TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

===== o0o =====

BÀI TẬP LỚN MÔN: Trí Tuệ Nhân Tạo

Đề tài: Xây dựng trò chơi Tíc Tắc Toe

Giáo viên hướng dẫn: Lê Thị Thủy

Thực hiện: Nhóm 3

Thành viên nhóm: Đỗ Việt Đồng

Lê Quang Khải

Đinh Quốc Toản

Hà Nội, tháng 1 năm 2022

Mục lục

Danh mục hình ảnh:

LỜI CẢM ƠN

Lời đầu tiên cho phép chúng em gửi lời cảm ơn sâu sắc tới các thầy cô trong **khoa Công nghệ thông tin - Trường Đại học Công Nghiệp Hà Nội**, những người đã hết mình truyền đạt và chỉ dẫn cho chúng em những kiến thức, những bài học quý báu và bổ ích trong suốt kỳ học vừa qua. Để hoàn thành được đề tài này, đặc biệt chúng em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên **ThS. Lê Thị Thủy** người trực tiếp hướng dẫn, chỉ bảo chúng em trong suốt quá trình học tập và nghiên cứu để hoàn thành đề tài này. Sau nữa, chúng em xin gửi tình cảm sâu sắc tới gia đình và bạn bè vì đã luôn bên cạnh khuyến khích, động viên, giúp đỡ cả về vật chất linh tinh thần cho chúng em trong suốt qui trình học tập để chúng em hoàn thành tốt việc học tập của bản thân. Trong quá trình nghiên cứu và làm đề tài, do năng lực, kiến thức, trình độ bản thân chúng em còn hạn hẹp nên không tránh khỏi những thiếu sót và chúng em mong mỏi nhận được sự thông cảm và những góp ý từ quý thầy cô cũng như các bạn trong lớp. Chúng em xin chân thành cảm ơn!

Hà nội, ngày 20 tháng 1 năm 2022

Nhóm Sinh Viên Thực Hiện

===== *** =====

LỜI NÓI ĐẦU

Những năm gần đây, AI nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. Trí tuệ nhân tạo có thể được định nghĩa như một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh. Trí tuệ nhân tạo là một bộ phận của khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được ca lĩnh vực này. Ở thời điểm hiện tại, thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và các ứng dụng của trí tuệ nhân tạo.

Theo đà phát triển của công nghệ, ứng dụng trí tuệ nhân tạo luôn là xu hướng công nghệ tương lai mà các hãng công nghệ trên toàn thế giới đua nhau sáng tạo, nó là nền tảng cốt lõi của cuộc cách mạng công nghệ 4.0.

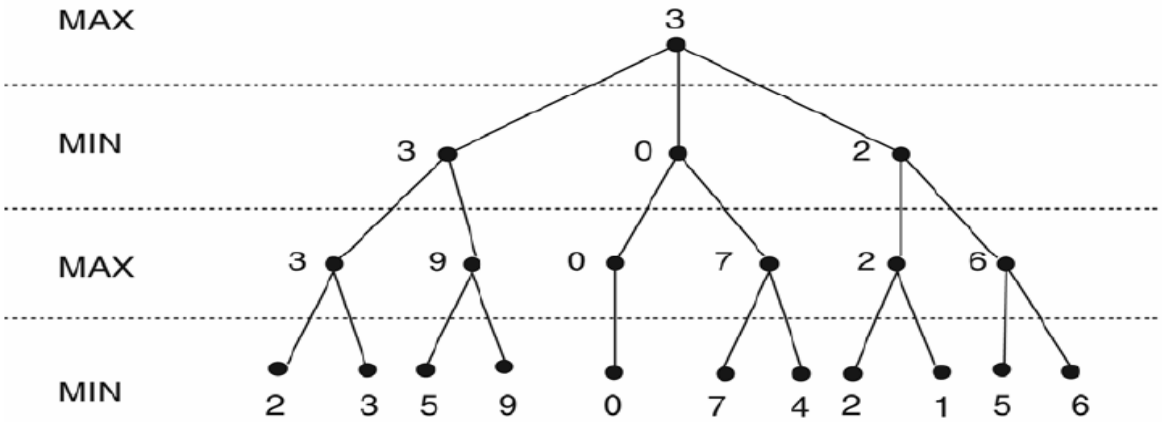
ML (Machine Learning) là một lĩnh vực của trí tuệ nhân tạo, được sinh ra từ khả năng nhận diện mẫu vả từ lý thuyết các máy tính có thể học mà không cần phải lập trình để xử lý các nhiệm vụ cụ thể nào đó.

Hầu hết mọi nghành công nghiệp đang làm việc với hàm lượng lớn đã liệu đều nhận ra tâm quan trọng của công nghệ AI. Những cái nhìn sáng suôt từ nguồn dữ liệu này chủ yếu dong thời gian thực sẽ giúp các tổ chức vận hành hiệu quả hơn hoặc tạo lợi thế cạnh tranh với các đối thủ. Các ứng dụng của AI đã quá quen thuộc với con người: xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trên Facebook , hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim trên ứng dụng Netflix… chỉ là một vài ứng dụng trong muôn vàn những ứng dụng của trí tuệ nhân tạo và cụ thể là ML - Machine Learning.

A. Giới thiệu thuật toán Minimax

I. Thuật toán minimax

Minimax là một quy tắc quyết định được sử dụng trong trí tuệ nhân tạo, lý thuyết trò chơi, lý thuyết quyết định, v.v. Minimax rất hữu ích vì chúng tận dụng khả năng của máy tính đánh giá một loạt các tình huống có thể xảy ra đang phát triển theo cấp số nhân. Đối với trò chơi, Minimax thường được sử dụng cho các trò chơi đối kháng. Có hai giá trị tiện ích được gọi là giá trị tối thiểu và giá trị tối đa giúp tác nhân AI quyết định bước đi tối ưu tiếp theo của nó. Đối với trường hợp này, AI ưu tiên các nước đi có giá trị lớn nhất là 10 và sau đó là giá trị nhỏ nhất là -10.(Tác nhân muốn thắng nhiều hơn là ngăn cản đối thủ thắng) Nếu không có nước đi nào có giá trị tối thiểu hoặc tối đa lớn hơn hoặc nhỏ hơn 10, nó sẽ chọn nước đi có giá trị tuyệt đối cao nhất là giá trị tối thiểu cộng với giá trị tối đa. Hơn nữa, Thuật toán Minimax giả định rằng đối thủ của bạn chơi tối ưu.

