Báo Cáo Đồ Án Cơ Sở Trí Tuệ Nhân Tạo

PROJECT 01 - Lật Cờ

Sinh viên:

19127102 – Võ Hoàng Gia Bảo

19127406 – Ngô Huy Hoàng

19127457 – Nguyễn Tuấn Kiệt

Logo

Description automatically generated

Bộ môn Cơ sở trí tuệ nhân tạo

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP. HCM

Cơ sở trí tuệ nhân tạo-Project 01

**1,Nội dung và yêu cầu đồ án:**

**1,1 Lý Thuyết Adversarial Search**

**1,2 Mức độ hoàn thành về nội dung trò chơi**

**1,3 Phân tích đồ án**

**2, Cách thức biên dịch và chạy mã nguồn:**

**3,Ưu Nhược điểm của bot**

**3,1 Ưu điểm:**

**3,2 Nhược điểm:**

**3,3 Phương hướng cải tiến**

**3,4 Các phương pháp có thể ứng dụng mà thế giới hiện dang sử dụng**

Cơ sở trí tuệ nhân tạo-Project 01

1-

1,Nội dung và yêu cầu đồ án:

1,1 Lý thuyết Adversarial Search:

-**Adversarial search:** là tìm kiếm khi có một "kẻ thù" hoặc "đối thủ" thay đổi trạng thái của vấn đề từng bước theo hướng bạn không muốn. Ví dụ: Cờ vua, kinh doanh, buôn bán, chiến tranh. Bạn thay đổi trạng thái, nhưng sau đó bạn không kiểm soát trạng thái tiếp theo. Đối thủ sẽ thay đổi trạng thái tiếp theo theo cách: không thể đoán trước.

1,3 Phân tích đồ án AI:

-Môi trường lập trình: PyCharm  
-Ngôn ngữ lập trình: Python

-Thực hiện các chức năng cơ bản :

2, Cách thức biên dịch và chạy mã nguồn:

9

3,Ưu nhược điểm của bot :

3,1 Ưu điểm:

-Không sử dụng đệ quy

3,2 Nhược điểm:

-Không sử dụng cây tìm kiếm Alpha-Beta

3,3 Phương hướng cải tiến:

-Sử dụng cắt tỉa Alpha-Beta để giảm thời gian tìm kiếm bước đi tốt nhất

3,4 Các phương pháp có thể ứng dụng mà thế giới hiện dang sử dụng

- Năm 1997, Deep Blue của IBM lần đầu tiên đánh thắng nhà vô địch cờ Vua Kasparov. Deep Blue là môt chương trình đánh cờ được xây dựng trên cây tìm kiếm Alpha-Beta và có hàm lượng giá trạng thái tốt.

- Năm 2007, Crazy Stone của Remi Coulon đánh thắng Hiroshi Sagawa, vận động viên 5 đẳng nghiệp dư cờ Vây. Crazy Stone là một chương trình đánh cờ được xây dựng trên cây tìm kiếm Monte Carlo và có hàm lượng giá hành động tốt.

- Năm 2015, AlphaGo của Google lần đầu tiên đánh thắng nhà vô địch cờ Vây Lee Sedol. AlphaGo là một chương trình đánh cờ được xây dựng trên cây tìm kiếm Monte Carlo và kỹ thuật Deep Learning, kỹ thuật Deep Learning tìm kiếm được nhiều đặc trưng chính xác để xây dựng các hàm lượng giá hành động tốt.

**\*CÂY TÌM KIẾM**

Cây tìm kiếm đối kháng là một thành phần quan trọng trong chương trình đánh cờ. Tất cả cây tìm kiếm đối kháng đều được phát triển từ cây ban đầu Minimax, và cây tìm kiếm Monte Carlo là sự kết hợp hoàn hảo của cây Minimax và phương pháp giả lặp Monte Carlo.

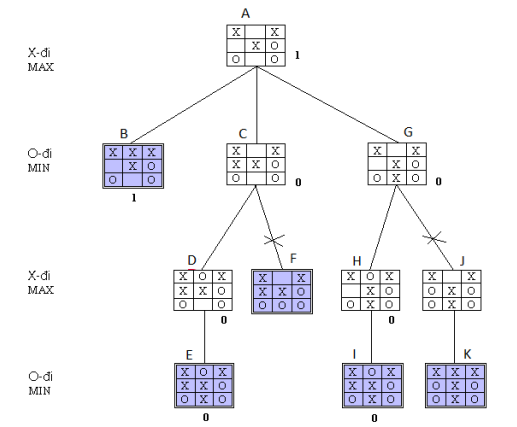
**\* CÂY TÌM KIẾM Alpha-Beta** Theo cây tìm kiếm Minimax, một trạng thái đích mong chờ của người chơi được hình thành từ một chuỗi các nước đi tối ưu. Chiến lược của người chơi Max luôn bị phụ thuộc vào các nước đi của người chơi Min và ngược lại. Lý do vì người chơi Max muốn cực đại hóa hàm lượng giá của mình, trong khi người chơi Min muốn cực tiểu hóa ưu thế của đối phương. Do Minimax là chiến lược tìm kiếm theo độ sâu, nên số trạng thái của cây tìm kiếm rất lớn nếu cây càng sâu.

