

ĐẠI HỌC UEH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ
KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH



ĐỒ ÁN MÔN HỌC
ĐỀ TÀI:
XÂY DỰNG GAME TETRIS DỰA TRÊN
GIẢI THUẬT DI TRUYỀN

Học phần: Trí Tuệ Nhân Tạo

Nhóm Sinh Viên:

1. Mai Trần Mỹ Uyên
2. Nguyễn Quỳnh Khánh Hà
3. Ngô Thị Huyền

TP Hồ Chí Minh, ngày 29 tháng 5 năm 2022

MỤC LỤC

MỤC LỤC.....	1
CHƯƠNG 1. TỔNG QUAN.....	1
<i>1.1. Giới Thiệu Về Trò Chơi Tetris</i>	<i>1</i>
<i>1.2. Phát Biểu Bài Toán.....</i>	<i>1</i>
<i>1.3. Một Số Hướng Tiếp Cận Giải Quyết Bài Toán.....</i>	<i>1</i>
CHƯƠNG 2. GIẢI THUẬT DI TRUYỀN.....	3
<i>2.1. Giới Thiệu Về Giải Thuật Di Truyền</i>	<i>3</i>
<i>2.2. Ứng Dụng Giải Thuật Di Truyền Cho Bài Toán Tetris</i>	<i>3</i>
2.2.1. Định hình trò chơi.....	3
2.2.2. Các đặc trưng đánh giá.....	3
2.2.3: Giải thích thuật toán.....	7
CHƯƠNG 3: CÁC KẾT QUẢ THỰC NGHIỆM	7
<i>3.1. Các Tình Huống.....</i>	<i>8</i>
<i>3.2. Phân Tích và Đánh Giá</i>	<i>8</i>
Chương 4: KẾT LUẬN.....	9
<i>4.1. Các Kết Quả Đạt Được.....</i>	<i>9</i>
<i>4.2. Những Hạn Chế và Hướng Phát Triển</i>	<i>10</i>
TÀI LIỆU THAM KHẢO	12
PHỤ LỤC.....	13

CHƯƠNG 1. TỔNG QUAN

1.1. Giới Thiệu Về Trò Chơi Tetris

- **Giới thiệu**

Tetris là một trò chơi điện tử đầu tiên được thiết kế và phát triển bởi nhà khoa học máy tính người Liên Xô Alexey Pajitnov. Nó được phát hành bởi một số công ty, nổi bật nhất là trong cuộc chiến tranh giành quyền sở hữu ứng dụng của trò chơi vào cuối những năm 1980. Sau một khoảng thời gian phát hành phiên bản kinh điển của Nintendo, quyền sở hữu trí tuệ được hoàn lại cho Pajitnov vào năm 1996, người đồng sáng lập công ty The tetris với Henk Roger để quản lý việc cấp phép tetris.

- **Hướng dẫn chơi**

Trò chơi được chơi trên một bàn cờ có hai chiều khởi tạo rỗng. Bàn cờ dần dần đầy lên khi các khối gạch có hình dạng khác nhau, được gọi là Tetromino, rơi từ trên xuống và rơi từng mảnh một. Người chơi có thể kiểm soát cách mỗi Tetromino tiếp đất bằng cách xoay nó và di chuyển nó theo chiều ngang, sang trái hoặc sang phải, bất kì số lần nào, Tetromino sẽ rơi từng hàng một cho đến khi một trong các ô của nó nằm ngay trên đầu của những viên gạch khác hoặc trên đáy bàn cờ. Khi toàn bộ hàng trở nên đầy, toàn bộ hàng này sẽ bị xoá, tạo thêm không gian trên bàn cờ và ghi điểm.

Trò chơi kết thúc khi không còn khoảng trống trên đầu bàn cờ cho lượt Tetromino tiếp theo. Người chơi có thể trì hoãn trạng thái kết thúc này càng lâu thì điểm của họ sẽ càng cao.

