

電腦對局理論

電腦暗棋 Report

B01902032 江東峻

一、審局函數設計

動態子力價值

由於將每個棋子的價值固定，會造成資料上判斷錯誤。例如只剩下5個兵、5個卒、1個帥的情況，那這個帥的價值就非常低，因為他什麼也吃不到。所以一個棋子的價值應該取決於它還能吃到的棋子價值，例如盤面上有1個紅馬、1個黑炮、1個黑車、1個紅帥，那黑炮的價值在於只有他能夠吃帥，而帥又能吃目前黑方所有的棋子，紅馬的價值在於他可以吃黑炮，若把紅馬換成紅兵，因為兵不能吃目前場上任何的棋子，所以他的價值為0，但是當吃掉紅帥以後，必須也把紅兵吃掉才算贏，所以若是價值0的話，那就會變成吃不到的狀況，所以最後所有棋子都必須給一個基本分（設為50）。

子力算法採用謝政孝在<暗棋中棋種間食物鏈關係之探討與實作>裡的方法：

（一）計算所有棋子的基本價值：

1. 每顆棋子基本價值為 $1+4\times(\text{無法反擊的子數})+1\times(\text{可以反擊的子數})$ 。

2. 炮(包)基本價值則為 $4\times(\text{我方棋子數})+1\times(\text{對方棋子數})$ 。

（二）每個棋子的子力則為能吃到的所有棋子基本價值總和。

（三）根據每個棋子棋種乘上它的棋種權重。

（四）最後加上基本分50。

距離分數

但是單純算棋子的價值不足以代表整個棋面的價值，例如：1個黑馬、1個紅馬分別間隔1格、2格，若是黑方先下，很明顯若是間隔1格時，黑方有利（因為下一步就可以吃掉紅馬）。所以棋面必須將距離因素考慮進去。這裡採用的方法是計算1個棋子對於他所能吃到的棋子的距離（需要花幾步）算入加分：

對於所有 A 能吃到的棋子：

$\text{分數} + = \text{能吃到棋子的價值} / \text{距離} * \text{距離權重}$

距離的算法是：

對於路線中A不能吃的棋子，視為障礙物，找到一條路可以以最小步數吃到目標棋子。

而在暗棋中，棋子和棋子之間常常被覆蓋的棋子所區隔，所以當棋子A無法走到棋子B時，A的距離分數就不用考慮棋子B，而能不能走到這件事情可以用 **flooding** 演算法解決。

而距離權重的部分是為了讓吃子行為不會受到距離分數太大的影響，例如：吃了一個卒以後，若他周遭的棋子對他的距離分數總和太高，反而會造成吃了之後分數下降，因此選擇不吃的狀況。所以加入一個距離權重限制距離分數的影響力。

位置分數

因為有可能出現子力價值和距離分數都無法判斷的情況，例如：場面上只剩下1個黑馬、1個紅馬，這時候若距離為9，則有可能為5(行) + 4(列)，或者4(行) + 5(列)兩種情況，這時候需要加入一個位置分數，讓棋盤有起伏，以便在這種情況作出較好的判斷。

位置分數參考了賴學誠的論文<電腦暗棋程式與經驗法則配合之實作>：在邊與角的位置自由度較差，分數較低，中間的位置自由度較高，分數較高。而位置分數比起距離分數以及子力價值來說是非常微小，只有在分數相當的時候，位置分數才會有影響力。

二、搜尋樹

Alpha-Beta pruning 演算法

因為助教sample code是max-min tree，為了方便修改，採用的是課堂投影片裡的F2'以及G2'，做限制深度的搜尋(深度為7)，因為若是限制時間，會比較不穩定(有可能這次搜尋深度夠，在棋子數多的時候反而深度變淺，但合法步數多的時候會更需要較精準的搜尋，所以深度應該要比合法步數少的深)。

perfect-ordering

若node的排列方式越接近perfect-ordering，則砍樹越有效率，所以在每次搜尋時，會先根據審局函數將合法步做一次排序。

空步搜尋

由於除了明棋的步(包含移動及吃子)，還有可能出現翻棋，所以在Alpha-Beta pruning時，加入翻棋的可能性：

1. 呼叫Alpha-Beta pruning前，先計算目前覆蓋的棋子數A並作為參數傳入Alpha-Beta pruning function。
2. 呼叫Alpha-Beta pruning，若 $A > 0$ ，則加入翻棋的可能。
3. 若現在走的是翻棋的步，則呼叫下一層時，傳入 $A-1$ 。

加入空步搜尋能有效提高搜尋的正確性，例如：當1個黑馬被1個紅仕關在角落時，若沒有做空步搜尋，那黑馬怎麼算都會死，若做空步搜尋，則有翻牌的可能性。(當然，在沒有覆蓋的牌情況下還是只能送死。)

深度加權

Alpha-Beta pruning常常會在不一樣的合法步得出一樣的分數，因為它無法判斷是走了幾步達到這個分數，例如：紅仕貼著黑馬，有可能直接吃掉，或者繞了一大圈再吃掉，但分數都相同。所以需要考慮深度，在分數相同的情況下，選擇深度較淺的步走。方法是每回傳一次，分數都要-1，所以回傳越多次(從深度越深的地方傳回來)分數就越低，而原本分數相同的步就會因為回傳次數被區分開來(由於只做深度為7的搜尋，所以原本分數相同的最多只會差到7，影響不會太大。)

三、翻棋策略

(一) 避免翻在敵方的炮打的到的位置

因為若是自己方的棋子，一翻開就可能被炮打死，所以儘量不翻。

(二) 避免翻在敵方的士旁邊

因為若是自己方的棋子，士的旁邊能存活的只有將帥，機率太低，不翻。

(三) 能不翻儘量不翻

翻棋充滿太多未知數，就算是大好的局勢依然有可能因為翻棋運氣不好而落敗，所以只有在不得不翻的時候才翻。

四、比賽心得

1. 因為不太翻棋，若一開始翻的棋子若沒有造成優勢，容易無法挽回頹勢。
2. 翻棋應該要考慮覆蓋棋子的可能性，例如：若覆蓋的棋子只有一個，且帥還沒翻出來，那士旁邊就可以翻開覆蓋的帥，增加優勢。
3. 因為使用動態子力價值，審局函數的棋子權重很重要，沒調好很容易在關鍵時刻吃到不對的棋子。
4. 對於一些刻意要和棋的策略，應該加入一些可以包夾的手段。
5. 因為比賽時只有30步的NoEatFlipper，所以在終局要收尾時必須很有效率。
6. 由於第5點，在絕對劣勢的時候，應該儘量拖時間及步數。
7. 因為使用動態子力價值，必要時要送一些子，讓吃子的優先順序作變動。

五、參考

謝政孝 暗棋中棋種間食物鏈關係之探討與實作
賴學誠 電腦暗棋程式與經驗法則配合之實作
勞永祥 電腦暗棋之人工智慧改良