

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# **Cơ sở trí tuệ nhân tạo**

Tìm kiếm đối kháng

**Nguyễn Ngọc Đức**

**2024**

# Trò chơi đối kháng

# Trò chơi

- Trò chơi là một trong những nghiên cứu chính cho việc phát triển các chương trình **thông minh**

- **Đặc điểm:**

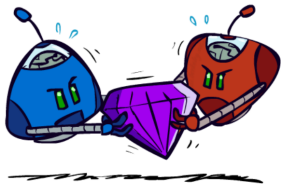
- 1 Môi trường đa nhân tố:** sự hiện diện của những người chơi với mục đích không đồng nhất
- 2 Bất định (uncertainty):** chiến lược di chuyển không rõ ràng



# Các loại trò chơi

■ Trò chơi có thể được phân loại dựa trên các yếu tố:

- 1 Hành động
- 2 Số lượng người chơi
- 3 Đối kháng?
- 4 Cụ thể (perfect information)?



# Trò chơi đối kháng 2 người I

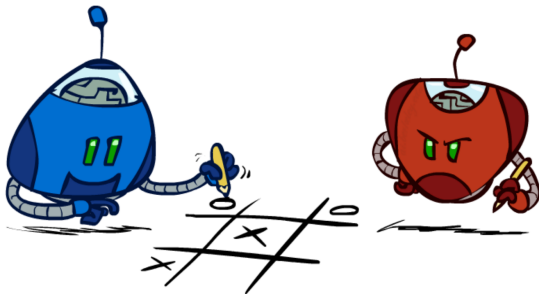
Tập người chơi  $Players = \{agent, opp\}$

- 1  $s_0$ : trạng thái bắt đầu
- 2  $Action(s)$ : Các hành động có thể thực hiện với trạng thái  $s$
- 3  $Succ(s, a)$ : trạng thái kết quả thực hiện hành động  $a$  ở trạng thái  $s$
- 4  $IsEnd(s)$ : kiểm tra trạng thái  $s$  có phải là trạng thái cuối (kết thúc trò chơi)
- 5  $Utility(s)$ : điểm của agent ở trạng thái  $s$
- 6  $Player(s) \in Players$ : người chơi điều khiển trạng thái  $s$

# Trò chơi đối kháng 2 người II

## ■ Có 2 đặc điểm chính:

- 1 Tất cả mục tiêu đều nằm ở trạng thái cuối cùng
- 2 Người chơi khác nhau điều khiển các trạng thái khác nhau



# Cờ vua



- $\text{Players} = \{\text{đen, trắng}\}$
- $\text{State}(s)$ : vị trí của các quân cờ
- $\text{Action}(s)$ : nước đi hợp lệ mà người chơi  $\text{Player}(s)$  có thể thực hiện
- $\text{IsEnd}(s)$ : kiểm tra trạng thái hiện tại là chiếu bí hay hòa
- $\text{Utility}(s)$ :  $+\infty$  nếu đen thắng,  $-\infty$  nếu trắng thắng, 0 nếu hòa

# Thành tựu hiện nay I

## ■ Cờ đam:

- **Độ phức tạp:**  $\approx 10^{18}$  nút
- 1950: chương trình đầu tiên
- 1991: Chinook đánh bại hoàn toàn nhà vô địch Marion Tinsley





# Thành tựu hiện nay II

## ■ Cờ vua:

- **Độ phức tạp:**  $b \approx 35, d \approx 100, 10^{154}$  nút
- 1997: Deep Blue đánh bại Gary Kasparov trong ván đấu 6 trận.



