**HỌC VIỆN NÔNG NGHIỆP VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\*\***

****

**Báo Cáo Bài Tập Nhóm**

**Môn: Trí Tuệ Nhân Tạo**

**Đề Tài:**

**Phát Triển Game Cờ Caro**

**Với**

**Giải Thuật Minimax**

**Giảng viên hướng dẫn: PGS. TS. Nguyễn Thị Thủy**

**Nhóm Sinh Viên:**

**Đỗ Đăng Anh - 614083**

**Nguyễn Ngọc Công - 611213**

**Trương Văn Khải - 611236**

***Hà Nội – 2019***

Mục Lục

[I. Nhận Xét Của Giảng Viên 2](#_Toc25696915)

[II. Mở Đầu 3](#_Toc25696916)

[III. Giải Thuật Minimax 3](#_Toc25696917)

[IV. Game Cờ Caro 5](#_Toc25696918)

[1. Thiết kế game 5](#_Toc25696919)

[2. Các cải tiến áp dụng cho giải thuật 8](#_Toc25696920)

[V. Một số kết quả 9](#_Toc25696921)

[VI. Kết Luận 10](#_Toc25696922)

[VII. Tài liệu tham khảo 11](#_Toc25696923)

1. Nhận Xét Của Giảng Viên
2. Mở Đầu

Nhằm tìm hiểu về trí tuệ nhân tạo và những ứng dụng của nó, nhóm chúng em – nhóm 1 đã quyết định chọn đề tài “Ứng dụng thuật toán minimax vào trong game cờ caro”

Trước hết, chúng ta sẽ cùng tìm hiểu sơ qua về thuật toán minimax:

Minimax (còn gọi là minmax) là một phương pháp trong lý thuyết quyết định có mục đích là tối thiểu hóa (minimize) tổn thất vốn được dự tính có thể là "tối đa" (maximize). Có thể hiểu ngược lại là, nó nhằm tối đa hóa lợi ích vốn được dự tính là tối thiểu (maximin). Nó bắt nguồn từ trò chới có tổng bằng không. Nó cũng được mở rộng cho nhiều trò chơi phức tạp hơn và giúp đưa ra các quyết định chung khi có sự hiện diện của sự không chắc chắn.

Một phiên bản của giải thuật áp dụng cho các trò chơi như tic-tac-toe, khi mà mỗi người chơi có thể thắng, thua, hoặc hòa. Nếu người chơi A có thể thắng trong 1 nước đi, thì "nước đi tốt nhất" chính là nước đi để dẫn đến kết quả thắng đó. Nếu người B biết rằng có một nước đi mà dẫn đến tình huống người A có thể thắng ngay ở nước đi tiếp theo, trong khi nước đi khác thì sẽ dẫn đến tình huống mà người chơi A chỉ có thể, tốt nhất, là hòa thì nước đi tốt nhất của người B chính là nước đi sau.

Ta sẽ nắm rõ, thế nào là một nước đi "tốt nhất". Giải thuật Minimax giùm tìm ra nước đi tốt nhất, bằng cách đi ngược từ cuối trò chơi trở về đầu. Tại mỗi bước, nó sẽ ước định rằng người A đang cố gắng tối đa hóa cơ hội thắng của A khi đến phiên anh ta, còn ở nước đi kế tiếp thì người chơi B cố gắng để tổi thiểu hóa cơ hội thắng của người A (nghĩa là tối đa hóa cơ hội thắng của B).

1. Giải Thuật Minimax

* Giải thuật Minimax với các nước đi khác nhau:

Một thuật toán minimax là một thuật toán đệ quy cho việc lựa chọn bước đi kế tiếp trong một trò chơi có hai người chơi. Một giá trị được gán cho mỗi vị trí hay một trạng thái của trò chơi. Giá trị này được tính toán bằng một hàm tính giá trị vị trí và nó cho biết độ tốt nếu như một người chơi đạt được đến đó. Người chơi sau đó đi một bước làm tối đa giá trị tối thiểu của vị trí là kết quả từ tập hợp những bước đi có thể của đối thủ. Nếu đó là phiên A sẽ đi, A sẽ cho một giá trị cho mỗi bước đi hợp pháp của anh ta.

Một phương pháp bố trí là gán cho một số vị trí thắng cho A như là +1 và cho B là −1. Điều này sẽ dẫn đến lý thuyết trò chơi tổ hợp được phát triển bởi John Horton Conway.

