

TÌM KIẾM CÓ ĐỐI THỦ

4.1. Vấn đề chơi cờ

4.1.1 Trò chơi hai người

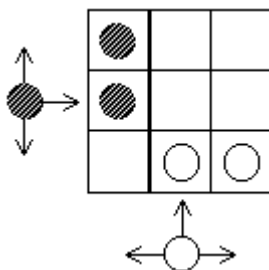
- Hai người chơi thay phiên nhau đưa ra các nước đi tuân theo các luật đi nào đó, các luật này là như nhau cho cả hai người và hai người chơi đều được biết thông tin đầy đủ về các tình thế trong trò chơi.
- Ta cần tìm kiếm nước đi tốt nhất cho máysao cho sau một số nước đi máy giành phần thắng. Việc tìm kiếm sẽ phức tạp hơn vì máy khôngbiết được người chơi sẽ đi nước nào trong tương lai.
- Giả sử máy tính cầm quân trắngTrắng,người chơi cầm quân Đen và ta gọi máy tính là trắng (hay Max), người chơi gọi là đen (hay Min).
- Vấn đề chơi cờ có thể xem như vấn đề tìm kiếm trong không giantrạng thái.
 - ✓ Mỗi trạng thái là một tình thế (sự bố trí các quân của hai bên trênbàn cờ).
 - ✓ Trạng thái ban đầu là sự sắp xếp các quân cờ của hai bên lúc bắtđầu cuộc chơi.
 - ✓ Các toán tử là các nước đi hợp lệ.
 - ✓ Các trạng thái kết thúc là các tình thế mà cuộc chơi dừng, thườngđược xác định bởi một số điều kiện dừng nào đó.
- Như vậy vấn đề của Trắng là, tìm một dãy nước đi sao cho xen kẽ vớicác nước đi của Đen tạo thành một đường đi từ trạng thái ban đầu tới trạngthái kết thúc là thắng cho Trắng.

4.1.2 Cây trò chơi

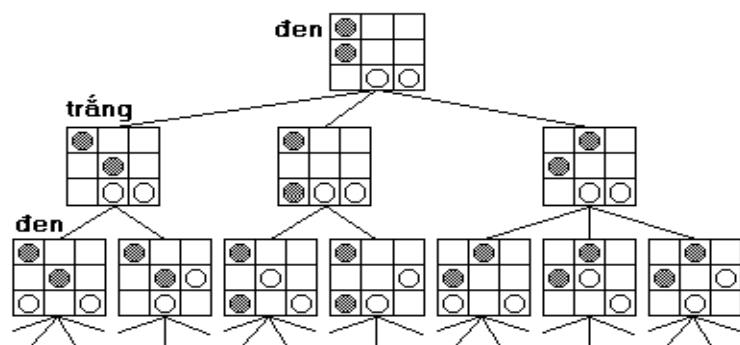
- Gọi đỉnh ứng với trạng thái mà Trắng (Đen) đưa ra nước đi làđỉnh Trắng (Đen). Cây trò chơi được xây dựng như sau.
 - ✓ Gốc của cây ứng với trạng tháiban đầu,các lá của cây ứng với các trạngthái kết thúc.
 - ✓ Nếu một đỉnh là Trắng (Đen) ứng với trạng thái u, thì các đỉnh con của nó là tất cả các đỉnh biểu diễn trạng thái v, v nhận được từ u do Trắng (Đen) thực hiện nước đi hợp lệ nào đó. Do đó, trên cùng một mức củacây các đỉnh đều là Trắng hoặc đều là Đen.
- Quá trình chơi cờ là quá trình Trắng và Đen thay phiên nhau đưa ra quyết định, thực hiện một trong số các nước đi hợp lệ. Trên cây trò chơi, quá trình đó sẽ tạo ra đường đi từ gốc tới lá.

✚ Ví dụ trò chơi Dodgem

Có hai quân Trắng và hai quân Đen, ban đầu được xếp vào bàn cờ 3*3 (hình 4.1). Quân Đen có thể đi tới ô trống ở bên phải, ở trên hoặc ở dưới. Quân Trắngcó thể đi tới trống ở bên trái, bên phải, ở trên. Quân Đen nếu ở cột ngoài cùng bên phải có thể đi ra khỏi bàn cờ, quân Trắng nếu ở hàng trên cùng có thể đi ra khỏi bàn cờ. Ai đưa hai quân của mình ra khỏi bàn cờ trước sẽ thắng, hoặc tạo ra tình thế mà đối phương không đi được cũng sẽ thắng. Giả sử Đen đi trước, ta có câytrò chơi được biểu diễn như trong hình 4.2.