# Thành tựu hiện nay III

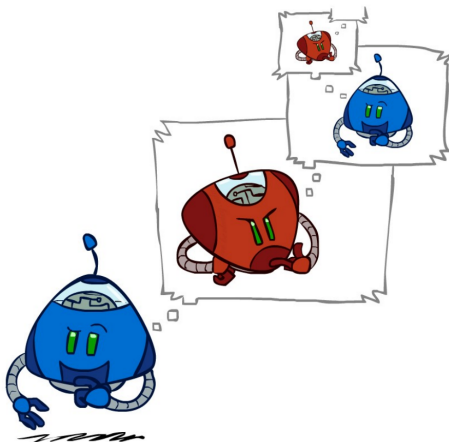
## ■ Cờ vây:

- **Độ phức tạp:**  $b \approx 361, d \approx 200, 10^{174}$  trạng thái bàn cờ
- Không thể đoán trước được ngay cả ở giai đoạn thu quan
- 2016: AlphaGo đánh bại kỳ thủ cửu đẳng Lee Sedol (4-1)



# Vấn đề

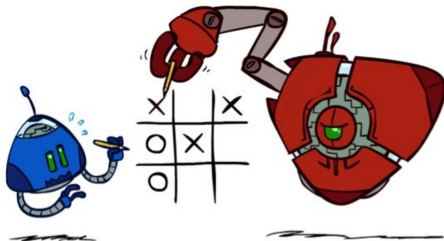
- Ta không biết rõ chiến lược của đối thủ
- Duy lý cá nhân
  - Mỗi người tham gia cuộc chơi sẽ cố gắng giành lợi ích tuyệt đối về bản thân
- Thuật toán Minimax



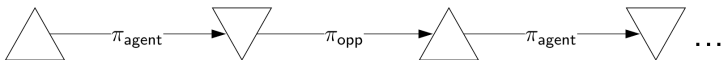
# Minimax

# Minimax I

- Nếu chúng ta có khả năng **đọc suy nghĩ**  $\Rightarrow$  **Expectimax**
- Tuy nhiên, trên thực tế chúng ta không biết chiến lược của đối thủ
- Giả định **trường hợp xấu nhất**: đối thủ làm mọi cách để giảm thiểu lợi ích



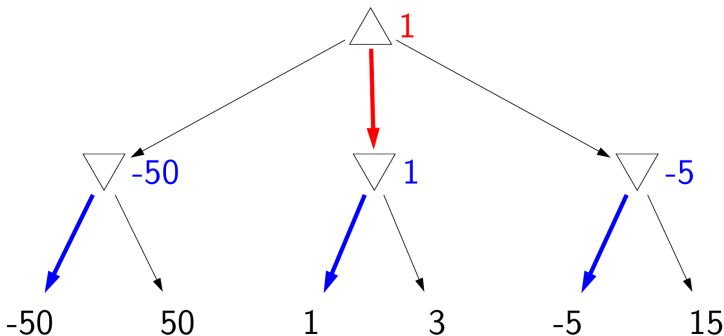
# Minimax II



- Lợi ích của đối thủ được thay thế bằng lợi ích tối thiểu

$$V_{\minmax}(s) = \begin{cases} Utility(s) & IsEnd(s) \\ \max_{a \in Actions(s)} V_{\minmax}(Succ(s, a)) & Player(s) = agent \\ \min_{a \in Action(s)} V_{\minmax}(Succ(s, a)) & Player(s) = opp \end{cases}$$

# Minimax III



$$V_{\min\max}(s_{\text{start}}) = 1$$

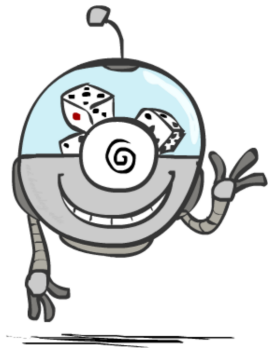
Hình 1: Minimax

# Expectiminimax



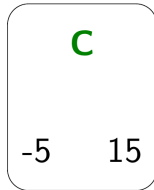
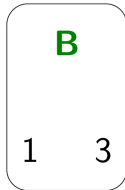
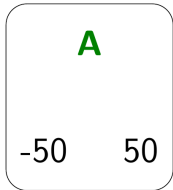
# Expectimax

- Nếu trò chơi có **yếu tố ngẫu nhiên**?
- Giả sử mỗi nút ngẫu nhiên ta đều biết được **phân phối** của các successor
- **Chiến lược bất định**