Một cách khác là sử dụng một quy định rằng nếu như kết quả của một bước đi là một chiến thắng lập tức cho A nó được gán dương vô hạn và, nếu như là một chiến thắng lập tức cho B, âm vô hạn. Giá trị cho A của bất kì nước đi nào khác là giá trị minimum của các giá trị kết quả từ mỗi bước trả lời có thể của B. (A được gọi là người chơi là cực đại và B gọi là người chơi làm cực tiểu), do vậy được gọi là thuật toán minimax. Thuật toán trên sẽ gán một giá trị dương hay âm vô hạn cho mỗi vị trí bởi vì giá trị của mỗi vị trí sẽ là giá trị của một số vị trí thắng hay thua nào đó. Thông thường nhìn chung điều này chỉ có thể xảy ra tại điểm cuối của những trò chơi phức tạp như cờ vua hay cờ vây, bởi vì về mặt tính toán ta không có khả năng tính xa đến mức kết thúc trò chơi, trừ khi là trò chơi sắp kết thúc, và các vị trí không đi khác nhau được cho các giá trị hữu hạn như là các đánh giá về mức độ tin tưởng là chúng sẽ dẫn đến chiến thắng cho người này hay người khác.

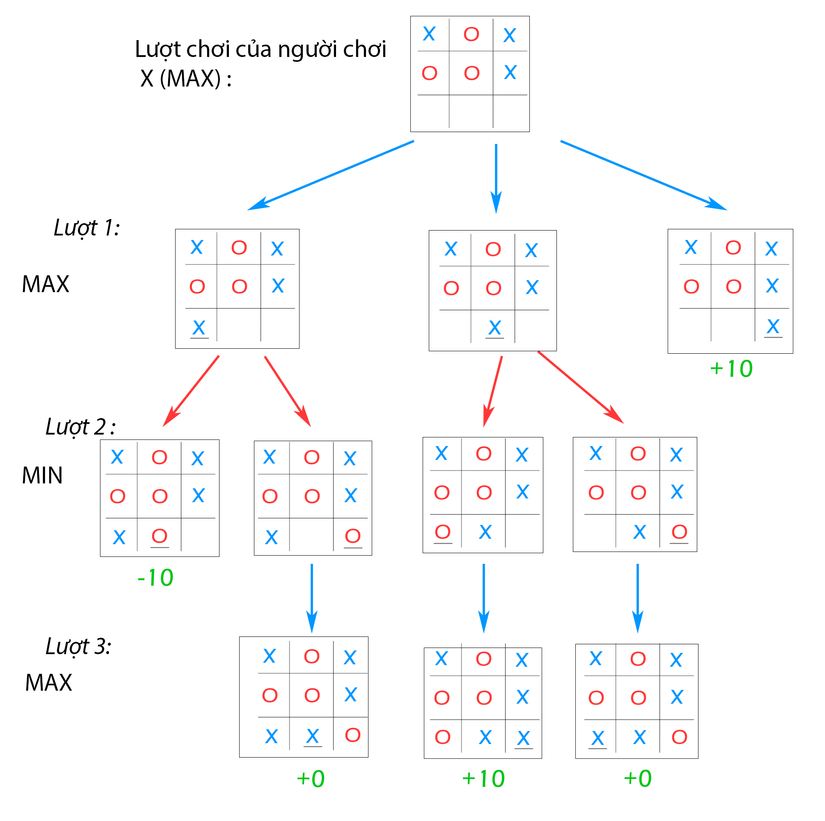
Điều này có thể được mở rộng nếu như chúng ta cung cấp một hàm đánh giá heuristic đưa ra các giá trị cho các vị trí trò chơi chưa phải là cuối cùng mà không xét tất cả mọi trường hợp theo sau một chuỗi đầy đủ. Chúng ta sau đó có thể giới hạn thuật toán minimax để chỉ xét một số nào đó các nước đi kế tiếp. Số này được gọi là "số bước kế tiếp", đo bằng "ply". Ví dụ, "Deep Blue" nhìn trước 12 ply.

Thuật toán này có thể được nghĩ như là khám phá các node của một cây trò chơi. Số cắt xén hiệu quả của một cây là trung bình của số các con của mỗi nốt (i.e., trung bình của các nước đi hợp pháp trong một vị trí). Số lượng các nodes được khám phá thường là tăng theo hàm mũ với số lượng ply