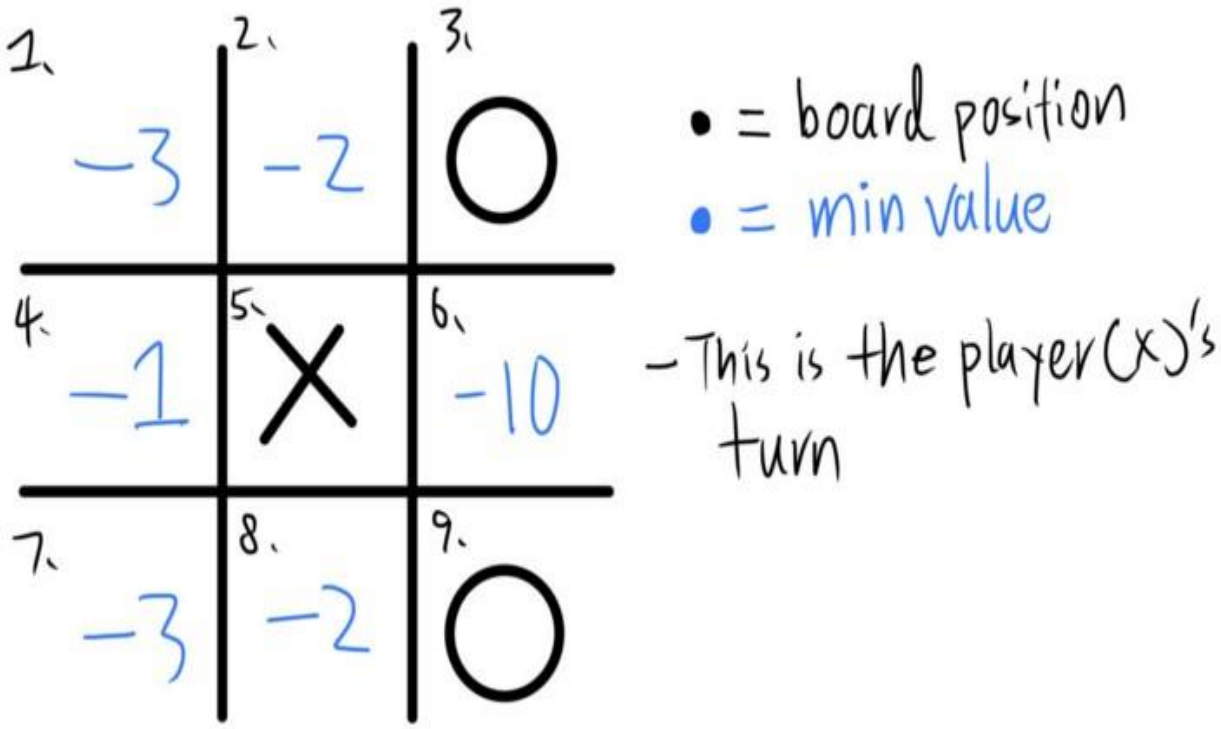


Ảnh 1.1: minimax với độ sâu lớp cố định

- ***Giá trị tối thiểu***

Đây là giá trị mà tác nhân AI tìm cách **giảm thiểu** tổn thất có thể xảy ra trong **trường hợp xấu nhất** . Đối với mọi làn đường ngang, dọc, chéo trên vị trí bàn cờ, giá trị tối thiểu -1 trên mỗi làn nếu 3 vị trí không có nước đi của bạn. Sau đó, đối với mỗi đối thủ khác di chuyển trên một làn đường, bạn -1 (Chỉ khi bạn không di chuyển trên làn đường đó). Tuy nhiên, nếu có 2 đối thủ di chuyển, giá trị tối thiểu của bạn trở thành -10 (Kẻ thù của bạn sẽ thắng nếu bạn không chặn vị trí đó!).

Giả sử bạn là người chơi (X), hãy xem xét tình huống bảng sau:



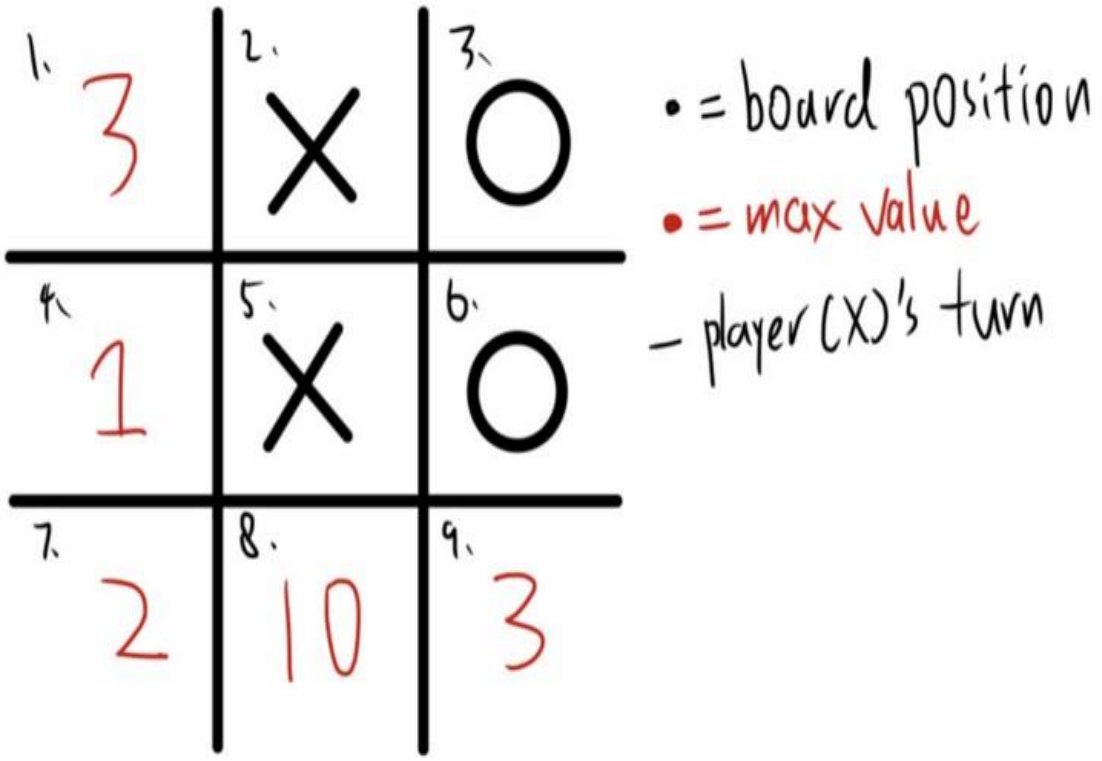
Ảnh 1.2: xét giá trị tối thiểu

Đối với vị trí bảng 1, không có X trên hàng của nó (làn ngang) và cột (làn dọc). Điều này có nghĩa là bằng cách đặt một dấu X ở đó, bạn đã loại bỏ 2 đường có thể để O giành chiến thắng. Ngoài ra, trên hàng đó (làn đường ngang), có chữ O để bạn -1 ngoài -2. Chúng tôi đã không trừ điểm từ làn đường chéo vì nó chứa X mặc dù có một O. Đối với vị trí bàn cờ 6, giá trị tối thiểu là -10 vì cột của nó (làn dọc) có 2 Os.

- ***Giá trị tối đa***

Đây là giá trị mà AI tìm cách **tối đa hóa** lợi nhuận có thể cho một **tình huống tốt nhất** . Đối với mọi làn đường ngang, dọc, chéo trên vị trí bàn cờ, bạn +1 mỗi làn nếu 3 vị trí không có nước đi của đối thủ. Sau đó, đối với mỗi lần di chuyển bổ sung bạn thực hiện trên làn đường, bạn +1 lại (Chỉ khi kẻ thù của bạn không thực hiện bất kỳ động thái nào trên làn đường đó). Tuy nhiên, nếu có 2 trong số các nước đi của bạn, giá trị tối đa của bạn sẽ trở thành +10 (Bạn thắng bằng cách di chuyển đến đó!)

Giả sử bạn là người chơi (X), hãy xem xét tình huống bảng sau:



Ảnh 1.3: xét giá trị tối đa

Đối với vị trí bàn cờ 1, bạn thêm 2 để không có đối thủ (O) di chuyển trên cột (làn dọc) và làn đường chéo của nó. Ngoài ra, bạn đã thực hiện một bước di chuyển (X) trên làn đường chéo nên bạn thêm 1 vào giá trị lớn nhất, dẫn đến kết quả là 3. Đối với vị trí bàn cờ 8, giá trị tối đa là 10 vì bạn đã thực hiện 2 (X) di chuyển trên cột của nó (làn dọc). Bằng cách đặt nước đi của bạn ở vị trí 8, bạn giành chiến thắng.

II. Thuật toán cắt Alpha-beta

Thuật toán Minimax đã được nói ở trên yêu cầu mở rộng tới toàn bộ khoảng không gian trống. Đây là 1 hạn chế nghiêm trọng, đặc biệt với vấn đề mở rộng khoảng trống. Để giải quyết khó khăn này một cách giải quyết là dung đánh giá heuristic tình trạng bước đi tiếp theo của người chơi để chọn một bước đi hiện tại cho cùng 1 người chơi. Chúng ta sẽ minh họa quá trình tính toán của các biện pháp heuristic của một nút dưới đây với trò chơi TIC TAC TOE

Hàm Heuristic: E(n) – M(n) – O(n)

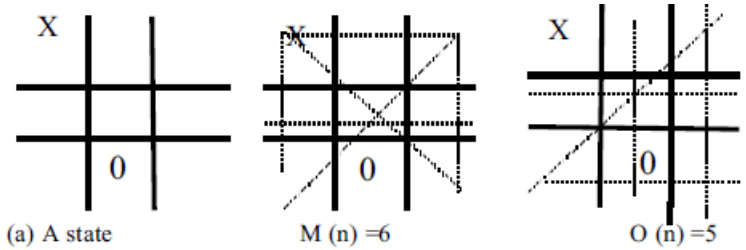
Trong đó:

M(n) là tổng số đường thắng có thể của tôi

O(n) là tổng số đường thắng có thể của đối thủ

E(n) là trị số đánh giá tổng cộng cho trạng thái n

Ví du:



Ảnh 2.1: hàm heuristic trong trò chơi tic – tac – toe

M(n) = 6, O(n) = 5 và E(n) = M(n) – O(n) = 6 – 5 =1.

Bây giờ, chúng tôi sẽ thảo luận về một loại mới của thuật toán, mà không yêu cầu mở rộng toàn bộ không gian exhaustively. Thuật toán này được gọi