1.2. Phát Biểu Bài Toán

Đồ án này trước tiên mô tả sơ lược về trò chơi Tetris, hướng tiếp cận trò chơi. Sau đó sơ lược về giải thuật di truyền và ứng dụng của nó cho bài toán xây dựng Tetris, phân tích các tình huống từ đó ra đánh giá, liệt kê những kết quả đạt được, những hạn chế và phát triển của thuật toán di truyền trong việc xây dựng trò chơi tetris.

1.3. Một Số Hướng Tiếp Cận Giải Quyết Bài Toán

Trước khi xây dựng tác tử cho Tetris, chúng ta hãy bắt đầu bằng cách xác định đâu là hướng có thể tiếp cận của thuật toán cho trò chơi này. Nói chung, tác tử của trò chơi này quyết định hành động sẽ chọn tiếp theo ở bất kỳ trạng thái nhất định nào của trò chơi dựa trên phần thưởng mà họ sẽ nhận được sau khi thực hiện hành động đó. Một khi tác tử chọn, trong số những hành động có thể thực hiện tiếp theo, tác tử sẽ trả lại kết quả tốt nhất. Tuy nhiên, rất khó và đôi khi không thể đánh giá giá trị của một hành động, và một

cách giải quyết vấn đề này là đánh giá giá trị của trạng thái mà hành động sẽ dẫn đến. Cụ thể hơn, điều này đạt được bằng cách sử dụng một hàm đánh giá gán các giá trị số cho các trạng thái trò chơi. Và tác tử sẽ lựa chọn hành động dẫn đến trạng thái có giá trị tốt nhất. Đây cũng có thể được coi là một chiến lược tham lam, trong đó tác nhân đưa ra quyết định được đánh giá cao nhất ở mỗi bước. Tetris được ước tính là có 7×2^{20} trạng thái. Với số lượng trạng thái khổng lồ này, phương pháp chung là ước lượng một hàm giá trị, hoặc đưa về một chính sách sử dụng một tập hợp các đặc trưng mô tả trạng thái hiện tại hoặc cặp trạng thái hiện tại - hành động. Trong bài báo này, các tác tử Tetris sử dụng những hàm đánh giá, đánh giá trạng thái bàn cờ hiện tại bằng cách gán các giá trị số. Với mỗi một Tetromino Ứng dụng thuật giải di truyền vào game xếp hình Tetris xuất hiện, tác tử sẽ ra quyết định (dịch và xoay) của mảnh đó sẽ đưa ra trạng thái bàn cờ tốt nhất khi bị rơi. Tiếp theo, tác tử này sẽ mô phỏng và đánh giá tất cả các trạng thái của bàn cờ từ tất cả các nước đi có thể có của Tetromino, rồi chọn nước đi dẫn đến bàn cờ có giá trị tốt nhất. Cách tiếp cận phổ biến nhất đối với Tetris là phát triển một hàm đánh giá tuyến tính, trong đó mỗi vị trí có thể có của khối gạch này được đánh giá để chọn vị trí có giá trị cao nhất.

CHƯƠNG 2. GIẢI THUẬT DI TRUYỀN

2.1. Giới Thiệu Về Giải Thuật Di Truyền

Giải thuật di truyền là kỹ thuật phỏng theo quá trình thích nghi tiến hoá của các quần thể sinh học dựa trên học thuyết Darwin. GA là phương pháp tìm kiếm tối ưu ngẫu nhiên bằng cách mô phỏng theo sự tiến hoá của con người hay của sinh vật. Tư tưởng của thuật toán di truyền là mô phỏng các hiện tượng tự nhiên, là kế thừa và đấu tranh sinh tồn. GA thuộc lớp các giải thuật xuất sắc nhưng lại rất khác các giải thuật ngẫu nhiên vì chúng kết hợp vì chúng kết hợp các phần tử tìm kiếm trực tiếp và ngẫu nhiên.