Hình 4.1: Trò chơi Dodgem



Hình 4.2: Cây trò chơi Dodgem với đen đi trước

4.2 Các thuật toán tìm kiếm trên cây trò chơi

- Thuật toán MINIMAX
- Thuật toán ALPHA – BETA cutoff

4.2.1 Thuật toán MINIMAX

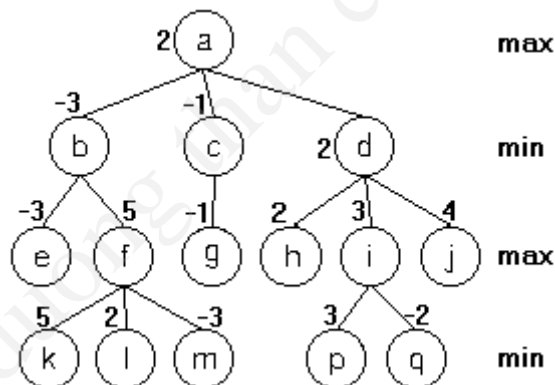
- Hàm đánh giá xây dựng sao cho đỉnh có giá trị càng lớn (nhỏ) là đỉnh càng tốt cho Trắng (Đen). Cần giả thiết là Trắng (Đen) luôn chọn nước đi kế tiếp tốt nhất cho Trắng (Đen). Khi đó chọn nước đi kế tiếp cho đỉnh Trắng (Đen) là đỉnh con có giá trị lớn nhất (nhỏ nhất) trong số các đỉnh con của Trắng (Đen).
- Thuật toán gọi là Minimax vì Trắng đã chọn nước đi ứng với đỉnh con có giá trị là max của các giá trị các đỉnh con, và Đen đáp lại bằng nước đi ứng với đỉnh con có giá trị là min của các giá trị các đỉnh con. Thuật toán Minimax chính là thuật toán tìm kiếm theo độ sâu.
- Để tính giá trị của đỉnh u ta xây dựng cây trò chơi gốc u có chiều cao d (d do ta chọn và d càng lớn thì máy chơi càng hay). Các đỉnh ở mức d hoặc các đỉnh kết thúc sẽ được tính giá trị bởi hàm đánh giá. Từ những đỉnh ở mức d đã biết giá trị, sẽ tính được giá trị của những đỉnh ở mức d-1 theo nguyên tắc đỉnh Trắng (Đen) có giá trị là giá trị lớn nhất (nhỏ nhất) trong những giá trị của các đỉnh con. Lặp lại việc tính giá trị cho các đỉnh ở mức d-2, ..., 0 đến khi tính được giá trị của đỉnh gốc u ở mức 0 thì ngừng.

🌈 Ví dụ:

Giả sử cần tính giá trị cho đỉnh a là đỉnh trắng với độ sâu 3, ta xây dựng cây trò chơi như trong hình 4.3. Khi xây dựng cây tới mức 3, ta tính giá trị cho các đỉnh k, l, m, p, q ở mức 3 bằng hàm đánh giá, giả sử ta có: $k=5, l=2, m=-3, p=3, q=-2$. Trong quá trình xây dựng cây mà gặp đỉnh kết thúc thì đánh giá luôn chứ không phát triển tiếp, ví dụ đỉnh e, g, h, j là các đỉnh kết thúc và được đánh giá $e=-3, g=-1, h=2, j=4$.

Sau khi xây dựng cây, ta lần lượt tính giá trị cho các đỉnh ở mức 2, rồi tính giá trị cho các đỉnh ở mức 1, rồi đến mức 0, như vậy đã tính được giá trị đỉnh a.

Cách tính giá trị cho các đỉnh như sau: Do đỉnh i là Trắng, nên giá trị của nó là $\max(3, -2) = 3$, đỉnh d là đỉnh Đen, nên giá trị của nó là $\min(2, 3, 4) = 2$, vv... Sau cùng tính được $a=2$.