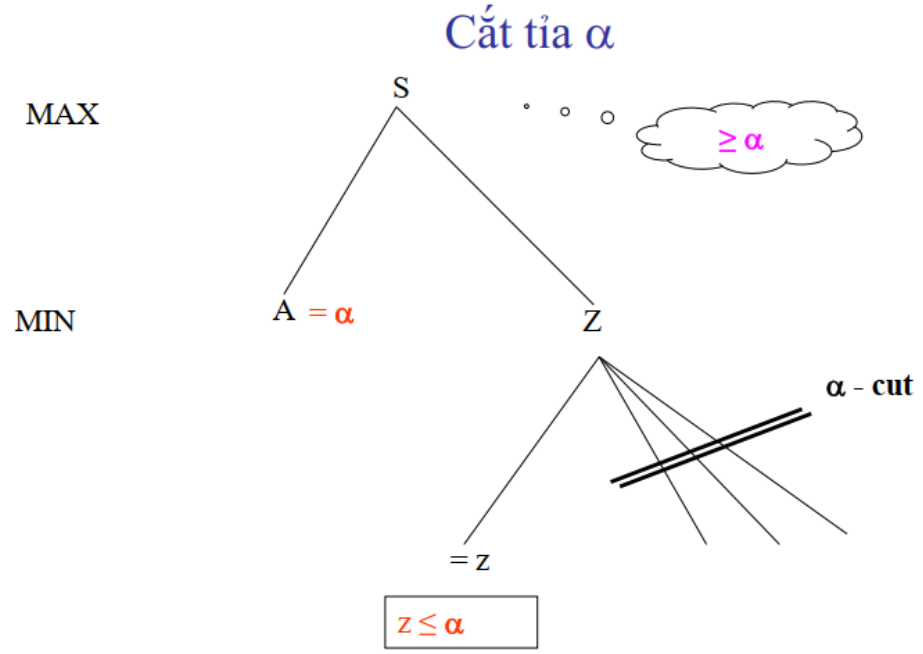
là thuật toán cắt alpha-beta. Trong thuật toán này, hai phụ lớp của chuyển động là xem xét để lựa chọn di chuyển hiện tại từ lựa chọn thay thế. Alpha và beta biểu hai cách cắt các cấp liên kết với các nút MAX và MIN. Giá trị của alpha MAX node không thể giảm, trong khi giá trị beta của các nút MIN có thể không tăng. Nhưng làm thế nào chúng ta có thể tính toán các giá trị alpha và beta? Chúng là những giá trị sao lưu lên đến gốc như Minimax. Có một vài thú vị điểm đó có thể được khám phá ở giai đoạn này

Trước khi quá trình tính toán MAX / MIN của các giá trị sao lưu của lớp con, thuật toán alpha-beta cắt ước lượng E(n) tại tất cả các nút rìa n.
Thuật toán cắt alpha-beta Tìm kiếm theo kiểu depth-first.

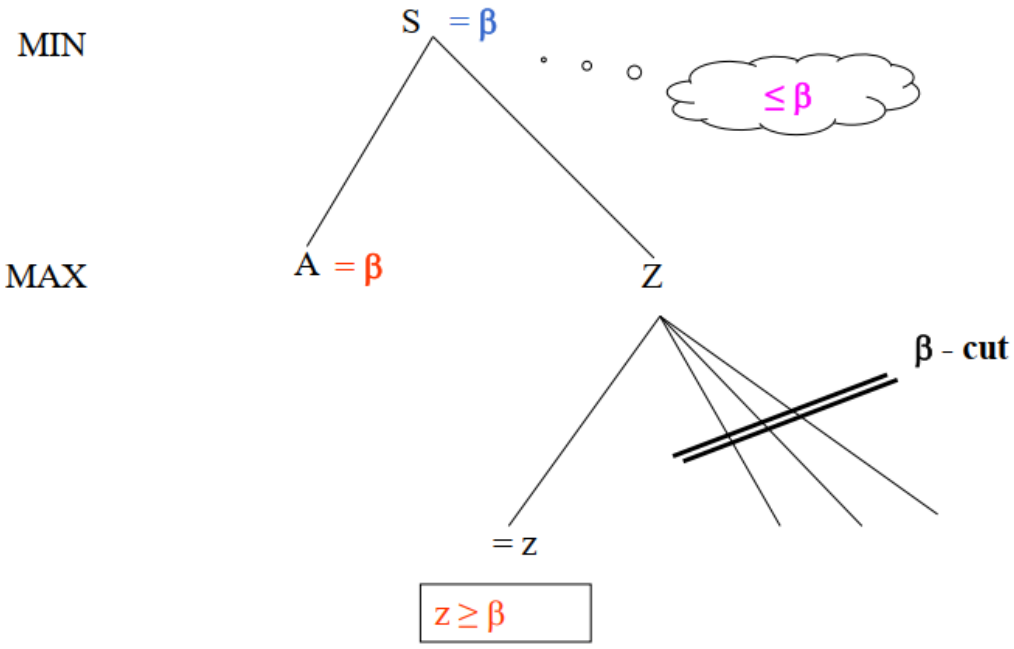
Nút MAX có 1 giá trị a (luôn tăng)
Nút MIN có 1 giá trị B (luôn giảm)
TK có thể kết thúc dưới bất kỳ:

- Nút MIN nào có ß <= α của bất kỳ nút cha MAX nào.
- Nút MAX nào có α >= ß của bất kỳ nút cha MIN nào.

Giải thuật cắt tia α-ß thể hiện mối quan hệ giữa các nút ở lớp n và n+2, mà tại đó toàn bộ cây có gốc tại lớp n+1 có thể cắt bỏ.

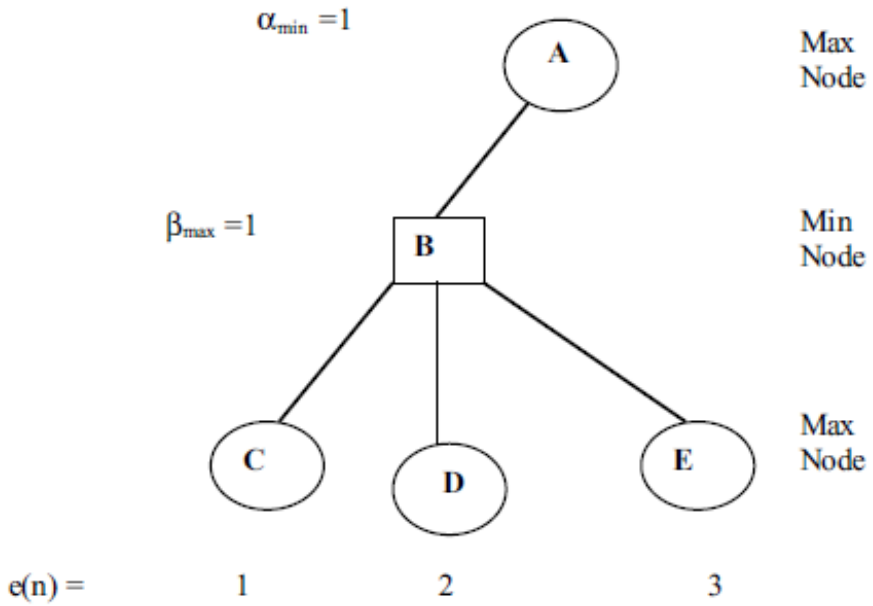


Ảnh 2.2: cắt tỉa alpha



Ảnh 2.3: cắt tỉa beta

- Các bước thực hiện của thuật toán cắt alpha – beta



Ảnh 2.4: *Hai lớp di chuyển của Maximizer với e tính (n) tại rìa các nút: C, D, E; sao lưu các giá trị tại nút B và A và thiết lập của αmax và βmin giá trị tại các nút A và B tương ứng*