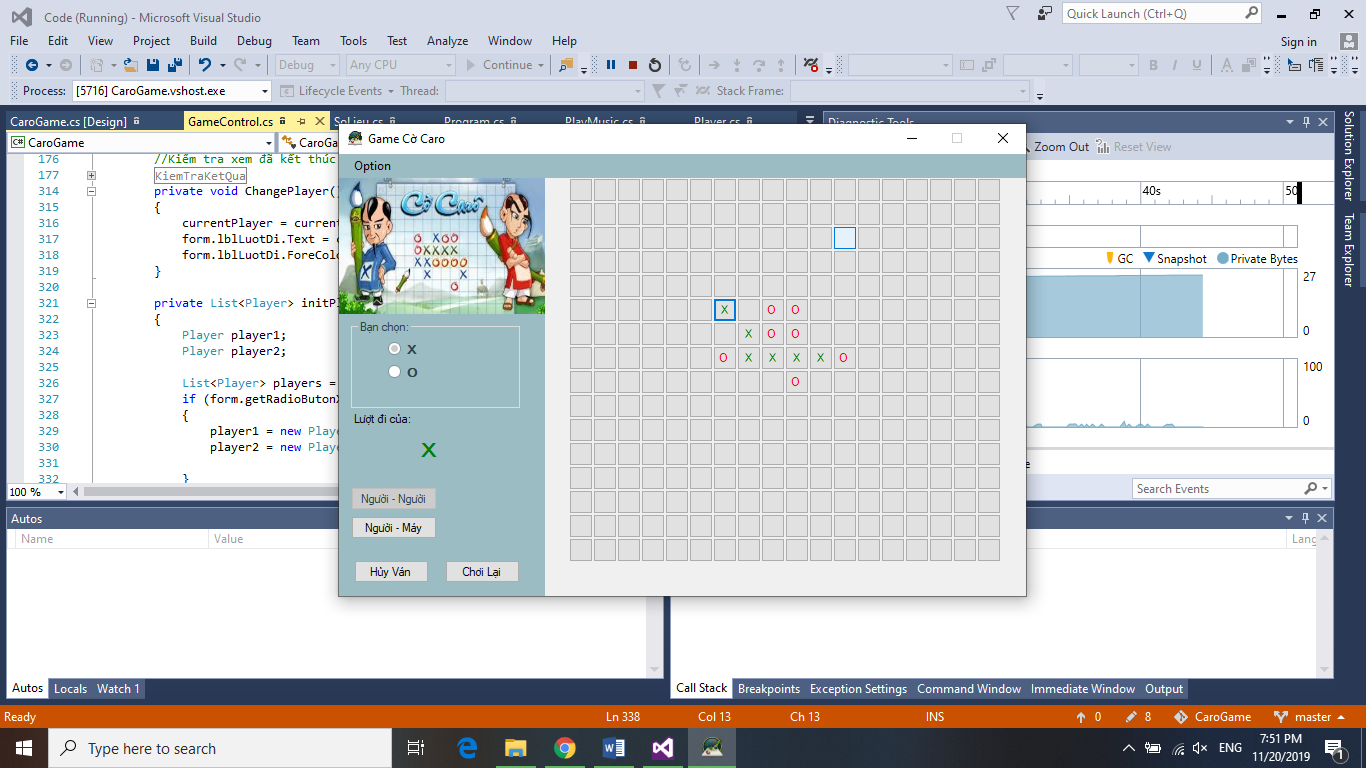
(nó sẽ nhỏ hơn hàm mũ nếu đánh giá các nước đi bắt buộc hay là các bước lặp lại). Số lượng các nodes cần khám phá cho việc phân tích một trò chơi do đó gần bằng số cắt xét nâng lên luỹ thừa số ply. Do vậy là không thể phân tích trò chơi ví dụ như cờ vua một cách hoàn toàn chỉ bằng thuật toán minimax.

Để khắc phục tình trạng này ta có thể dùng phương pháp cắt tỉa Alpha-Beta để giảm bớt số nhánh không cần thiết

Giải thuật Minimax Hai người chơi trong game được đại diện là MAX và MIN. MAX đại diện cho người chơi luôn muốn chiến thắng và cố gắng tối ưu hóa ưu thế của mình còn MIN đại diện cho người chơi cố gắng cho người MAX giành số điểm càng thấp càng tốt. Giải thuật Minimax thể hiện bằng cách định trị các Node trên cây trò chơi: Node thuộc lớp MAX thì gán cho nó giá trị lớn nhất của con Node đó. Node thuộc lớp MIN thì gán cho nó giá trị nhỏ nhất của con Node đó. Từ các giá trị này người chơi sẽ lựa chọn cho mình nước đi tiếp theo hợp lý nhất.