Thuật toán Alpha-Beta là một cải tiến của thuật toán Minimax nhằm tỉa bớt nhánh của cây trò chơi, làm giảm số lượng trạng thái phải sinh (mà không ảnh hưởng đến sự đánh giá đỉnh đang xét) và lượng giá, do đó có thể tăng độ sâu của cây tìm kiếm. Như chúng ta đều biết, người chơi tính càng xa thì việc chọn nước đi sẽ chính xác, và độ sâu của cây tìm kiếm trong chương trình ứng với mức độ tính trước của người chơi. Cho nên chương trình tính càng sâu thì sẽ cho ra nước đi càng chính xác.

Thay vì tìm kiếm toàn bộ không gian đến một độ sâu lớp cố định, tìm kiếm Alpha-Beta thực hiện theo kiểu tìm kiếm sâu. Có hai giá trị, gọi là alpha và beta được tạo ra trong quá trình tìm kiếm, trong đó:

- Giá trị alpha liên quan với các nút Max và có khuynh hướng không bao giờ giảm.

- Giá trị beta liên quan đến các nút Min và có khuynh hướng không bao giờ tăng.



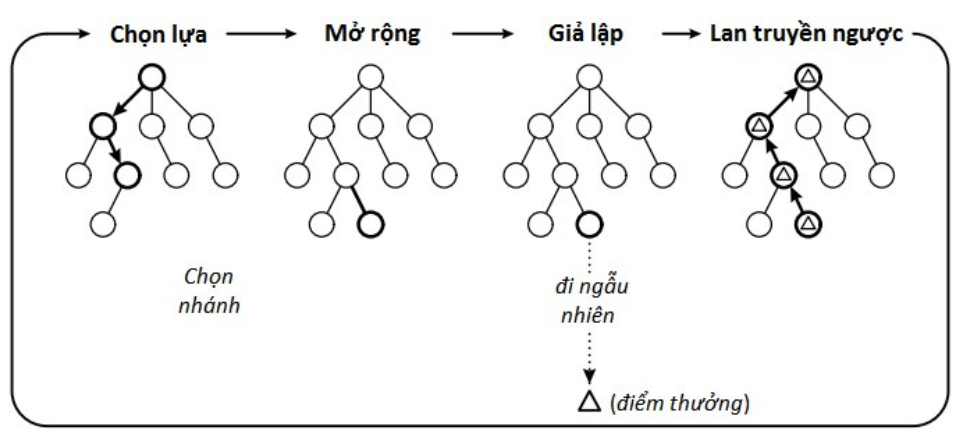
Cây tìm kiếm Alpha-Beta

**\* CÂY TÌM KIẾM Monte Carlo**

Các phương pháp Monte-Carlo hiện nay đang rất hiệu quả cho các chương trình cờ Vây. Năm 2006, Kocsis và Szepesvári [1, 3, 4] đề xuất thuật toán UCT (Upper Confidence Bounds for Tree) áp dụng cho cờ Vây. Hiện nay, thuật toán UCT và các biến thể của nó là các thuật toán chủ đạo cho các chương trình có sử dụng MCTS. MCTS là phương pháp tìm kiếm dựa theo lấy mẫu dùng giả lập ngẫu nhiên để ước lượng tỷ lệ thắng thua của một trạng thái bàn cờ nhằm tìm ra nước đi tốt nhất và để cân bằng giữa việc khám phá và khai thác của tất cả các nước đi. Điểm mấu chốt của MCTS so với các phương pháp tìm kiếm truyền thống như Alpha-Beta và A\* là nó không phụ thuộc vào tri thức đặc trưng của trò chơi, nói cách khác là không phụ thuộc vào hàm lượng giá trạng thái. Khi đó, MCTS có thể áp dụng vào nhiều trò chơi dạng không may rủi và thông tin trạng thái trò chơi rõ ràng sau mỗi lượt đi. Đối với các trò chơi mà khó xây đựng hàm lượng giá trạng thái tốt như cờ Vây thì việc áp dụng MCTS rất hiệu quả.