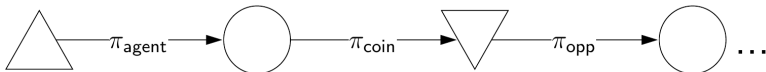
## Trò chơi 3 chiếc hộp

- Có 3 chiếc hộp, mỗi hộp chứa 2 con số.
- Bạn chọn một chiếc hộp sau đó tung đồng xu; nếu ra mặt ngửa, thay vì chọn hộp của bạn, mình sẽ chọn hộp bên trái và một con số nằm trong hộp đó
- Nhiệm vụ của bạn là phải tối đa con số mà mình chọn



# Expectiminimax I

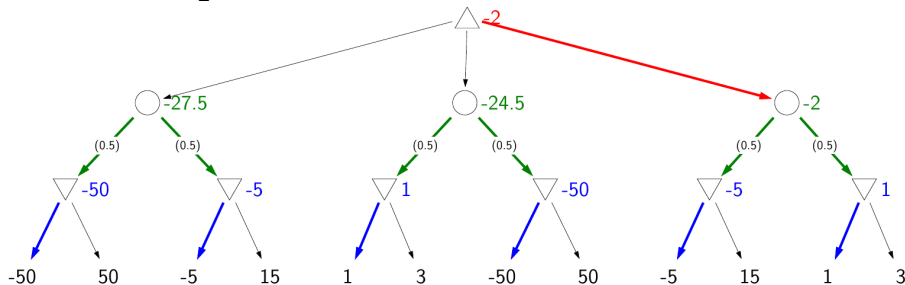
- Trò chơi trên có thể được mô hình bằng **expectiminimax**.



$$V_{\text{exptminmax}}(s) = \begin{cases} \text{Utility}(s) & \text{IsEnd}(s) \\ \max_{a \in \text{Actions}(s)} V_{\text{exptminmax}}(\text{Succ}(s, a)) & \text{Player}(s) = \text{agent} \\ \min_{a \in \text{Actions}(s)} V_{\text{exptminmax}}(\text{Succ}(s, a)) & \text{Player}(s) = \text{opp} \\ \sum_{a \in \text{Actions}(s)} \pi_{\text{coin}}(s, a) V_{\text{exptminmax}}(\text{Succ}(s, a)) & \text{Player}(s) = \text{coin} \end{cases}$$

# Expectiminimax II

$$\pi_{\text{coin}}(s, a) = \frac{1}{2} \text{ for } a \in \{0, 1\}$$



$$V_{\text{exptminimax}}(s_{\text{start}}) = -2$$

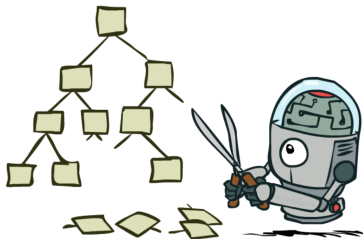
Hình 2: Expectiminimax

# Tỉa nhánh Alpha-Beta

# Tỉa nhánh Alpha-Beta I

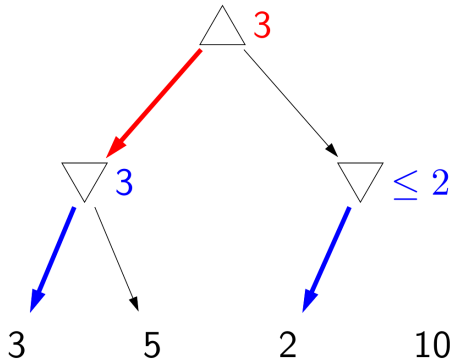
Ý tưởng chính:

- Giữ **ngưỡng trên** -  $\alpha$  của nút **min** và **ngưỡng dưới** -  $\beta$  của nút **max**
- Tỉa nhánh một nút nếu khoảng giá trị của nút đó không giao với nút cha ( $\alpha < \beta$ )



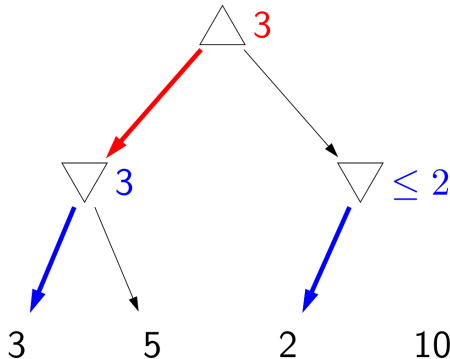
# Tỉa nhánh Alpha-Beta II

- Việc tỉa nhánh **không ảnh hưởng** đến giá trị cuối cùng của trò chơi
- Trong trường hợp tốt nhất độ phức tạp giảm xuống  $O(b^{m/2})$
- Tìm kiếm toàn bộ cây trò chơi cờ vua???