2.2. Ứng Dụng Giải Thuật Di Truyền Cho Bài Toán Tetris

2.2.1. Định hình trò chơi

Mặc dù phiên bản tiêu chuẩn của Tetris là trò chơi kinh điển, nhưng vẫn tồn tại rất nhiều biến thể của trò chơi. Để có sự rõ ràng cho sự trực quan và phương pháp tiếp cận, chúng tôi xây dựng tác tử AI từ nền tảng có các quy tắc như sau:

- Bàn cờ Tetris có kích thước hàng và cột lần lượt là 10 và 20, trong đó là có 2 hàng bị ẩn ở trên.
- Tất cả các Tetrominoes (các mảnh Tetris) sẽ bắt đầu ở giữa 2 dòng trên cùng.
- Có tất cả bảy loại Tetromino: 'I', 'O', 'J', 'L', 'S', 'Z', 'T'.
- Tác tử sẽ biết trước Tetromino rơi xuống tiếp theo (look-ahead).

2.2.2. Các đặc trưng đánh giá

Bốn đặc trưng đánh giá

Cũng giống như những người chơi đã chơi cho đến nay. Mục tiêu của chúng tôi là xóa càng nhiều đường càng tốt, nói cách khác là hoãn trạng thái kết thúc nhất có thể ..

Để đạt được mục tiêu này, tác tử của chúng tôi sẽ quyết định nước đi tốt nhất cho một Tetromino nhất định bằng cách suy tính tất cả các động tác có thể (xoay và vị trí). Nó tính điểm số (điểm đánh giá) cho mỗi nước đi có thể (look-ahead) và chọn một nước có điểm tốt nhất làm nước đi tiếp theo.

Điểm số cho mỗi bước di chuyển được tính bằng cách đánh giá trạng thái bàn cờ mà bước di chuyển sẽ dẫn đến. Kết quả này dựa trên bốn đặc trưng: tổng chiều cao của trạng

thái, số đường đã xóa, hố và độ gập ghềnh. Mỗi đặc trưng này tác tử sẽ cân nhắc, hoặc là cố gắng giảm thiểu, hoặc là tối đa hóa.

a. Tổng chiều cao của trạng thái

Đặc trưng này cho chúng ta biết lưới "cao" như thế nào. Để tính toán chiều cao tổng hợp, chúng tôi lấy tổng chiều cao của mỗi cột (khoảng cách từ ô cao nhất trong mỗi cột đến cuối bàn cờ).

Tất nhiên chúng tôi muốn giảm thiểu giá trị này, vì chiều cao tổng hợp thấp hơn có nghĩa là chúng tôi có thể thả nhiều Tetromino hơn vào bàn cờ trước khi chạm vào đỉnh của nó.

4 4 0 3 6 5 5 4 5 4



Tổng chiều cao của trạng thái này là 48

b. Số hàng đã xóa

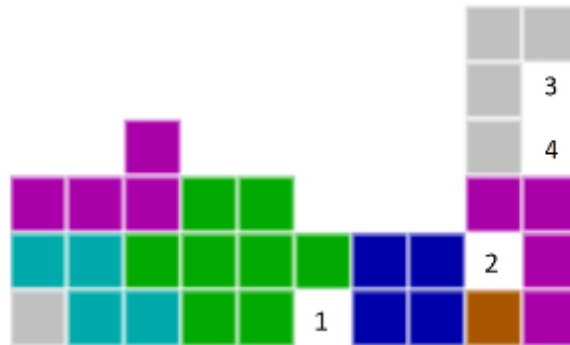
Đây có lẽ là đặc trưng dễ hiểu nhất trong số bốn đặc trưng. Nó chỉ đơn giản là số hàng đã hoàn thành trong một bàn cờ. Chúng tôi muốn tối đa hóa số hàng đã hoàn thành này, do mục tiêu của AI là xóa càng nhiều hàng càng tốt và việc xóa các hàng này sẽ cho chúng ta nhiều khoảng trống hơn để có nhiều khối gạch hơn.