MCTS là một quá trình lặp đi lặp lại bốn bước trong một khoảng thời gian hữu hạn .Chọn lựa, từ một nút gốc (trạng thái bàn cờ hiện hành) cho đến nút lá, vì vậy sẽ có nhiều hướng đi được mang ra đánh giá. Mở rộng, thêm một nút con vào nút lá của hướng được chọn trong bước chọn lựa, việc mở rộng không thực hiện trừ khi kết thúc ván cờ tại nút lá. Giả lập, một ván cờ giả lập được chơi từ nút mở rộng, sau đó kết quả thắng thua của ván cờ giả lập sẽ được xác định. Lan truyền ngược, kết quả thắng thua sẽ được cập nhật cho tất cả các nút của hướng được chọn theo cách lan truyền ngược.

Cây trò chơi tăng trưởng sau mỗi lần lặp của MCTS, tăng trưởng rộng hơn và sâu hơn. Nước đi hứa hẹn là nút con nào có tỷ lệ thắng thua cao hơn được chọn trong giai đoạn chọn lựa, và rồi các cây con cũng tăng trưởng ngày càng rộng hơn và sâu hơn, và việc lấy mẫu ngày càng chính xác hơn. Sau khi kết thúc thời gian tìm kiếm, nút con nào được thăm nhiều nhất tại nút gốc sẽ được chọn để đi.

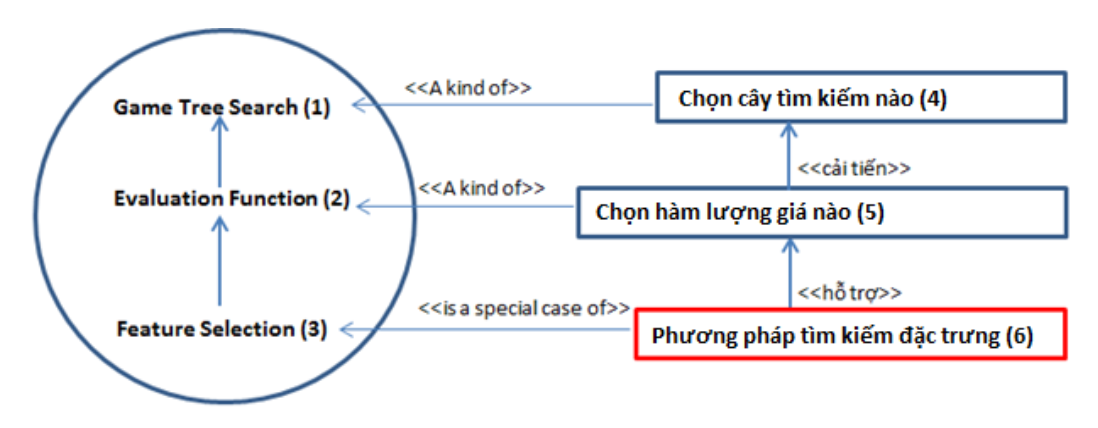


Cây tìm kiếm Monte Carlo

**\*KẾT LUẬN**

Như chúng ta đã biết, chương trình đánh cờ đều phải dựa trên cây tìmkiếm. Mỗi trò chơi là một cây tìm kiếm có độ lớn khác nhau, khác nhau về độ sâu của cây và độ phân nhánh. Vì vậy, không có một máy tính nào có thể vét cạn được những cây tìm kiếm vừa và lớn. Cho nên, cây tìm kiếm thì cần phải có hàm lượng giá hỗ trợ, chi phí xây dựng hàm lượng giá thì rất cao, và chủ yếu là chi phí tìm ra các đặc trưng. Như vậy chương trình đánh cờ mạnh yếu khác nhau tùy thuộc vào phương pháp xây dựng hàm lượng giá và phương pháp tìm kiếm đặc trưng sao cho tối ưu. Một mô hình được đúc kết qua nhiều chương trình đánh cờ Deep Blue, Crazy Stone,AlphaGo.

Phương pháp tìm kiếm đặc trưng hiện nay là học có giám sát từ những dữ liệu ván cờ có sẵn, việc học này được lặp đi lặp lại nhiều lần cho đến khi tìm được những đặc trưng tối ưu đủ để xây dựng một hàm lượng giá tốt. Có hai phương pháp tìm kiếm tự động phổ biến là Bradley-Terry Minirization-Maximization, và Deep Learning. Để kiểm chứng tính hiệu quả của các đặc trưng, phương pháp lấy trung bình logarit của các minh chứng (MLE-Mean Log Evidence) thường được sử dụng.



Các thành phần chính trong bài toán đánh cờ