Hình 4.3: Tính giá trị cho đỉnh a với độ sâu d=3

▪ Thuật toán MiniMax

- ✓ Hàm tính giá trị đỉnh Trắng u với độ sâu d, $f(u)$ là giá trị của hàm đánh giá đỉnh u.

```
int MaxVal(u,d){
    int val=-∞;
    if (d=0 hoặc u là đỉnh kết thúc) val=f(u);
    else {
        for(mỗi đỉnh v là đỉnh con của đỉnh u){
            int temp=MinVal(v,d-1);
            if(temp>val){
                val=temp;
            }
        }
    }
    return val;
}
```

- ✓ Hàm tính giá trị đỉnh Đen u với độ sâu d, $f(u)$ là giá trị của hàm đánh giá đỉnh u.

```

int MinVal(u,d){
    int val= $+\infty$ ;
    if (d=0 hoặc u là đỉnh kết thúc) val=f(u);
    else {
        for(mỗi đỉnh v là đỉnh con của đỉnh u){
            int temp=MaxVal(v,d-1);
            if(temp<val){
                val=temp;
            }
        }
    }
    return val;
}

```

- ✓ Hàm tìm nước đi cho đỉnh Trắng u với độ sâu d

```

void MiniMax(u, v, d){
    val =  $-\infty$ ;
    for (mỗi w là đỉnh con của u){
        int tempval=MinVal(w,d);
        if (tempval>val) { val = tempval; v = w;}
    }
}

```

- ✓ Hàm tìm nước đi v cho đỉnh Đen u với độ sâu d

```

void MiniMax(u, v, d){
    val =  $+\infty$ ;
    for (mỗi w là đỉnh con của u){
        int tempval =MaxVal(w,d);
        if (tempval<val) { val = tempval; v = w;}
    }
}

```

- Nếu hàm đánh giá, đánh giá không chính xác, nó có thể hướng dẫn máy tính đi tới trạng thái được xem là tốt, nhưng thực tế lại rất bất lợi cho máy. Thiết kế một hàm đánh giá tốt là một việc khó, đòi hỏi phải quan tâm đến nhiều yếu tố như là các quân còn lại của hai bên, sự bố trí của các quân đó,... và hàm đánh giá chính xác thường cần rất nhiều thời gian tính toán, mà trò chơi lại bị giới hạn thời gian phải đưa nước đi. Do đó ta thường thỏa hiệp là xây dựng hàm đánh giá chỉ tương đối chính xác và thực thi tương đối nhanh.

✚ Ví dụ 1: xây dựng hàm đánh giá cho cờ vua. Mỗi loại quân được gán một giá trị số phù hợp với “sức mạnh” của nó. Chẳng hạn, mỗi tốt Trắng (Đen) được cho 1 (-1), mã hoặc tượng Trắng (Đen) được cho 3 (-3), xe Trắng (Đen) được cho 5 (-5) và hậu Trắng (Đen) được cho 9 (-9). Lấy tổng giá trị của tất cả các quân trong một trạng thái, ta sẽ được giá trị đánh giá của trạng thái đó. Hàm đánh giá như thế được gọi là hàm tuyến tính có trọng số, vì nó có thể biểu diễn dưới dạng: $s_1w_1 + s_2w_2 + \dots + s_nw_n$. Trong đó, w_i là giá trị mỗi loại quân loại i, còn s_i là số quân loại i. Cách đánh giá này chưa hay vì ta đã không tính đến sự bố trí của các quân, các mối tương quan giữa chúng.

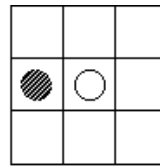
✚ Ví dụ 2: xây dựng hàm đánh giá cho trò chơi Dodgem. Mỗi quân Trắng/Đen ở một vị trí trên bàn cờ được cho một giá trị tương ứng trong bảng hình 4.4. Nếu quân Trắng cản trực tiếp một quân Đen, nó được thêm 40 điểm, nếu cản gián tiếp nó được thêm 30 điểm (hình 4.5). Tương tự, nếu quân Đen cản trực tiếp quân Trắng nó được thêm -40 điểm, còn cản gián tiếp nó được thêm -30 điểm. Trong đánh giá này, ta đã xét đến vị trí của các quân và mối tương quan giữa các quân. Áp dụng các qui tắc trên, ta tính được giá trị của hai trạng thái ở hình 4.6 là 75, và -5.

30	35	40
15	20	25
0	5	10

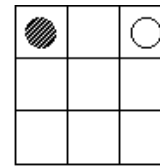
Giá trị quân Trắng.

-10	-25	-40
-5	-20	-35
0	-15	-30

Giá trị quân Đen.



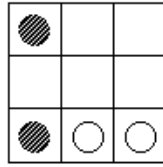
Trắng cần trực tiếp Đen được thêm 40 điểm.



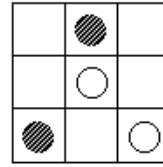
Trắng cần gián tiếp Đen được thêm 30 điểm.

Hình 4.4: Đánh giá các quân cờ trong trò chơi Dodgem.