Số hàng đã xóa là 2

c. Số hố

Như đã đề cập ở trên, một hố được định nghĩa là một khoảng trống sao cho có ít nhất một ô trong cùng một cột phía trên nó. Hố thì khó xóa hơn, do chúng ta phải xóa tất cả các đường phía trên trước khi chúng tôi có thể chạm tới hố và lấp đầy nó. Do đó, chúng tôi sẽ phải giảm thiểu số các hố này.



Số hố hiện tại là 2

d. Độ gập ghềnh

Tổng chiều cao của trạng thái, xét trường hợp sau, các Tetromino xếp trên bàn cờ của chúng ta và chừa ra một cái giếng sâu (được đánh số 0). Điều này thật khó chịu, bởi vì: Những giếng này có thể được lấp đầy một cách dễ dàng, và thật khủng khiếp khi giếng bị chặn bởi những khối gạch. Và sau đó, rất khó để có thể lấp đầy. Để khái quát hóa ý tưởng về giếng, chúng tôi định nghĩa một đặc trưng được đặt tên là độ gập ghềnh. Độ gập ghềnh cho chúng ta biết sự thay đổi chiều cao của từng cột trong bàn cờ. Nó được tính bằng cách tính tổng các giá trị tuyệt đối của hiệu chiều cao hai cột liền kề.

Từ các chỉ số chiều cao của các cột trên bàn cờ của Hình 3, dưới đây là cách tính độ gập ghềnh:

$$|4-4+4-0|+|4-5|+|5-4|=14$$

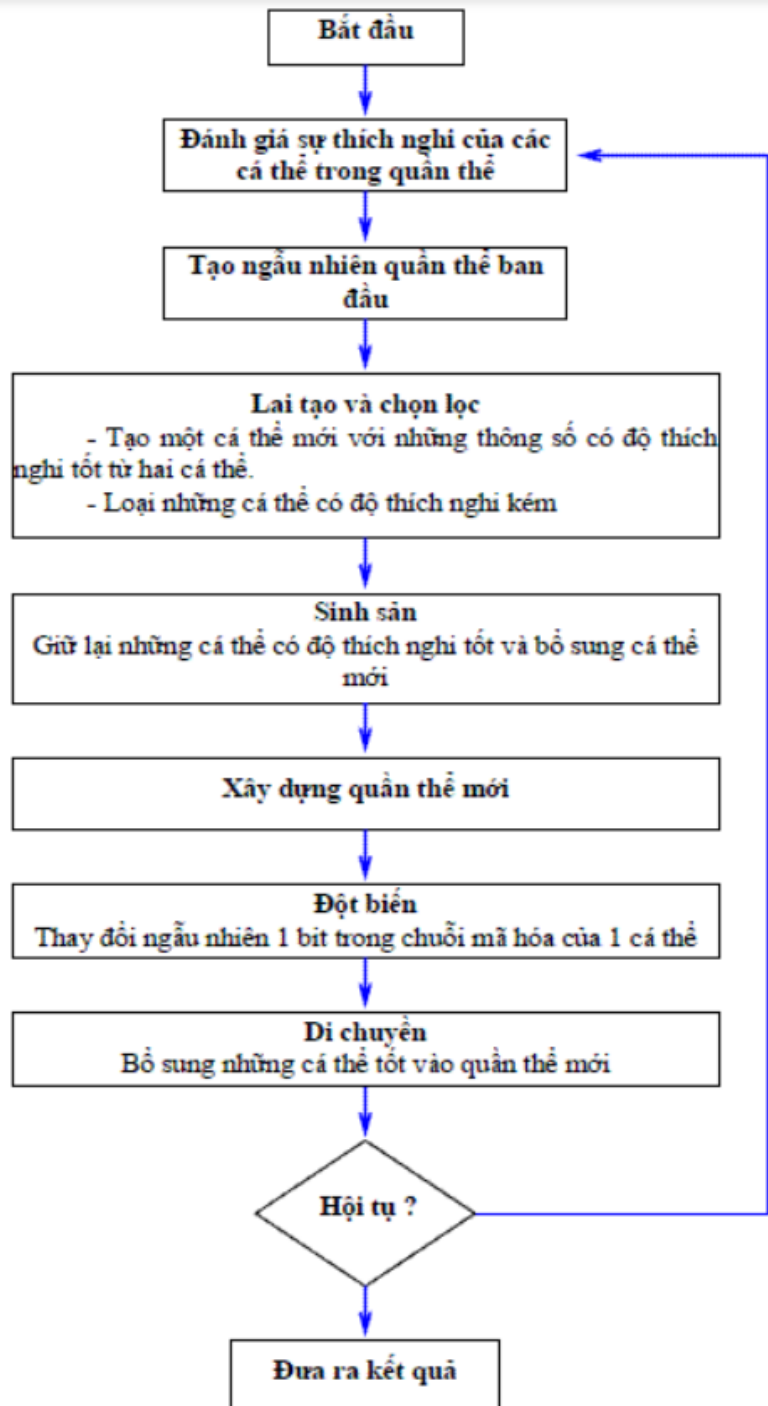
Mục tiêu của chúng tôi là giảm thiểu sự xuất hiện và độ sâu của những cái giếng này. Vì vậy, đây là giá trị cần được giảm thiểu.

