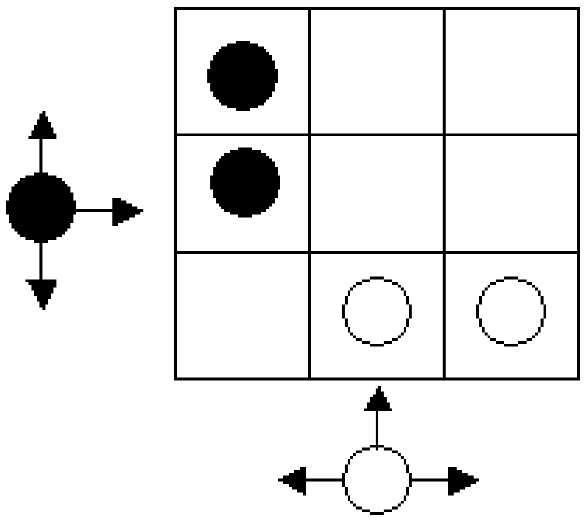
Chương 4:TÌM KIẾM CÓ ĐỐI THỦ

Ví dụ: Xét trò chơi Dodgem



Quân đen có thể đi tới ô trống bên phải, ở trên hoặc ở dưới

Quân trắng có thể đi tới ô trống bên trái, bên phải, ở trên

Quân đen nếu ở cột ngoài cùng bên phải có thể đi ra ngoài bàn cờ

Quân trắng nếu ở hàng trên cùng có thể đi ra khỏi bàn cờ

Ai đưa cả hai quân của mình ra khỏi bàn cờ trước sẽ thắng, hoặc tạo ra tình huống mà đối phương không đi được cũng sẽ thắng

Vấn đề đặt ra là nghiên cứu chiến lược chọn nước đi cho người cầm quân trắng (máy tính cầm quân trắng)

Đặc điểm của các trò chơi có hai người tham gia là:

+ 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

+ Hai người chơi đều biết được thông tin đầy đủ về các tình thế trong trò chơi

Vấn đề chơi cờ có thể xem như vấn đề tìm kiếm nước đi, tại mỗi lần đến lượt mình, người chơi phải tìm trong số rất nhiều nước đi hợp lệ một nước đi tốt nhất sao cho qua một dãy các nước đi đã thực hiện anh ta giành phần thắng.

Tổng quát vấn đề chơi cờ được phát biểu như sau:

Vấn đề chơi cờ có thể xem như vấn đề tìm kiếm trong không gian trạng thái. Mỗi trạng thái là một tình thế (sự bố trí các quân của hai bên trên bàn cờ)