1. Game Cờ Caro
2. Thiết kế game



* + - Game được thiết kế với phần menu trạng thái của game, phần tùy chọn và phần bàn cờ.
    - Bàn Cờ: 16x18 ô cờ
    - Luật đánh: 5 quân đồng màu trên 1 hàng = Win
    - Chức Năng: Đánh cờ Người vs Người, Người vs Máy
    - Nền tảng lập trình sử dụng: Winform(C# .Net)
* Các class:
* Player – chứa thông tin về người chơi (Người hay máy, chọn x hay o)
* Ô cờ - chứa thông tin của mỗi ô cờ trên bàn cờ(sở hữu bởi player nào)
* Game control – chứa các hàm chức năng điều khiển trò chơi (vẽ bàn cờ, khởi tạo trận đấu, tính điểm, xác định thắng thua, điều khiển giải thuật AI)
* Ý tưởng: Tạo 1 list các button theo kích thước bàn cờ, mỗi button là một ô cờ. Khi người chơi nhấn vào 1 button tương tương với việc đánh 1 nước cờ.
* Mỗi lượt đánh, sẽ tiến hành kiểm tra kết quả và đổi tiếp lượt chơi.
* Trường hợp Người vs Máy:
* Máy đánh trước: khi bàn cờ hoàn toàn trống, máy sẽ đánh ngẫu nhiên trong một khoảng các ô cờ. Từ các nước đánh sau của máy sẽ áp dụng thuật toán.
* Người đánh trước: người chơi sẽ đánh một nước sau đó đổi lượt cho máy.

1. **Giải Thuật Minimax áp dụng trong game:**

* Với mỗi lần tới lượt đánh của máy, duyệt từng ô cờ trống trên bàn cờ để xác định các trạng thái bàn cờ dựa vào trạng thái của các ô xung quanh nó
* Xác định trạng thái và duyệt gắn điểm tấn công, phòng thủ cho từng trạng thái
* Các hướng duyệt: 4 hướng (dọc, ngang, đường chéo chính, đường chéo phụ)
* Cách tính điểm: với mỗi ô cờ trên đường duyệt nếu các ô cờ xung quanh nó có cùng sở hữu => cộng thêm một số điểm
* Điểm tấn công = tổng điểm của các quân đồng minh – tổng điểm của các quân đối thủ xung quanh
* Điểm phòng ngự = tổng điểm của các quân đối thủ - tổng điểm của các quân đồng minh
* **Giả mã**

Điểm max = 0; imax = 0; jmax = 0;

For i = 1 to hàng

For j = 1 to cột

If ô cờ [i,j ] chưa có sở hữu

Điểm tấn công = tấn công hàng dọc + hàng ngang+chéo chính+chéo phụ

Điểm phòng thủ = phòng thủ hàng dọc+hàng ngang+chéo chính+chéo phụ

If điểm tấn công > điểm phòng thủ: điểm tạm = điểm tấn công

Else điểm tạm = điểm phòng thủ

If điểm max < điểm tạm

Điểm max = điểm tạm ; imax = i; jmax = j.

End.

End.

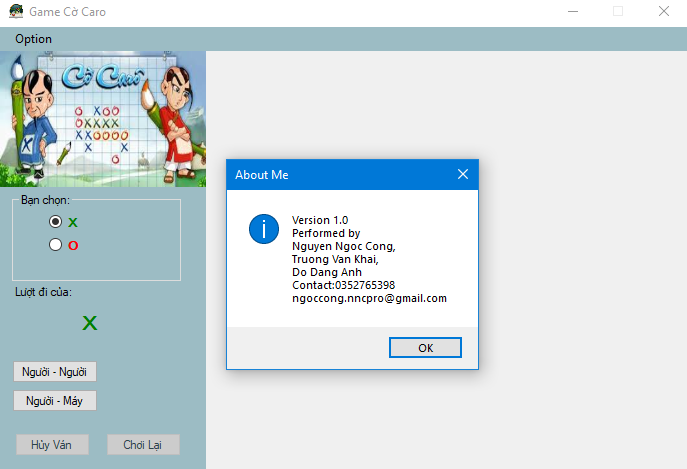
Sau khi kết thúc vòng lặp thì ô cờ (imax,jmax) chính là nước đi máy quyết định đánh.

