**CHƯƠNG TRÌNH AI CHO MÁY TÍNH CHƠI GAME 8-PUZZLE VỚI NGƯỜI**

**Trần Nhật Thành 18110199**

**Lê Đức Thắng 18110201**

*Trường đại học Sư phạm Kỹ thuật TP.HCM*

**TÓM TẮT**

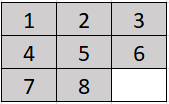
Hiện nay, nhu cầu giải trí của con người vô cùng đa dạng, người ta thường xuyên dùng những trò chơi nhỏ để giải trí, trong đó điển hình nhất là trò chơi 8-puzzle. Tuy nhiên để tăng thêm tính hấp dẫn cho trò chơi, nhóm chúng em đã thực hiện xây dựng chương trình 8-puzzle dành cho hai người chơi, trong đó việc quyết định lượt chơi dựa vào may mắn khi tung đồng xu. Có rất nhiều phương pháp để thực hiện chương trình này, tuy nhiên để chương trình có thể hoạt động tối ưu nhất, nhóm chúng em đã sử dụng thuật toán Expectimax. Thuật toán này tạo trí tuệ nhân tạo cho máy tính để máy tính có thể chơi cùng với người. Chương trình mà nhóm chúng em đã thực hiện thành công cho máy chơi cùng với người, có thể giải được 8-puzzle có độ sâu từ 14-15 bước đi. Bài báo cáo này mô tả lại chương trình mà nhóm chúng em đã xây dựng được bằng thuật toán Expectimax.

Từ khóa**:** 8-puzzle hai người chơi; 8-puzzle Expectimax; 8-puzzle trí tuệ nhân tạo

**1. ĐẶT VẤN ĐỀ**

8-puzzle là một thể loại game trí tuệ giống như trò chơi xếp hình. Trò chơi này gồm 1 bảng 3x3 được đánh số từ 1-8 và một ô trống được đánh số 0. Mục tiêu của người chơi là đưa trạng thái ban đầu về trạng thái đích bằng cách di chuyển các ô số

Với 8-puzzle dành cho hai ngươi chơi, hai người sẽ sử dụng một đồng xu mà một bảng 8-puzzle để chơi. Đồng xu được dùng để lựa chọn người thực hiện một bước đi trên bảng 8-puzzle. Nếu người chơi tung được mặt ngửa thì sẽ được lựa chọn bước đi, nếu mặt sấp thì sẽ đến lượt người kia tung dồng xu với xác suất của đồng xu là 0,5. Trò chơi kết thúc khi một trong hai người chơi có bước đi đến đích



Hình 1. 8-puzzle

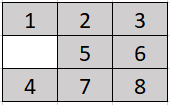
Các phần còn lại của báo cáo trình bày về phương pháp áp dụng thuật toán Expectimax để xây dựng AI cho máy tính chơi game với người chơi.

**2. PHƯƠNG PHÁP**

**2.1. Dữ liệu**

8-puzzle sử dụng dữ liệu đầu vào với không gian trạng thái là (3x3)! = 362880 với một chuỗi các số từ 0-8 gồm 4 bước đi là sang trái, sang phải, đi lên và đi xuống bằng cách di chuyển số 0. Có những trạng thái ban đầu người chơi sử dụng không thể đi đến trạng thái đích. Điều kiện để trạng thái ban đầu đi đến trạng thái đích là với mỗi ô trong 8-puzzle, xét theo chiều từ trái sang phải và từ trên xuống dưới, tổng số ô có giá trị nhỏ hơn nó theo sau phải là số chẵn.

Ví dụ với trạng thái ban đầu là {1, 2, 3, 0, 5, 6, 4, 7, 8}



Hình 2. 8-puzzle trạng thái đầu

thì ở vị trí ô số 5 có số 4 theo sau nhỏ hơn và ở vị trí số 6 cũng có số 4 theo sau là nhỏ hơn. Vậy tổng ô có giá trị nhỏ hơn theo sau nó bằng 2, nên trạng thái ban đầu này là hợp lệ.

**2.2. Xử lý tung đồng xu**

Để tung đồng xu, nhóm chúng em xây dựng một hàm sinh số ngẫu nhiên hai số là 1 và 2. Nếu số nhận được là 1 có nghĩa người chơi nhận được mặt ngửa và sẽ được chọn bước đi và ngược lại, nếu là 2 thì sẽ tới lượt người còn lại tung đồng xu.

**2.3. Xử lý dữ liệu đầu vào**

Để trò chơi được tiến hành thuận tiện nhất và không bị dừng trong quá trình khởi tạo, nhóm chúng em đã tiến hành kiểm tra hợp lệ của dữ liệu đầu vào. Với trạng thái đầu và trạng thái đích, thì các số nhập vào phải thõa mãn từ 0 đến 8 và không có số trùng nhau. Ngoài ra để chắc chắn trạng thái hiện tại có thể đi được đến trạng thái đích thì nhóm đã sử dụng thuật toán A\* tìm đường đi đến trạng thái đích. Thuật toán A\* mà nhóm đã xây dựng có thể giải được hơn 80% dữ liệu đầu vào.