# Tỉa nhánh Alpha-Beta II

- Việc tỉa nhánh **không ảnh hưởng** đến giá trị cuối cùng của trò chơi
- Trong trường hợp tốt nhất độ phức tạp giảm xuống  $O(b^{m/2})$
- Tìm kiếm toàn bộ cây trò chơi cờ vua???



**vô vọng!!!**



# Hàm đánh giá

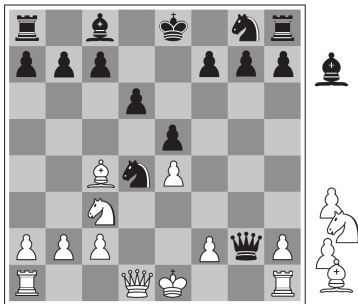
# Hàm đánh giá

- **Xấp xỉ** lợi ích của trạng thái hiện tại
- Thứ tự trạng thái cuối cùng đúng với lợi ích của trò chơi
  - Các trạng thái chiến thắng cần được đánh giá tốt hơn hòa và hòa tốt hơn thua
- Việc tính toán không quá lâu!
- Với các trạng thái không phải là trạng thái cuối, việc đánh giá cần phải tương quan chặt chẽ với tỷ lệ chiến thắng

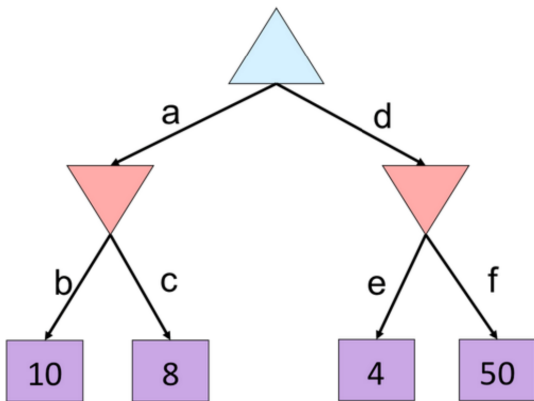
# Ví dụ

$Eval(s) = material + mobility +$   
 $king - safety + center - control$

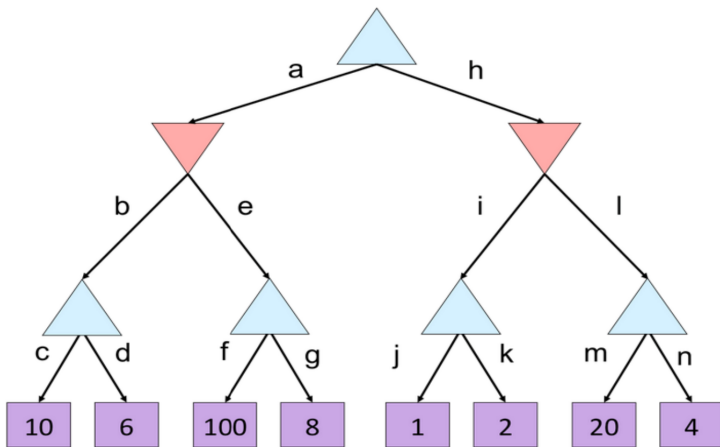
- $material = 10^{100}(K - K') +$   
 $9(Q - Q') + 5(R - R') + 3(B -$   
 $B' + N - N') + 1(P - P')$
- ...



# Bài tập I



# Bài tập II



# Tài liệu tham khảo

- [1] Bùi Tiến Lên, Bộ môn Khoa học máy tính  
Bài giảng môn Cơ sở trí tuệ nhân tạo
- [2] Michael Negnevitsky  
Artificial Intelligence: A Guide to Intelligent Systems (3rd Edition)
- [3] Dan Klein and Pieter Abbeel  
CS188: Introduction to Artificial Intelligence