Hình 4.5: Đánh giá sự tương quan giữa quân trắng và quân đen



75

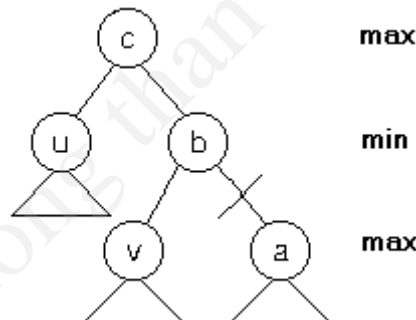


-5

Hình 4.6: giá trị của một số trạng thái trong trò chơi Dodgem

4.2.2 Thuật toán cắt tỉa Alpha-beta (Alpha-Beta Pruning)

- Trong MiniMax số đỉnh của cây trò chơi có thể rất lớn. Chẳng hạn trong cờ vua, hệ số nhánh trung bình khoảng 35, thời gian phải đưa ra nước đi là ≤ 150 giây, MiniMax chỉ có thể sử dụng độ sâu 3,4. Một người chơi cờ trình độ trung bình cũng có thể tính trước được 5, 6 nước. Do đó chương trình mới đạt trình độ người mới tập chơi!
- Khi đánh giá đỉnh u , Minimax đánh giá tất cả các đỉnh của cây gốc u , độ sâu d . Thuật toán cắt tỉa Alpha-beta cải tiến thuật toán Minimax bằng cách cắt bỏ các nhánh không cần thiết cho sự đánh giá đỉnh u .
- Giả sử tại đỉnh đen b , ta đã đánh giá được đỉnh con v (hình 4.7). Nếu $\text{val}(v) \leq \text{val}(u)$, thì do $\text{val}(b) \leq \text{val}(v) \Rightarrow \text{val}(b) \leq \text{val}(u) \Rightarrow \text{val}(c) = \text{val}(u)$; nên ta không cần đánh giá đỉnh a mà vẫn không ảnh hưởng đến việc đánh giá đỉnh c . Hay nói cách khác ta có thể cắt bỏ cây con gốc a . Lập luận tương tự cho trường hợp b là đỉnh Trắng.



Hình 4.7: cắt bỏ cây con gốc a nếu $\text{val}(u) \geq \text{val}(v)$

- Để cài đặt thuật toán cắt tỉa Alpha-Beta, mỗi đỉnh gán ba giá trị $(\alpha, \beta, \text{val})$; α là giá trị lớn nhất trong các giá trị của các đỉnh con đã đánh giá của một đỉnh Trắng, β là giá trị nhỏ nhất trong các đỉnh con đã đánh giá của một đỉnh Đen, val là giá trị của đỉnh. Thuật toán trong quá trình thực hiện sẽ cập nhật α, β tại mỗi đỉnh. Tại một đỉnh nếu có $\alpha \geq \beta$ thì loại bỏ các cây con còn lại của đỉnh này.

Thuật toán cắt tỉa Alpha-Beta

- ✓ Hàm tính giá trị đỉnh Trắng u với độ sâu d .

```
int MaxVal(u,d,  $\alpha$ ,  $\beta$ ) { //  $\alpha, \beta$  truyền theo kiểu tham trị.
    int val;
    if (d=0 hoặc u là đỉnh kết thúc) val=f(u);
    else {
        for(mỗi đỉnh v là đỉnh con của đỉnh u){
            int tempval=MinVal(v,d-1, $\alpha$ ,  $\beta$ );
            if(tempval> $\alpha$ ){
                 $\alpha$  =tempval;
            }
        }
    }
}
```

```

        if ( $\alpha \geq \beta$ ) break; // Cắt bỏ các cây con gốc v còn lại
    }
    val= $\alpha$ ;
}
return val;
}

```

- ✓ Hàm tính giá trị đỉnh Đen u với độ sâu d.

int MinVal(u,d, α , β) { // α , β truyền theo kiểu tham trị.

int val;

if (d=0 hoặc u là đỉnh kết thúc) val=f(u);

else {

for(mỗi đỉnh v là đỉnh con của đỉnh u){

int tempval=MaxVal(v,d-1, α , β);

if(tempval< β){

β =tempval;

}

if ($\alpha \geq \beta$) break; // Cắt bỏ các cây con gốc v còn lại

}

val= β ;

}

return val;