+ Trạng thái ban đầu là sự sắp xếp các quân cờ của hai bên lúc bắt đầu cuộc chơi

+ Các toán tử là các nước đi hợp lệ

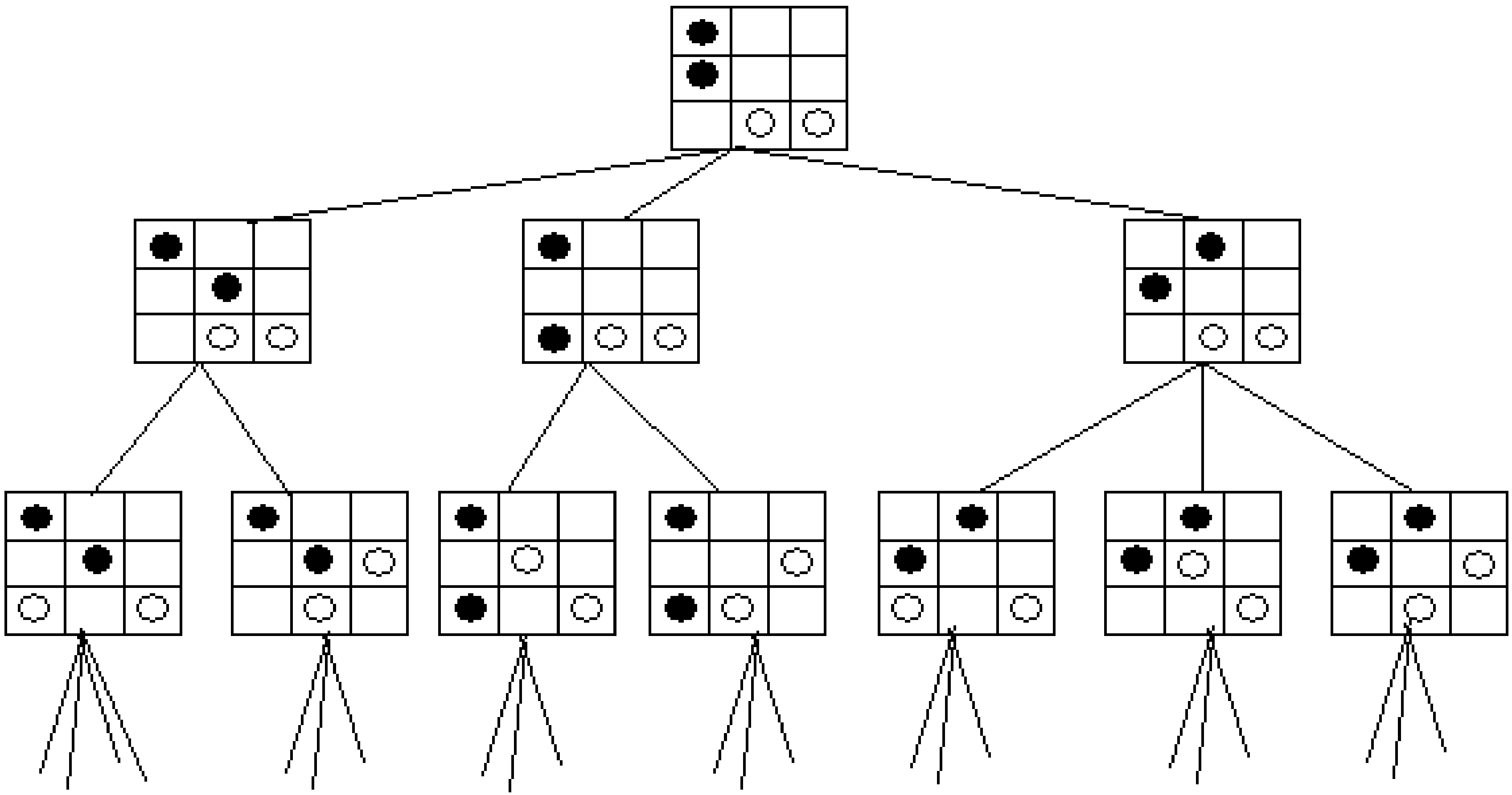
+ Các trạng thái kết thúc là các tình thế mà cuộc chơi dừng, thường được xác định bởi một số điều kiện dừng nào đó, các trạng thái kết thúc này thường được gắn với một giá trị gọi là giá trị của hàm kết cuộc

Như vậy vấn đề của trắng là tìm một dãy các nước đi xen kẽ với các nước đi của đen tạo thành một đường đi từ trạng thái ban đầu tới trạng thái kết thúc là thắng cho trắng

Để thuận lợi cho việc nghiên cứu các chiến lược chọn nước đi ta biểu diễn không gian trạng thái dưới dạng cây trò chơi

