和#)，且一個棋形可旋轉+鏡射共有 8 個方位，最高可到達 $5^{23/8}$ 種變化棋形。

缺點 1. 當棋形<5x5 格式的時候容易誤判(當擷取超出棋盤範圍，如圖 18)，當敵方落子在第一線則擷取出的棋形格式有**意義區域(非 X)**縮減為 3x3 or 3x4 or 3x5，其中又以 3x3 為最差，程式比對時將不列入棋步考量。

缺點 2. 當棋形出現爭劫狀況，無法找出劫材，需額外判斷進入打劫系統。

缺點 3. 當資料庫棋形筆數愈多，程式載入時所須記憶體容量相對增加許多。

(二) 日後發展

往後將會以 19 路棋譜為探勘對象，假設都沒有出現重複棋形的話，目前初估一個 19 路棋譜平均可以採集到一百多個近戰棋形(9 路平均採樣為 25~30 個)。

五、誌謝

感謝東華大學資訊工程系副教授 顏士淨的教導和輔仁大學資訊工程所學長周政緯的指正，以及東華大學資訊工程所同學 蘇祭程的協助。

六、參考文獻

- [1] 許家平、陳擎文，”電腦圍棋製作與棋型勢力分析之研究”，全國數位內容學術研討會，立德管理學院，2006 年。
- [2] 許家平、陳擎文，”電腦圍棋棋串表示法及其應用”，第 11 屆人工智慧與應用研討會，高雄應用科技大學，2006 年，12 月。
- [3] 顏士淨，“電腦圍棋程式 Jimmy5.0 之設計與製作”，博士論文，pp. 46-50，1999 年 6 月
- [4] 顏士淨、嚴初麒，“棋形知識庫之設計與製作”，全國計算機會議(NCS2003)，逢甲大學，2003 年。
- [5] 嚴初麒，“電腦圍棋 Archmage 1.0 之設計與製作”，國立台灣大學資訊工程研究所碩士論文，1992 年。
- [6] K. Takuya, N. Saburo, and U. Kazuhiro, “A case study on acquisition of pattern knowledge in Go using ecological analogy”, Game programming workshop in Japan, pages133—139,1996。
- [7] L. Richard J., “Pattern matching in a Go Playing Program”, Game programming workshop in Japan, pages 167-174, 1995
- [8] M. Martin, “Computer Go as a Sum of Local Games:An Application of Combinatorial Game Theory”, Ph.D. Dissertation, Swiss Federal Institute of Technology Zurich, 1995.
- [9] R. Walter and W. Bruce, “Pattern recognition and pattern-directed inference in a program for playing Go.”,Pattern-Directed Inference Systems, pages 503-523, 1978