Dựa trên các cuộc thảo luận ở trên, hiện nay chúng tôi trình bày các bước chính trong các α-β thuật toán tìm kiếm :

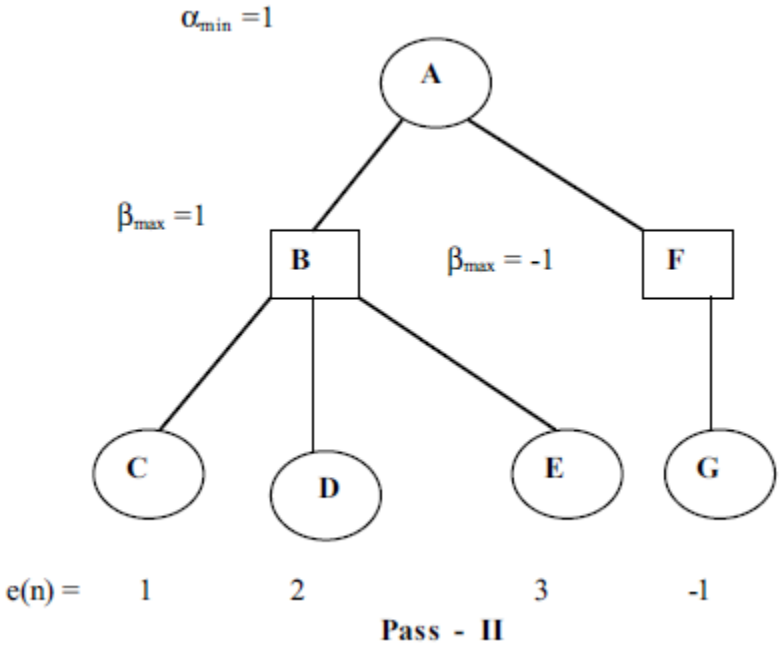
I) Tạo một nút mới, nếu nó là di chuyển bắt đầu, mở rộng khác hiện cây theo độ sâu cách đầu tiên. Để thực hiện một quyết định về lựa chọn của một di chuyển ở độ sâu d, cây nên được mở rộng ít nhất lên đến độ sâu (d + 2).

II) Tính e (n) cho tất cả để lại (rìa) các nút n trong cây.

IV) Tính α ' min (cho các nút tối đa) và β ' tối đa các giá trị (đối với các nút min) tại gốc của các nút rìa bởi các nguyên tắc sau đây.

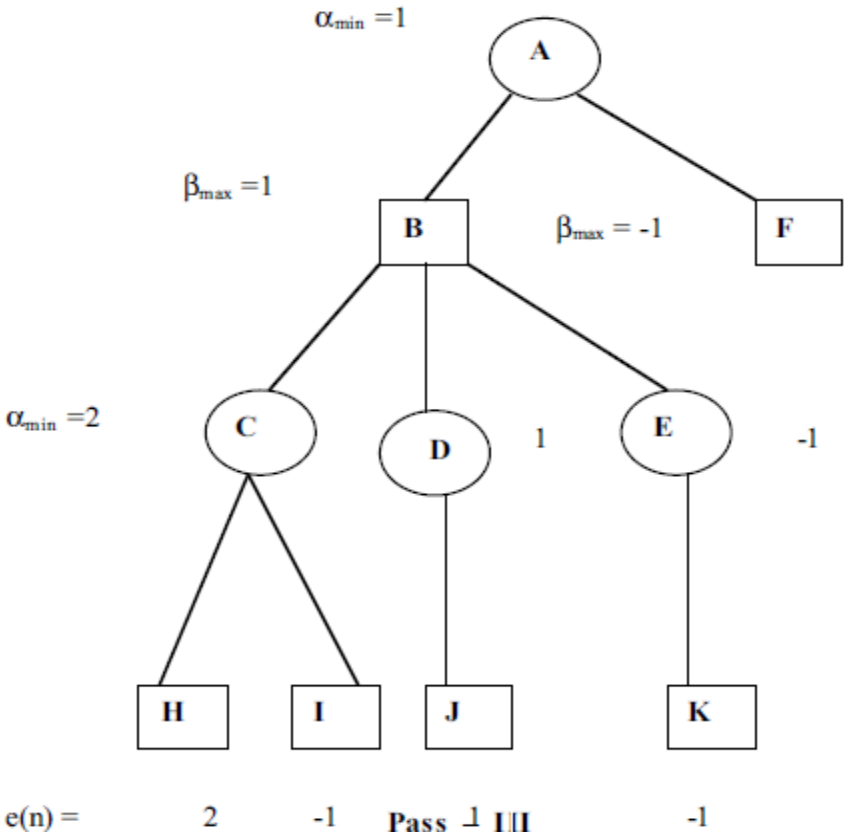
Ước tính tối thiểu của các giá trị (e hoặc α) sở hữu do β của con của một nút MINIMIZER N và gán nó ' tối đa giá trị.

Tương tự như vậy, ước tính tối đa của các giá trị (e hoặc β) sở hữu do con của một nút Maximizer N và gán nó giá trị αmin .