e. Kết hợp các đặc trưng đánh giá

Bây giờ chúng tôi sẽ tính điểm của một trạng thái bàn cờ, bằng cách lấy sự kết hợp tuyến tính của bốn đặc trưng đã đề cập ở trên. Hàm đánh giá của tác tử có công thức như sau:

Hàm đánh giá = $a \cdot (\text{tổng chiều cao của trạng thái}) + b \cdot (\text{số hàng đã xóa}) + c \cdot (\text{số hố}) + d \cdot (\text{độ gập ghềnh})$

Trong đó a,b,c,d là các tham số ràng buộc. Như đã phân tích chức năng và nhiệm vụ của bốn đặc trưng, để tối đa hóa khả năng của hàm đánh giá này, chúng tôi đặt a,c,d âm và b dương.



Sơ đồ thuật toán

2.2.3: Giải thích thuật toán

- Xây dựng (khởi tạo) quần thể (population) ban đầu: Chúng ta có thể khởi tạo quần thể bằng cách sinh ra các agent chứa các string ngẫu nhiên. Giả sử quần thể của chúng ta có 60 agents
- Đánh giá quần thể
- Đánh giá(cho điểm)mỗi giả thiết. Trong bài toán này với mỗi tetra hiện tại sẽ đánh giá điểm di chuyển trên mỗi tetra tiếp theo để xác định tetra nào rơi xuống bằng cách thử 2 phép tính theo cấp số cộng và cấp số nhân . Sau đó tính toán tất cả vị trí thả có thể thực hiện, xem xét từng di chuyển để tính toán từng vị trí thả có thể xảy ra.
- Tìm chiều cao lớn nhất của trạng thái, đảm bảo rằng tetra không va chạm với bất kỳ thứ gì trước giá trị chiều cao lớn nhất để tìm đúng điểm liên hệ. Tính toán điểm theo sự sắp xếp của lưới nhị phân. Đúng (1) nếu cột đã được lấp đầy, sai (0) nếu cột rỗng. Tìm chiều cao của ô cao nhất trong mỗi cột của lưới bàn cờ Tetris
- Tái tổ hợp: Các hàng đã được lấp đầy, số hồ, sự khác biệt về chiều cao từ cột này sang cột khác. Trả về trọng lượng tái tổ hợp
- Đột biến: Random trọng lượng đột biến theo tỷ lệ đột biến. Sử dụng trị tuyệt đối đường cong phân phối chuẩn sử dụng biến đổi box muller
- Thay thế các cá thể yếu kém: Với những cá thể yếu có ít điểm đánh giá nhất trong quần thể sẽ bị xóa và thay thế với các cá thể con. Toàn bộ quá trình lặp lại cho đến khi quần thể có thể chấp nhận được