Ví dụ: Xét trò chơi Dogem như trên, ta có cây trò chơi như sau

Cây trò chơi Dodgem



Chiến lược MINIMAX

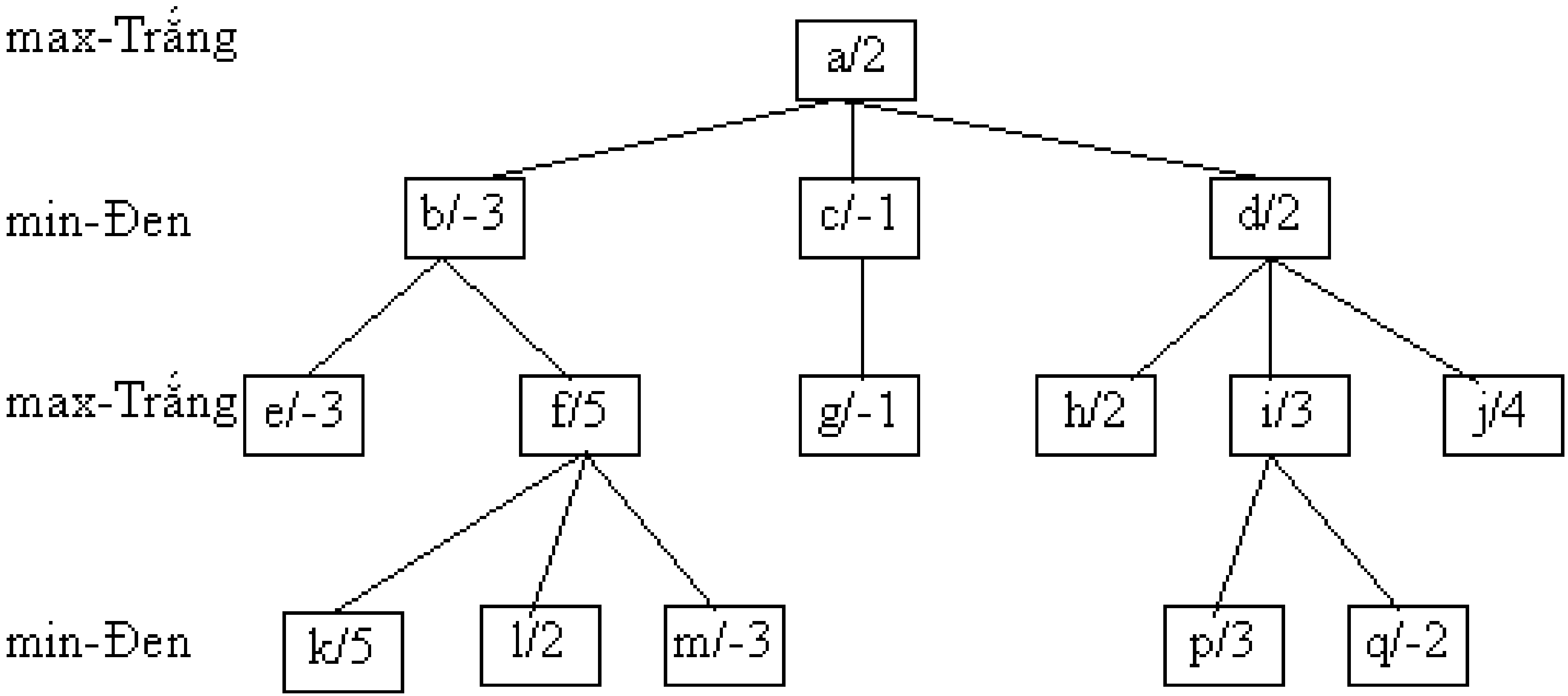
- Quá trình chơi cờ là quá trình Trắng và Đen thay phiên nhau đưa ra quyết định, thực hiện một trong số các nước đi hợp lệ. Trên cây trò chơi, quá trình đó sẽ tạo ra đường đi từ gốc tới lá
- Giả sử tới một thời điểm nào đó, đường đi đã dẫn tới đỉnh u. Nếu u là đỉnh Trắng thì Trắng cần chọn đi tới một trong các đỉnh Đen v là con u. Tại đỉnh Đen v mà Trắng vừa chọn. Đen sẽ phải chọn đi tới một trong các đỉnh trắng w là con của v.
- Quá trình trên sẽ dừng lại khi đạt tới một đỉnh là lá của cây