Ảnh 2.5: *Những cách nghĩ để có thể di chuyển thay thế F và cũng tự sinh ra ở lớp tiếp theo là G e(G)=-1, β max ở F =-1, , β max ở F nhỏ hơn so với α min ở A*

Như vậy không có nhu cầu tìm kiếm dưới F, G có thể bớt không gian tìm kiếm



Ảnh 2.6: *Nút G đã bị cắt tỉa , các nút C, D và E được mở rộng e(n) được thiết lập ở n = H , I , J và K và giá trị của αmin được ước lượng tại các nút C, D, E.Bởi vì giá tri αmin ở C lớn hơn giá trị βmax ở B, giá tri αmin ở D bằng giá trị βmax ở B, ở đó ko cần tìm kiếm dưới nút C và D*

IV) Nếu các nút của MAXIMIZER có các giá trị αmin. Sau đó giá trị αmin = MAX(αmin). Một mặt khác nếu các nút của MINIMIZER có giá trị βmax và sau đó βmax = MAX(βmax)

V) Nếu thiết lập giá trị βmax ở nút MINIMIZER tại N nhỏ hơn αmin của nút cha MAXIMIZER tại N' thì sau đó ko cần tìm kiếm dưới nút N. Tương tự Nếu thiết lập giá trị αmin ở nút MAXIMIZER tại N nhỏ hơn αmin của nút cha MINIMIZER tại N' thì sau đó ko cần tìm kiếm dưới nút N

Những bước ở trên được thực hiện cho đến khi nào trò chơi kết thúc

B. Trò chơi tíc tắc toe

I. Giới thiệu trò chơi

Trò chơi này được chơi bởi 2 đối thủ với 9 ô. Một trong 2 đối thủ sẽ đi trước đánh O hoặc (X) vào 1 ô bất kỳ trên bàn cờ , đối thủ còn lại chọn 1 trong 8 ô còn lại để đi .Hai đối thủ thay nhau đánh vào các ô trống cho đến khi có 1 đối thủ có 3 ô nằm trên 1 đường thẳng, hoặc 1 hàng ngang, hoặc 1 hàng chéo trước thì thắng. Nếu hết 9 ô cờ mà không có đối thủ nào có 3 ô nằm trên 1 đường thẳng, hoặc 1 hàng ngang, hoặc 1 hàng chéo thì ván cờ kết thúc với tỷ số hòa

X		O
	X	O
	O	X

Trường hợp X thắng.

O	O	X
X	X	O
O	O	X

Trường hợp hòa nhau.

Ảnh b.1.1

II. Mục tiêu trò chơi

Đến lượt chơi mỗi người cố gắng để tạo ra 3 quân cờ nằm trên 1 đường thẳng để là người chiến thắng hoặc cố ngăn cản người kia tạo được 3 quân cờ trên một đường thẳng, hoặc 1 hàng ngang, hoặc 1 hàng chéo.

III. Hướng giải quyết

Sử dụng thuật toán Minnimax – cơ chế cắt tỉa alpha beta để tìm nước đi tốt nhất có lợi cho từng người chơi .

Hai đối thủ trong 1 trò chơi được gọi là MIN và MAX .MAX là đại diện cho đối thủ quyết dành thắng lợi hay cố gắng tối ưu hóa ưu thế của mình. Ngược

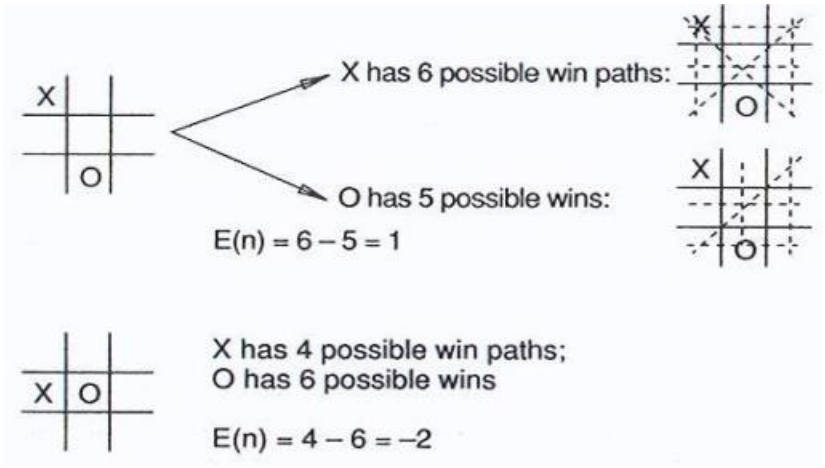
lại MIN là đối thủ cố gắng tối thiểu hóa điểm số của MAX . Ta giả thiết MIN cũng dùng thong tin như MAX

Khi áp dụng thủ tục Minnimax ta đánh dấu luân phiên từng mức trong không gian tìm kiếm phù hợp với đối thủ có bước đi ở mức đó trong ví dụ trên MIN được quyền đi trước, từng nút lá được gán giá trị 0 hay 1 tùy theo đó là thắng cuộc với MIN hay MAX.Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị qua các nút cha mẹ kế tiếp nhau theo luật sau:

-Nếu trạng thái cha mẹ là nút MAX , gán cho nó giá trị tối đa của các con cháu của nó

- Nếu trạng thái cha mẹ là nút MIN , gán cho nó giá trị tối thiểu của con cháu của nó

Giá trị được gán cho từng trạng thái bằng cách đó sẽ chỉ rõ trạng thái tốt nhất mà đối thủ này có thể đạt được . Các giá trị này sẽ được dùng để lựa chọn các bước đi có thể có. Kết quả của việc áp dụng Minnimax vào đồ thị không gian trạng thái đối với trò chơi Nim được thể hiện như hình trên. Vì tất cả các nước đi đầu tiên có thể sảy ra cho MIN sẽ dẫn đến giá trị 1 nên đối thủ MAX luôn có thể bắt trò chơi giành thắng lợi cho mình bất kể nước đi đầu tiên của MIN như nào