}

- ✓ Hàm tìm nước đi v cho đỉnh Trắng u với độ sâu d

void AlphaBeta (u, v, d){

$\alpha = -\infty$; $\beta = \infty$;

for (mỗi w là đỉnh con của u){

int tempval =MinVal(w,d, α , β);

if (tempval> α) { α = tempval; v = w; }

}

}

- ✓ Hàm tìm nước đi v cho đỉnh Đen u với độ sâu d

void AlphaBeta (u, v, d){

$\alpha = -\infty$; $\beta = \infty$;

for (mỗi w là đỉnh con của u){

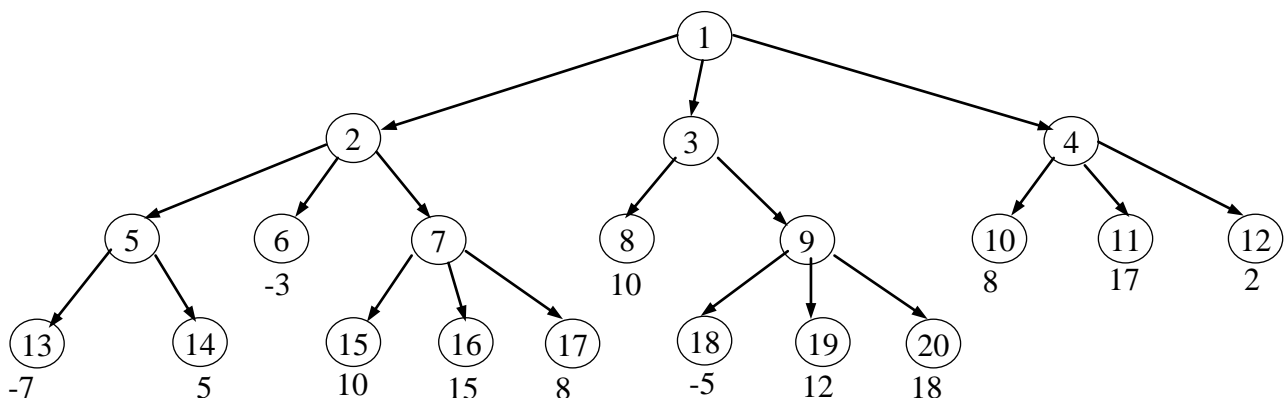
int tempval =MaxVal(w,d, α , β);

if (tempval< β) { β = tempval; v = w; }

}

}

- ✚ Ví dụ: xét cây trò chơi gốc 1 là đỉnh Trắng(hình 4.8), số ghi cạnh các đỉnh là giá trị của hàm đánh giá. Áp dụng thuật toán cắt tỉa Alpha Beta để tính giá trị đỉnh 1 với độ sâu 3.



Hình 4.8: một cây trò chơi với độ sâu 3

Đỉnh	Giá trị	α, β		α, β	α, β	Đỉnh	Giá trị	α, β
1	10	$-\infty, \infty$	-3, ∞	10, ∞		11	17	
2	-3	$-\infty, \infty$	$-\infty, 5$	$-\infty, -3$		12	2	
3	10	-3, ∞	-3, 10			13	-7	$-\infty, \infty$
4	8	10, ∞	10, 8 =>Cắt 11,12			14	5	-7, ∞
5	5	$-\infty, \infty$	-7, ∞	5, ∞		15	10	$-\infty, -3$
6	-3	$-\infty, 5$				16	15	
7	10	$-\infty, -3$	10, -3 =>Cắt 16,17			17	8	
8	10	-3, ∞				18	-5	-3, 10
9	12	-3, 10	12, 10 =>Cắt 20			19	12	-3, 10
10	8	10, ∞				20	18	

Từ bảng trên ta tính được đỉnh trắng 1 có giá trị là 10

Nếu đỉnh 1 là đỉnh đen, áp dụng thuật toán alpha betacutoff ta tính được đỉnh 1 có giá trị là 8 theo bảng sau:

Đỉnh	Giá trị	α, β	α, β	α, β	α, β	Đỉnh	Giá trị	α, β
1	8	$-\infty, \infty$	$-\infty, 8$			11	17	
2	8	$-\infty, \infty$	-7, ∞	-3, ∞	8, ∞	12	2	
3	10	$-\infty, 8$	10, 8 =>Cắt 9			13	-7	$-\infty, \infty$
4	8	$-\infty, 8$	8, 8 =>Cắt 11,12			14	5	$-\infty, -7$
5	-7	$-\infty, \infty$	$-\infty, -7$			15	10	-3, ∞
6	-3	-7, ∞				16	15	-3, 10
7	8	-3, ∞	-3, 10	-3, 8		17	8	-3, 10
8	10	$-\infty, 8$				18	-5	
9						19	12	
10	8	$-\infty, 8$				20	18	

4.3. Bài tập

1/ Cài đặt trò chơi Tic-tac-toe bằng Minimax, Cutoff Anpha_Beta.

2/ Cài đặt trò chơi Dodgem bằng Minimax, Cutoff Alpha_Beta.

- Hết -