- Giả sử Trắng cần tìm nước đi tại đỉnh u. Nước đi tối ưu cho Trắng là nước đi dẫn tới đỉnh con v, v là đỉnh tốt nhất (cho Trắng) trong số các đỉnh con của u
- Ta cần giả thiết rằng, đến lượt đối thủ chọn nước đi từ v, Đen cũng sẽ chọn nước đi tốt nhất cho anh ta
- Như vậy để chọn nước đi tối ưu cho Trắng tại đỉnh u, ta cần phải xác định giá trị các đỉnh của cây trò chơi gốc u
- Chú ý giá trị của các đỉnh lá (đỉnh ứng với các trạng thái kết thúc) là giá trị của hàm kết cuộc

- Đỉnh có giá trị càng lớn càng tốt cho Trắng, đỉnh có có trị càng nhỏ càng tốt cho Đen. Để xác định giá trị các định của cây trò chơi gốc u, ta đi từ mức thấp nhất lên gốc u.

- Giả sử v là đỉnh trong của cây và giá trị các đỉnh con của nó đã được xác định. Khi đó nếu v là đỉnh Trắng thì giá trị của nó đọc xác định là giá trị lớn nhất trong các giá trị của các đỉnh con. Nếu v là đỉnh Đen thì giá trị của nó là giá trị nhỏ nhất trong các giá trị của các đỉnh con

Ví dụ về xác định giá trị cho các đỉnh của cây trò chơi



Các hàm đệ quy xác định giá trị cho các đỉnh

```
Function MaxVal(u);
Begin
    if u là đỉnh kết thúc then MaxVal = f(u)
   else MaxVal = max {MinVal(v)| v là đỉnh con u}
End;

Function MinVal(u);
Begin
    if u là đỉnh kết thúc then MinVal = f(u)
   else MinVal = min {MaxVal(v)| v là đỉnh con u}
End;
```

Thủ tục chọn nước đi cho Trắng

```
Procedure Minimax(u,v);
Begin
    Val = -∞ ;
    for mỗi w là đỉnh con của u do
        if val <= MinVal(w) then
            {val = MinVal(w); v = w}
End;
```

Hạn chế

Thuật toán Minimax là thuật toán tìm kiếm theo độ sâu. Về mặt lý thuyết chiến lược Minimax cho phép ta tìm ra nước đi tối ưu cho Trắng nhưng trên thực tế chúng ta không có đủ thời gian để xác định được nước đi tối ưu vì phải xem xét đến toàn bộ các đỉnh của cây trò chơi, ví dụ đối với cờ vua nếu chỉ xét đến độ sâu 40 thì cây trò chơi đã có khoảng 10^{120} đỉnh. Độ phức tạp về thời gian của thuật toán có hàm số mũ

Khắc phục

- Hạn chế không gian tìm kiếm, cụ thể khi cần xác định nước đi cho trắng tại u ta chỉ xem xét cây trò chơi gốc u tới độ cao h nào đó
- Vẫn sử dụng chiến lược MINIMAX để tìm kiếm nước đi
- Tuy nhiên các lá của cây trò chơi hạn chế này có thể không phải là trạng thái kết thúc cho nên hàm kết cuộc không dùng được, thay vào đó ta sử dụng ***hàm đánh giá*** cho các lá của cây trò chơi hạn chế này

Hàm đánh giá

Hàm đánh giá eval ứng với mỗi trạng thái u của trò chơi với một giá trị eval(u), giá trị này là sự đánh giá "độ lợi thế" của trạng thái u.