Hàm Heuristic: **E(n) = M(n) – O(n)**

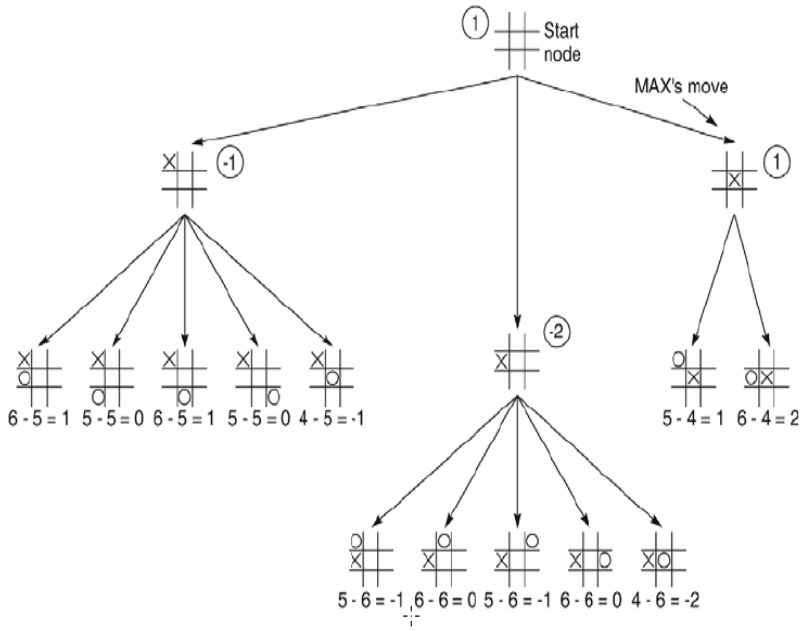
Trong đó: M(n) là *tổng số đường thắng có thể của tôi*

O(n) là *tổng số đường thắng có thể của đối thủ*

E(n) là *trị số đánh giá tổng cộng cho trạng thái n*

Ảnh b.3.1: Heuristic trong trò chơi tic-tac-toe

Áp dụng vào trò chơi:



Ảnh b.3.2

IV. Thuật toán

FullCode :

https://github.com/dovietdong/game-tic-tac-toe.git

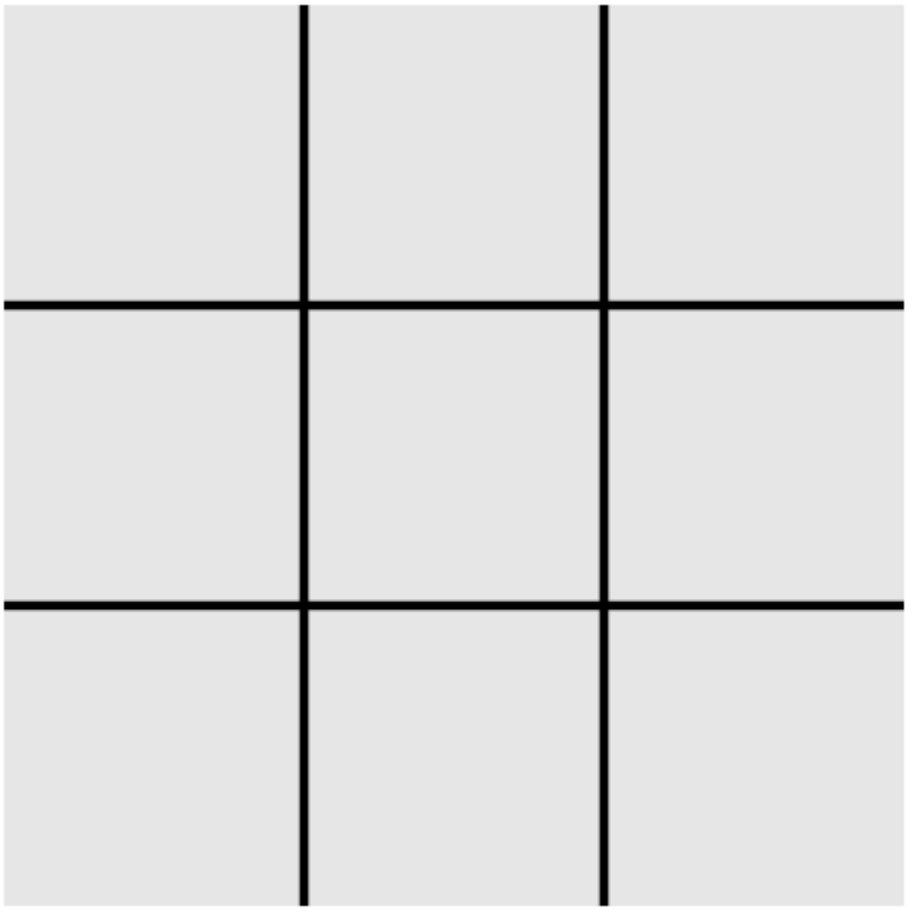
Code thuật toán Minimax

```
function minimax(board, depth, isMaximizing) {
 let bestScoreA;
 let bestScoreB;
 let result = checkWinner();
 if (result !== null) {
  return scores[result];
 }
 if (isMaximizing) {
  let bestScoreA = -Infinity;
  for (let i = 0; i < 3; i++) {
   for (let j = 0; j < 3; j++) {
    // Is the spot available?
    if (board[i][j] == ' ') {
     board[i][j] = ai;
     let score = minimax(board, depth + 1, false);
     bestScoreA = max(score, bestScoreA);
     board[i][j] = ' ';
     if (bestScoreA >= bestScoreB) {
      break;
```

```
    }
   }
  }
 }return bestScoreA;
} else {
 let bestScoreB = Infinity;
 for (let i = 0; i < 3; i++) {
  for (let j = 0; j < 3; j++) {
   // Is the spot available?
   if (board[i][j] == ' ') {
    board[i][j] = human;
    let score = minimax(board, depth + 1, true);
    bestScoreB = min(score, bestScoreB);
    board[i][j] = ' ';
    if (bestScoreA <= bestScoreB) {
     break;
    }
   }
  }
 }
 return bestScoreB;
}
}
```

