

15-puzzle Searching

00857036楊育斌

程式資料架構

State為int[side * side]結構，還有其他DataNode、PriQueNode結構 分別為search tree 與 priority queue(heap)所用的節點。DataNode含有State及其加權值與父節點，供A* search所用。PriQueNode定義DataNode的優先權，供A* search的pqueue所用。皆以指標記錄。而屬於State的函式都拉出來寫成全域函式，沒寫成類別而已。

搜尋法

IDS: 使用遞迴代替stack實作，由於沒有要求回推路徑，故不需用到DataNode，全程都以一個State操作(一步步挪)，使記憶體用量不會成長。(成長都是A* search的部分)

A* search: 使用priority queue實作

Heuristic function

額外加了兩個 out of line 與 connectivity

out of line: 有多少塊不在對應行 + 多少塊不在對應列。由於Manhattan distance能夠多區隔出 與對應行(列)差1還是2行(列)，故絕對比out of line精準。

connectivity: 有多少相鄰對 在Goal state也相鄰 的相反數。並會再與misplace混和彌補絕對位置性，因為misplace最大為8，故混和式為 $9 * connectivity + misplaced$ 。猜測由於一步最多可能會拆走三對連結(h)，等於相差三步的權值(g)，波谷較深使得在區域最佳值走更多冤枉路，所以效果較差。不過在計算量越大時(deep 16)，反而勝過純misplace，偶爾的個案還勝過Manhattan distance，難道有潛力？

相關研究結論

前三種heuristic function都是走一步可-1 heuristic值(h)，故cost值(g)設計為每走一步+1進而與h抵銷，使波形保持平整，單純地避免不可能的走法持續搜尋所有可能。有實驗過調配cost的增加量，都是+1最好，(connectivity由於平均也是-1，乘上9後cost也調為+9)。所以heuristic低的(好的)並沒有比較優先。因此若將優先權改定為 $g + h$ 相等時比h，便能同時維持波形又能同時使區域較好的走法優先搜尋，實驗下來展開節點確實明顯降低。

隨機測資

會產生的捷徑的冗餘走法: 如在4格內順時針走3圈就會回到原state 產生12步迴圈，因此當走7步時便有5步的逆時針捷徑出現，自稱12過半迴圈(下一個是30過半迴圈)。但我懶得排除，所以只避免2過半迴圈(即來回走)，丟到A*搜尋看如果goal不是在指定的deep數就重新再生一個這樣 反正很快

統計結果

----- Average expand nodes -----				
		Deep 8	Deep 12	Deep 16
Search by	IDSearch :	16352	1659335	201567824
Search by	misplace amount :	68	1171	47111
Search by	Manhattan distance :	38	223	2051
Search by	out of line :	42	392	7547
Search by	connectivity :	57	3128	15465

IDS搜尋次數依賴前幾步移動方向決定, misplace的隨deep增加成長幅度明顯較其他者大, 各個測資看下來 connectivity的和其他者的關係係數甚小(其他三者皆互為正關係, 畢竟原理相似), 沒有仔細研究其強處, 應該有更多改良空間。

僅前10個固定測資, 不跑IDS:

優先權為g + h相等時依FIFO

----- Average expand nodes -----				
		Deep 8	Deep 12	Deep 16
Search by	misplace amount :	73	1465	74715
Search by	Manhattan distance :	34	359	3256
Search by	out of line :	36	654	10904
Search by	connectivity :	45	5638	24167

優先權為g + h相等時比h

----- Average expand nodes -----				
		Deep 8	Deep 12	Deep 16
Search by	misplace amount :	50	1093	61474
Search by	Manhattan distance :	34	294	1994
Search by	out of line :	35	583	9121
Search by	connectivity :	45	6057	24081

既然要較好的走法優先, 那為什麼不抓新的呢? (因被淘汰的走法會堆在底下)

優先權為g + h相等時依LIFO

----- Average expand nodes -----				
		Deep 8	Deep 12	Deep 16
Search by	misplace amount :	45	840	46379
Search by	Manhattan distance :	30	159	1113
Search by	out of line :	34	379	5846
Search by	connectivity :	45	5944	23951