Trạng thái u càng lợi cho Trắng nếu eval(u) là số dương càng lớn

Trạng thái u càng thuận lợi cho Đen nếu eval(u) là số âm càng nhỏ

Nếu eval(u) ≈ 0 thì không có lợi thế cho ai cả

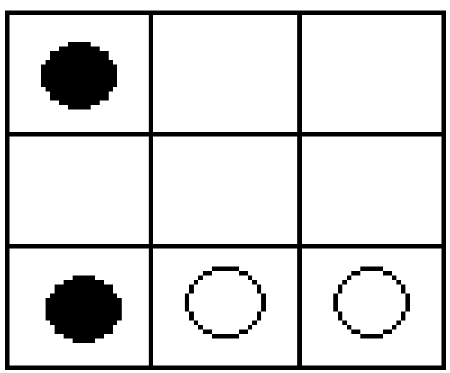
Nếu eval(u) = $+\infty$ thì Trắng thắng và eval(u) = $-\infty$ thì Đen thắng

Chất lượng của chương trình chơi cơ phụ thuộc rất nhiều vào hàm đánh giá

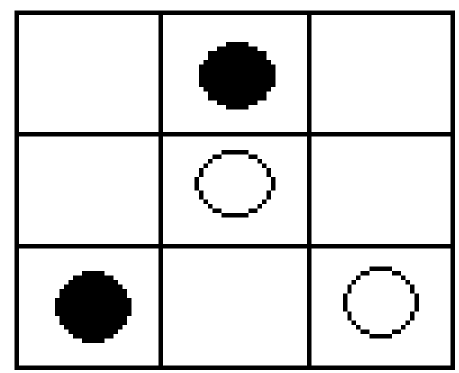
Ví dụ về xây dựng hàm đánh giá cho trò chơi Dodgem

2 trạng thái sau có lợi cho trắng hay đen, giá trị đánh giá các trạng thái này ??

u



v



Đánh giá

1. Cho điểm các vị trí

30	35	40
15	20	25
0	5	10

Giá trị quân trắng

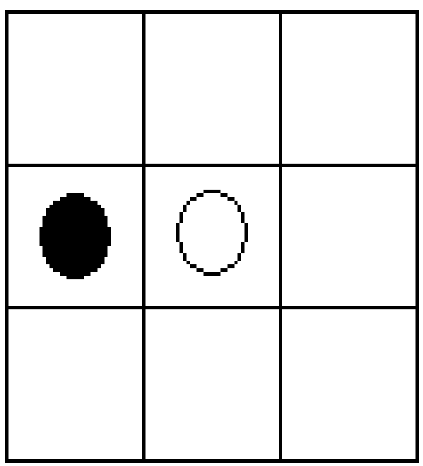
-10	-25	-40
-5	-20	-35
0	-15	-30

Giá trị quân đen

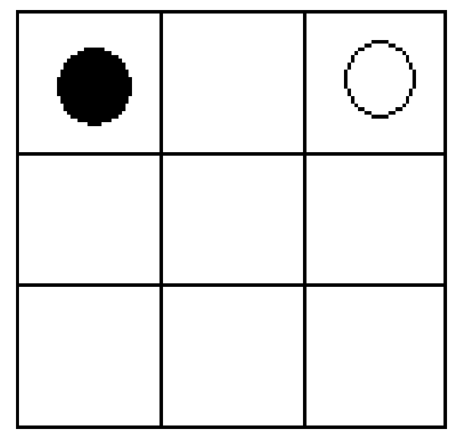
Đánh giá
2. Cho điểm cản

Nếu quân trắng cản trực tiếp một quân đen nó sẽ được thêm 40 điểm, nếu cản gián tiếp được thêm 30 điểm

Tương tự nếu quân đen cản trực tiếp quân trắng nó được thêm –40 điểm còn cản gián tiếp nó được thêm –30 điểm