V. Giới thiệu chương trình

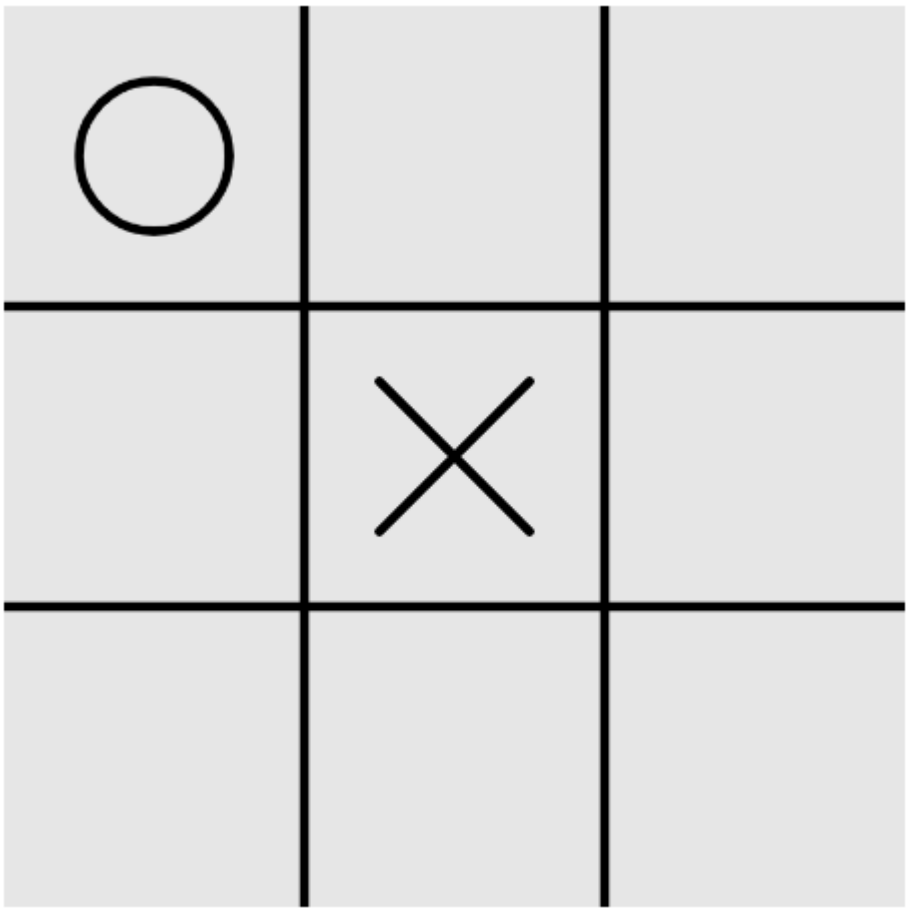
1. Giao diện chính của chương trình



Ảnh b.5.1

2. *Chức năng chính của chương trình*

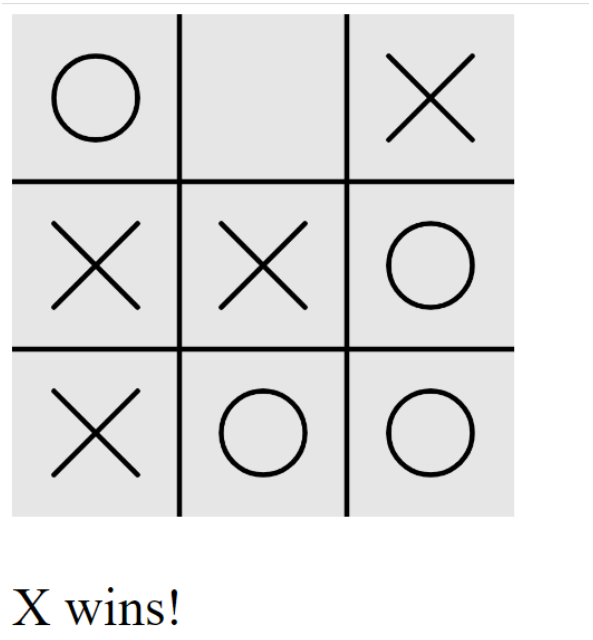
Người chơi O di chuyển trước, máy là X



Ảnh b.5.2

3. *Kết quả của trò chơi*

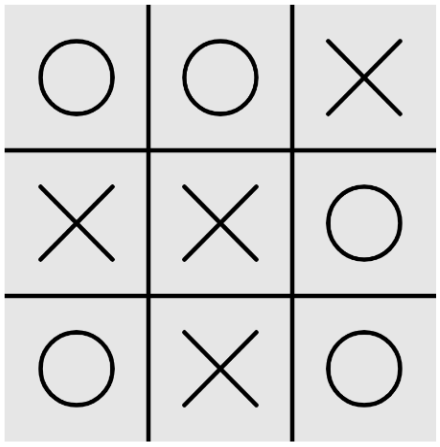
Máy thắng:



X wins!

Ảnh b.5.3

Máy vs người chơi hòa



Tie!

Ảnh b.5.4

C. Tài liệu tham khảo

https://ichi.pro/vi/toi-da-lap-trinh-mot-tro-choi-tic-tac-toe-voi-thuat-toan-minimax-nhung-no-hoat-dong-nhu-the-154380859001326

https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/

Sách tham khảo:

1,Ebook Python cơ bản

-Pythontiếng Việt siêu cơ bản của Võ Tuấn Duy xuất bản 30-08-2018

-Với 15 chương, mỗi chương trình bày một khía cạnh của Python thông qua những trải nghiệm thực tế của tác giả. Tài liệu miễn phí này sẽ giúp bạn nhanh chóng tự học ngôn ngữ lập trình Python.

2, **A Byte of Python**

-Copyright © 2003 – 2005 của Swaroop C.H

- Dành cho người hoàn toàn chưa có kiến thức gì về lập trình.
- Cuốn *sách* giúp bạn *nắm* bắt nhanh các nguyên tắc cơ bản của lập trình.
- Cung cấp các chỉ dẫn và tutorial để bước đầu làm quen với ngôn ngữ lập trình Python.