CHƯƠNG 3: CÁC KẾT QUẢ THỰC NGHIỆM

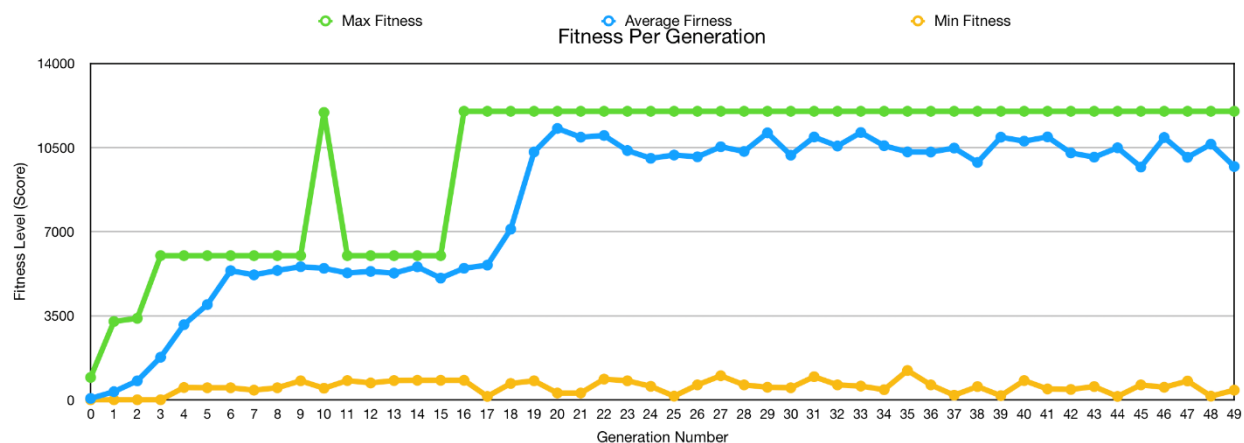
3.1. Các Tình Huống

Bài toán Tetris có 2 tình huống xảy ra:

_ Khi 1 hàng được lấp đầy, hàng ấy sẽ bị xóa, tạo thêm không gian trên bàn cờ và ghi điểm (lines cleared)

_ Khi các khối gạch chạm đến đỉnh bàn cờ, một lượt chơi sẽ kết thúc và khởi tạo một lượt chơi mới (generation)

3.2. Phân Tích và Đánh Giá



Giải thuật di truyền bắt đầu bằng cách khởi tạo quần thể bộ gen bằng cách tạo ngẫu nhiên các bộ gen cho một kích thước quần thể được xác định trước. Sau khi được tạo, các bộ gen sẽ được kiểm tra và đưa ra một mức độ thể chất nhất định dựa trên hiệu suất của chúng. Quá trình chọn lọc, trao đổi chéo được thực hiện với nhiều lần lặp lại qua nhiều thế hệ.

Sau khi tính toán bộ tính năng tối ưu, thuật toán di truyền thử nghiệm trong 50 thế hệ với kích thước quần thể 60. Biểu đồ trên tóm tắt số điểm fitness của quần thể của 50 thế hệ, mô tả xu hướng tổng thể của quần thể nói chung cải thiện qua mỗi thế hệ. Ta có thể thấy thuật toán đã đạt được fitness level lên đến hơn 12000 điểm. Nếu tiếp tục nhiều thế hệ có thể dẫn đến tăng nhẹ hiệu suất, nhưng những kết quả trên đã đủ để chứng minh tính hiệu quả của thuật toán di truyền này khi áp dụng cho bài toán Tetris.

Chương 4: KẾT LUẬN

4.1. Các Kết Quả Đạt Được

Qua quá trình nghiên cứu, chúng tôi vận dụng giải thuật di truyền vào việc giải quyết một số bài toán tối ưu, đặc biệt là các bài toán tối ưu có không gian tìm kiếm lớn.