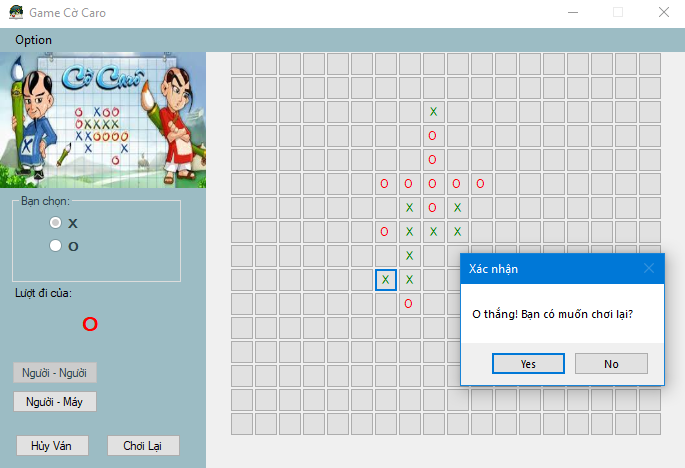
* **Cách duyệt tính điểm tấn công các đường**
* Xét hàng ngang:
* Tại ô đang xét, duyệt bên trái và bên phải của ô đó
* Đếm số quân đồng minh và tăng khoảng cách cho đến khi gặp quân đối phương thì thoát vòng duyệt
* Điểm tấn công = điểm của tổng các quân đồng minh – điểm của tổng các quân đối phương
* Nếu đối phương chặn hai đầu và khoảng cách giữa 2 đầu nhỏ hơn 5 => điểm tấn công = 0.
* Làm tương tự với các đường còn lại và với việc duyệt tính điểm phòng thủ các đường.
* **Quy tắc cho điểm**
* Gán điểm cho trạng thái theo kinh nghiệm cá nhân và các điều chỉnh khi test để mang lại kết quả chấp nhận được
* Điểm tấn công: { 0, 4, 25, 246, 7300, 16561, 59049 }
* Điểm phòng thủ: { 0, 3, 24, 243, 2197, 19773, 177957 }
* Chú thích: từ trái sang phải lần lượt là số các quân cờ đồng minh hoặc đối thủ gặp được trên đường duyệt
* Ví dụ: 0 có quân nào thì điểm là 0, có 1 quân thì điểm tấn công là 4,…

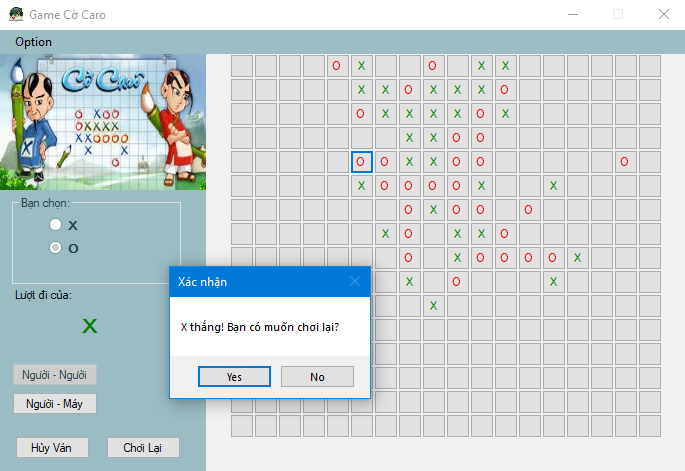
1. Các cải tiến áp dụng cho giải thuật

* Để tối ưu thời gian xử lý tính toán, áp dụng kỹ thuật cắt tỉa Alpha-Beta vào cắt tỉa bỏ những trạng thái dư thừa.
* Cắt tỉa bỏ các ô cờ mà tại đó xung quanh nó không có tính toán: các ô tại biên, tại các vùng trống quá rộng và cách xa khu vực đang tập trung nhiều nước đã đánh.

1. Một số kết quả







1. Kết Luận

Qua thời gian thực hành và tìm hiểu, nhóm đã xây dựng hoàn thành chương trình game. Tuy game đã có thể vận hành và có những kết quả khả quan nhưng bên cạnh đó vẫn tồn tại một số thiếu sót về độ hoàn thiện của thuật toán. Máy đôi lúc vẫn đưa ra những nước đánh không hiệu quả (trong các trường hợp đánh ở vùng biên ngoài bàn cờ, có những thế cờ khó quyết định tấn công hay phòng thủ, …)

Nhóm 1 rất mong nhận được sự nhận xét và góp ý của Cô và các bạn.

Xin chân thành cảm ơn!

1. Tài liệu tham khảo
2. Bài viết về giải thuật minimax trong Thư viện học liệu mở Việt Nam

Link: <https://voer.edu.vn/m/minimax/6c5ca780>

1. Bài viết về giải thuật minimax trong game cờ caro của thành viên @ttienqb (Diễn đàn Viblo Asia)

<https://viblo.asia/p/thuat-toan-minimax-ai-trong-game-APqzeaVVzVe>

1. Chương 2, Bài giảng Trí tuệ nhân tạo – PGS. TS. Nguyễn Thị Thủy