Trắng cản trực tiếo đen được 40 điểm



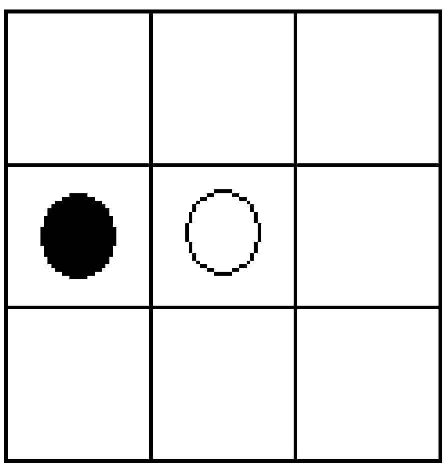
Trắng cản gián tiếp đen được 30 điểm

30	35	40
15	20	25
0	5	10

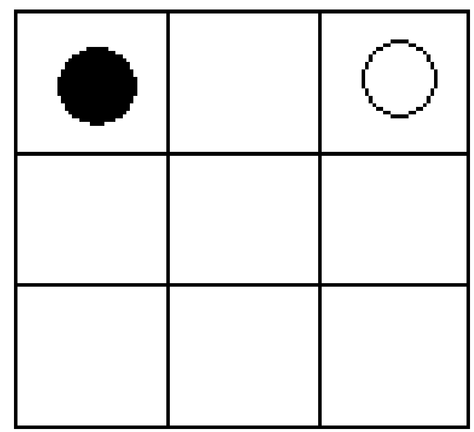
Giá trị quân trắng

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

Giá trị quân đen



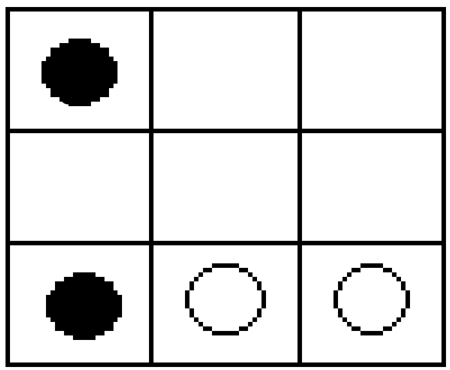
Trắng cản trực tiếo đen được 40 điểm

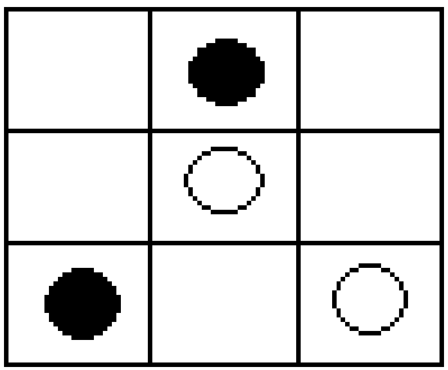


Trắng cản gián tiếp đen được 30 điểm

Áp dụng qui tắc trên cho 2 trạng thái sau ta có

eval(u) = 75, eval(v) = -5





5,10,0,-10,40,30=75

10,20,0,-25,30,-40=-5

Các hàm xác đinh giá trị các đỉnh - cải tiến

Function MaxVal(u,h);

Begin

if h=0 hoặc u là đỉnh kết thúc then MaxVal = eval(u)

else MaxVal = max {MinVal(v,h-1)| v là đỉnh con u}

End;

Function MinVal(u,h);

Begin

if h=0 hoặc u là đỉnh kết thúc then MinVal = eval(u)

else MinVal = min {MaxVal(v,h-1)| v là đỉnh con u}

End;

Thủ tục chọn nước đi cho Trắng -cải tiến

```
Procedure Minimax(u,v,h);
Begin
    val = -∞ ;
    for mỗi w là đỉnh con của u do
        if val <= MinVal(w,h-1) then
            {val = MinVal(w,h-1); v = w}
End;
```

BÀI TẬP

Vẽ cây trò chơi Dodgem với độ sâu =3

Xác định giá trị đánh giá của các đỉnh lá (đỉnh có độ sâu=3)

Xác định giá trị các đỉnh trong của cây

Chọn nước đi cho trắng từ đỉnh gốc và một số đỉnh khác

2) Nghiên cứu xây dựng hàm eval(u) cho bài toán cờ caro (3x3)