Nhóm đã vận dụng giải thuật GA để xây dựng hệ thống game Tetris tự động hóa. Số điểm (score) đạt được sẽ đánh giá độ hiệu quả của thuật toán này, nó tương ứng với những hàng đã xóa. Và nếu thuật toán của chúng tôi đủ tốt, nó có thể tiến tới vô hạn. Trên thực tế, chúng tôi đã triển khai thuật toán này trên game Tetris có bàn cờ kích thước 20x10 với số hàng xóa được trong một lượt chơi lên đến hơn 12.000 dòng



Sau khi thử nghiệm với hơn 100 thể hệ, chúng tôi đạt được số hàng xóa được trung bình là khoảng 7000 dòng/thế hệ

```
----- Starting Generation 111 -----
Lines cleared: 38775 28922 26458 23431 19055 15966 15926 15476 14429 13799 12864 12560 12177 10774
7647 7309 7267 7072 7024 6878 6714 6398 5492 5363 5034 4687 4170 4105 2899 2826 2741 2633 2579 2508
3 1500 1357 1211 1106 969 756 654 638 513 303 198 197 56 27 20 0 0
Lines cleared average: 7006.35
Most lines cleared: 38775 28922 26458 23431 19055 15966 15926 15476 14429 13799
Most lines cleared average: 21223.70
Most cleared row filled weights: [1.69, 1.24, 0.97, 0.69, 0.43, 0.53, 0.42, 0.31, 0.37, 0.74, 1.69]
Most cleared hole height weights: [1.97, 2.45, 2.31, 3.17, 2.09]
Most cleared column diff weights: [0.11, 0.28, 0.43, 0.70, 0.81]
----- Starting Generation 112 -----
```

4.2. Những Hạn Chế và Hướng Phát Triển

a) Những hạn chế

Vấn đề lớn nhất của nhóm chúng tôi là đã tùy tiện chọn ra rất nhiều tham số. Chúng có thể hoạt động tốt hoặc có thể không, nhưng chúng tôi không thực sự biết liệu có những giá trị tốt hơn cho chúng hay không. Do đó, chúng tôi đã cố gắng tối ưu hóa các thông số này bằng Giải thuật di truyền.

Giải thuật di truyền là thuật toán có tên gọi được lấy cảm hứng từ Thuyết tiến hóa của Mendel. Các thành tố của thuật toán này được trình bày vắn tắt như sau:

- Một *nhiệm sắc thể* biểu thị một giải pháp khả thi cho vấn đề dưới dạng một chuỗi. Hàm fitness lấy một nhiệm sắc thể làm đầu vào và trả về giá trị fitness
- Một *quần thể* gồm rất nhiều các nhiệm sắc thể
- Một phương pháp *selection* xác định cách các cá thể mẹ được chọn để nhân giống từ quần thể.
- Một toán tử *crossover* xác định cách các cá thể mẹ kết hợp để tạo ra cá thể con.
- Toán tử đột biến xác định cách đột biến của các cá thể.

Chúng tôi nhận thấy việc huấn luyện và hiệu năng của tác tử AI được xem là thành công. Đây là những vấn đề còn lại mà chúng tôi cần cải thiện trong những công trình sau này.

Về hàm đánh giá heuristic:

Đặc trưng tổng chiều cao trạng thái là hệ số rất lớn, áp đảo mọi đặc trưng khác và có thể gây ra sự dư thừa. Mặc dù đúng là tổng chiều áp dụng cho mọi khối gạch trên bàn cờ, nhưng nó không thực sự hoạt động như thế mà chỉ ảnh hưởng đến những khối Tetromino rơi hiện tại. Lý do là, phần còn lại của bàn cờ - mọi thứ trừ khối rơi hiện tại - sẽ không đổi cho dù nó đi đến đâu. Giống như việc bạn bỏ phiếu cho tất cả mọi người, bạn thực sự không bỏ phiếu cho ai cả vì phiếu bầu của bạn không ảnh hưởng đến kết quả.

Đặc trưng các hàng đã xóa cũng có một chút sai sót. Trong khi b (tham số thứ hai) có trọng số dương đối với việc xóa một hàng, nó vẫn xóa các đường bất cứ khi nào có thể: và lần nữa ràng buộc trở lại hệ số chiều cao. Việc xóa toàn bộ một hàng các khối - và thậm chí là xóa nhiều hàng đó sẽ tác động rất lớn đến hệ số chiều cao.