**2.4 Thuật toán Expectimax**

Thuận toán Expectimax là phương pháp trọng tâm của bài toán để xây dựng AI cho máy tính. Dựa theo điều kiện xác suất để tìm bước đi tối ưu nhất trong mỗi trạng thái, Expectimax đã chia trạng thái ban đầu thành 2 node là Chance và Max. Chance sẽ trả về giá trị trung bình của các node tiếp theo và Max sẽ trả về giá trị tốt nhất mà trạng thái con có thể đạt được.

**function** Expectimax(node, goal, depth)

**if** node is a terminal node **or** depth = 0

**return** the heuristic value of node

**if** is Max

// Return value of maximum-valued child node

**let** a: = -∞

**foreach** child of node

a: = max(a, Expectimax(child, goal, depth -1))

**else if** Expectation event at node

// Return weighted average of all child nodes’ values

**let** a: = 0

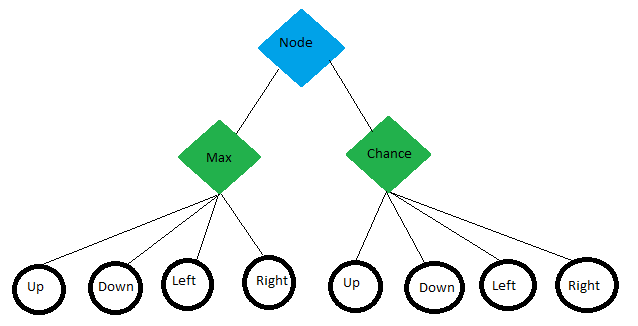
**foreach** child of node

a: = a+ (Probability[child] x Expectimax(child, goal, depth-1))

**return** a

Hình 3. Giải thuật Expectimax

Các trạng thái của thuật toán được xây dựng thành một node gồm có bốn node con là left, right, up và down. Với mỗi độ sâu, trạng thái sẽ lưu lại giá trị tốt nhất hoặc giá trị trung bình của các node con mà trạng thái đó có thể sinh ra. Tuy nhiên, để tránh việc sinh ra một node con nhiều lần thì nhóm chúng em đã tạo một danh sách để có thể lưu những node con đó. Những node con đã tồn tại trong danh sách thì không được sinh ra nữa.



Hình 4. Mô tả Expectimax

Để tránh việc vi phạm truy cập hay quá bộ nhớ, nhóm chung em đã giới hạn thuật toán với độ sâu là 16. Vì vậy với những trường hợp 8-puzzle có độ sâu lớn hơn 16 thì máy tính không thể giải ra hoàn toàn chính xác. Dưới đây là bảng thống kê máy tính thực hiện đơn phương trò chơi bằng thuật toán Expectimax

|  |  |  |  |
| --- | --- | --- | --- |
| Độ sâu | 0 - 10 | 10 - 15 | 15 - 20 |
| Mức độ hoàn thành | 100% | 100% | 15% |

Bảng 1. Tổng kết mức độ hoàn hành

Nhóm chúng em đã cho chương trình tiến hành chạy độc lập để tìm đường đi tới đích bằng thuật toán Expectimax, nhờ vào những bước đi đó mà máy có thể chọn ra đường đi tốt nhất và tiến hành di chuyển. Chương trình hoàn thành hầu hết được các trạng thái đầu vào, tuy nhiên ở một vài trạng thái, chương trình sẽ không thể tìm thấy lời giải do bước đi quá lớn.

**3. KẾT QUẢ**

Bảng 2. Thống kê mức độ hoàn thành khi chơi

Khi tiến hành người chơi với máy, thì chương trình có thể hoàn thành tốt hơn so với chạy độc lập vì sẽ có những bước đi mà người thực hiện tốt hơn, do đó mức độ hoàn thành của bài toán cũng được cải thiện.

**4. KẾT LUẬN**

**4.1. Hạn chế:**

Vì bị giới hạn về độ sâu do bị lỗi bộ nhớ nên thuật toán chỉ chạy được với số bước đi nhất định, do đó với trạng thái có bước đi lớn thì chương trình sẽ không thể tìm ra được đường đi tốt nhất. Ngoài ra, thuật toán Expectimax cũng không thể tận dụng tối ưu lợi thế trước đối thủ.

**4.2. Hướng phát triển:**

Để phát triển chương trình, nhóm sẽ cố gắng khắc phục nhược điểm lớn nhất của chương trình là giới hạn về độ sâu để chương trình có thể thực thi được toàn bộ các trạng thái. Chương trình được thực hiện trên Console nên giao diện có phần hạn chế, do đó trong tương lai nhóm chúng em sẽ cố gắng cải thiện phần giao diện để thu hơn với người chơi và không gây cảm giác nhàm chán bất tiện. Ngoài ra để tăng thêm độ khó, chương trình sẽ có thêm phần tự xáo trộn trạng thái ban đầu tùy vào độ khó mà người chơi muốn. Từ đó tạo ra một chương trình AI cho máy tính hoàn thiện nhất.

**TÀI LIỆU THAM KHẢO**

[1] Trần Thị Hà An, Phan Tuấn Anh; *Bài toán 8-puzzle;* Trường Đại học Sư Phạm Hà Nội, 2013

[2] AftaabZia; *Expectimax Algorithm in Game Theory*

[3] Wikipedia; *Expectiminimax*; 2006

[4] [redblobgames@gmail.com](mailto:redblobgames@gmail.com); *A\*’s Use of the Heuristic; 2010*