Phương pháp cắt cụt ALPHA-BETA

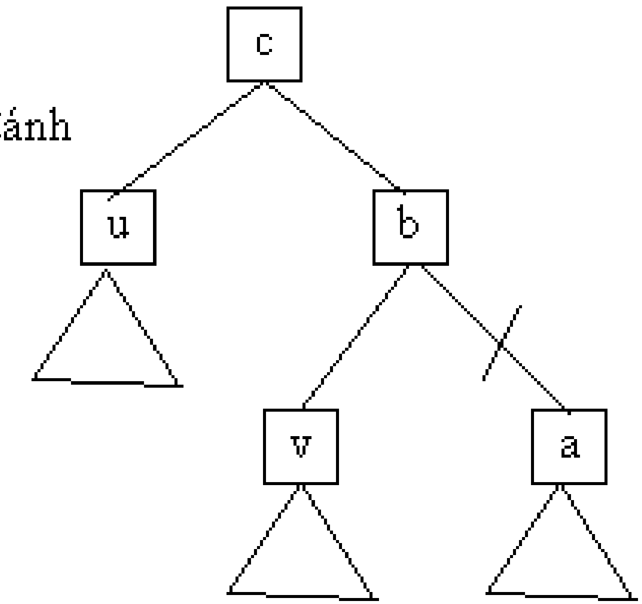
Để tìm nước đi cho Trắng tại u, giải thuật cải tiến không đòi hỏi xét toàn bộ các đỉnh của cây trò chơi gốc u mà chỉ xét các đỉnh của cây trò chơi gốc u với độ sâu hạn chế h

Tuy nhiên ta có thể giảm bớt số đỉnh cần đánh giá của cây trò chơi gốc u (độ sâu h) mà không ảnh hưởng gì tới sự đánh giá u

Phương pháp cắt cụt alpha - beta cho phép ta cắt bỏ các nhánh không cần thiết cho sự đánh giá đỉnh u

Tư tưởng của phương pháp

Nếu eval(u)>eval(v)
thì không cần đi xuống để đánh
giá đỉnh a nữa



max – ít nhất bằng eval(u)

min - nhiều nhất bằng eval(v)

max

Cắt bỏ cây con gốc a nếu eval(u)>eval(v)

b=min(v,a)<=v
u>v, suy ra **b<u**
c=max(u,b), vì b<u, max(u,b)=u, c=u

Kỹ thuật cài đặt

Dùng tham số α để ghi lại giá trị lớn nhất trong các giá trị của các đỉnh con đã đánh giá của một đỉnh trắng

Tham số β ghi lại giá trị nhỏ nhất trong các đỉnh con đã đánh giá của một đỉnh đen

Giá trị α, β sẽ được cập nhật trong quá trình tìm kiếm, được dùng như các biến địa phương trong các thủ tục sau

MaxVal(u, α, β) - hàm xác định giá trị của đỉnh trắng u

MinVal(u, α, β) - hàm xác định giá trị của đỉnh đen u

```
Function MaxVal(u, α, β);
Begin
    if u là lá của cây hạn chế hoặc u là đỉnh kết thúc
   then MaxVal = eval(u)
   else for mỗi đỉnh v là con của u do
     {α = max [α, MinVal(u, α, β)];
       if α>β then exit }// cắt bỏ cây con từ các đỉnh v còn lại
    MaxVal = α
End;
```

Function MinVal(u, α, β);

Begin

if u là lá của cây hạn chế hoặc u là đỉnh kết thúc

then MinVal = eval(u)

else for mỗi đỉnh v là con của u do

{β = max [β, MaxVal(u, α, β)];

if α>β then exit }// cắt bỏ cây con từ các đỉnh v còn lại

MinVal = β

End;

```
Procedure Alpha_beta(u,v)
Begin
α= -∞; β= +∞
for mỗi đỉnh w là con của u do
   if α<= MinVal(w,α,β) then
     {α=MinVal(w,α,β); v=w;}
End;
```