Về Giải thuật di truyền:

Hàm fitness đôi lúc sẽ gây rắc rối. Vào thời điểm AI đã và đang xóa vài nghìn hàng trên bàn cờ 22 cột, điều duy nhất khiến AI rơi vào bế tắc là một chuỗi các khối S và Z khó xử lý. Thực tế điều này cũng gây khó dễ cho con người - và trong bất kỳ bộ tham số được

tạo ngẫu nhiên nào, bạn cũng sẽ nhận được một loạt các Tetromino không may mắn như thế. Vì vậy, ở các thế hệ sau, nếu chỉ đơn giản mô phỏng một không gian Tetris 22 cột thì khá là tệ trong việc phân loại AI tốt và AI xuất sắc.

Selection cũng gặp vấn đề. Sau mười thế hệ, toàn bộ quần thể ít nhiều đều có các giá trị giống nhau, chỉ có một số thay đổi nhỏ - hơi mỉa mai vì tình huống này cũng là tình huống mà thuật toán *Tournament selection* được lập trình để ngăn chặn

Mặc dù Giải thuật di truyền có thể hội tụ về tối ưu cục bộ khá nhanh chóng, tạo ra một giải pháp phù hợp, nhưng rất khó để nó đạt được một giải pháp tối ưu toàn cầu - một giải pháp tốt nhất của bài toán. Việc thay đổi bất kỳ một giá trị nào trong bộ tham số sẽ gây hại nghiêm trọng cho cá thể, nhưng việc thay đổi hai hoặc nhiều giá trị đồng thời theo một cách nhất định sẽ làm cho nó tốt hơn. Đây là một nhược điểm đối với các thuật toán di truyền nói chung.

b) Hướng phát triển:

Giải thuật di truyền đã đề xuất phần lớn vẫn phụ thuộc vào sự “may mắn” để tìm ra kết quả. Do đó để tìm ra được kết quả tối ưu với các trường hợp số dòng xóa được lớn là rất hạn chế. Trong tương lai, nhóm sẽ tìm hiểu và cài đặt các giải thuật heuristic giúp tìm ra được lời giải tối ưu với đồ thị có kích thước lớn hơn, cùng với đó là các cải tiến và giải thuật GA như quần thể động, đa quần thể tương tác,... Bên cạnh đó có thể nghiên cứu thêm các mô hình học máy khác để giải bài toán Tetris như Reinforcement Learning (học tăng cường) với các giải thuật deep Q-learning để cải tiến bài toán hiệu quả hơn, thậm chí có thể chạy liên tục với số dòng xóa được là vô hạn

TÀI LIỆU THAM KHẢO

https://github.com/danielchang2002/tetris_ai

<https://github.com/SajjadLab/Tetris-ai>

<https://github.com/camilanovaes/tetris-ai>

<https://github.com/jgardner8/genetic-tetris-ai>

<https://github.com/fthomasmorel/Tetris-AI>

<https://github.com/johnliu0/tetro>

<https://github.com/irgiob/tetris-ai-9000>

<https://github.com/thyo9470/Genetic-Tetris>

<https://github.com/JLMadsen/TetrisAI>

https://github.com/Godsinred/Tetris_AI

PHỤ LỤC

Mã nguồn: [Link](#)

PHÂN CÔNG CÔNG VIỆC

Họ và tên	Công việc
Ngô Thị Huyền	Chương 1: Tổng quan "tetris.py" (Tetris game) Mục 4.1. Các kết quả đạt được
Mai Trần Mỹ Uyên	Chương 2: Giải thuật di truyền "ai.py" (AI logic) Cài đặt data đầu vào
Nguyễn Quỳnh Khánh Hà	Chương 3 : Kết quả thực nghiệm Mục 4.2: Những hạn chế và hướng phát triển "tetromino.py" (Main file) "tetromino.py" (Tetromino logic)