

# Thuật toán di truyền, Feedforward Neural Network và góc nhìn lập trình

Nguyễn Đức Bảo Lâm

Ngày 11 tháng 3 năm 2025

## Mục lục

<b>1</b>	<b>Người lập trình có thể lấy ý tưởng từ đâu?</b>	<b>3</b>
<b>2</b>	<b>Vậy thì thuật toán là gì?</b>	<b>3</b>
<b>3</b>	<b>Thuật toán trong góc nhìn sinh học</b>	<b>4</b>
3.0.1	Di truyền, ADN và nhiễm sắc thể . . . . .	4
<b>4</b>	<b>Tìm hiểu về thuật toán di truyền và trả lời câu hỏi trên</b>	<b>5</b>
4.1	Nền tảng Genetic Algorithm (GA) . . . . .	6
4.2	Triển khai ở góc nhìn thuật toán . . . . .	7
4.3	Câu trả lời của thuật toán . . . . .	7
4.4	Mã giả thuật toán . . . . .	8
<b>5</b>	<b>Mạng Feedforward Neural Network (FNN) và mạng Artificial Neural Network (ANN)</b>	<b>9</b>
5.1	Lịch sử phát triển của AI . . . . .	9
5.2	Ý nghĩa của AI trong bài viết này . . . . .	9
5.3	Mối quan hệ giữa AI và thuật toán . . . . .	9
5.4	AI, thuật toán và FNN . . . . .	10
5.5	FNN . . . . .	11
<b>6</b>	<b>Tính kết nối giữa GA và FNN</b>	<b>12</b>
6.1	Ở phạm vi con người . . . . .	12
6.2	Thử khai thác dấu ? . . . . .	13

<b>7</b>	<b>Ứng dụng GA, FNN vào tìm kiếm chiến lược sinh tồn của sinh vật</b>	<b>14</b>
7.1	Mô tả vấn đề . . . . .	14
7.2	Biện luận một số thành phần trong mô tả vấn đề . . . . .	14
7.3	Ý tưởng triển khai . . . . .	15
7.4	Chi tiết một số thành phần thuật toán GA . . . . .	15
7.5	Kết quả thực thi thuật toán . . . . .	16
<b>8</b>	<b>Kết luận</b>	<b>17</b>
<b>9</b>	<b>Tham khảo thêm</b>	<b>18</b>
<b>10</b>	<b>Minh hoạ bổ sung</b>	<b>20</b>

## **Tóm tắt nội dung**

Bài viết này sẽ tập trung vào trả lời câu hỏi Người lập trình có thể lấy ý tưởng từ đâu? Qua đó, khẳng định vai trò và thiết lập lại vị trí của toán học trong thuật toán và lập trình. Ngoài ra, bài viết cũng sẽ hướng vào ứng dụng thuật toán di truyền, mạng Neural Network cho tìm kiếm một chiến lược sinh tồn phù hợp cho môi trường biến động (một vấn đề ứng dụng).

## **1 Người lập trình có thể lấy ý tưởng từ đâu?**

Lập trình, một từ đơn giản nhưng lại có đúc kết nhiều phạm trù. Lập trình một mặt là làm việc máy tính, mặt khác, lập trình đôi khi lại hiểu là lập trình cuộc sống. Điều cốt lõi góp phần mở rộng ý nghĩa của lập trình là từ suy luận của con người và ảnh hưởng của thời đại.

Quay trở lại với câu hỏi, trước tiên hãy thiết lập phạm vi và ý nghĩa của câu hỏi đã nêu.

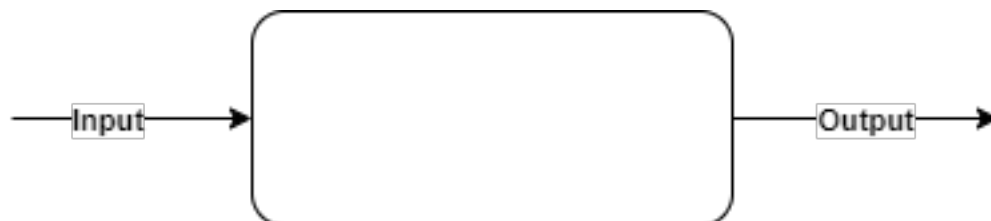
- Câu hỏi sẽ được trả lời và sẽ định trả lời trong một số lĩnh vực liên quan như triết học, sinh học, khoa học thần kinh.
- Nếu câu trả lời mà bài đưa ra là hợp lý thì việc trả lời câu hỏi này sẽ đưa đến góc nhìn mới về vai trò của toán trong lập trình.
- Dựa trên việc thiết lập câu trả lời, bài viết cũng đề cập đến một số thuật toán, qua đó mở rộng thêm kho vũ khí tích hợp cho giải quyết vấn đề.
- Ngoài ra câu trả lời cũng đem đến góc nhìn của lập trình trong bối cảnh AI hiện tại.

## **2 Vậy thì thuật toán là gì?**

Phần mở đầu của bài là một câu hỏi phụ về thuật toán - một câu hỏi về định nghĩa. Bài viết này sẽ khai thác một số khía cạnh của thuật toán. Qua đó kết nối với các chủ thể đã đề cập như "thuật toán di truyền (Genetic Algorithm)", "Mạng Feedforward Neural Network (Mạng thần kinh lan truyền thẳng)".

Ở các chủ thể khác nhau, việc tinh chỉnh hay hiểu khái niệm cốt lõi là một điều cần thiết để có thể áp dụng. Quay lại với định nghĩa, theo [23] và [25],

thuật toán (Algorithm) được hiểu là **một tập hữu hạn các bước được xác định rõ ràng** nhằm hướng dẫn cho máy tính giải quyết vấn đề/bài toán cụ thể nào đó.



Hình 1: Mô hình thuật toán chung, khối hình chữ nhật đen chưa định nghĩa gì, tượng trưng cho một chiếc hộp đen (blackbox). Hiểu rằng, thuật toán là biến đầu vào (Input) thành đầu ra (Output) tương ứng.

Như vậy khi tiến hành nêu ra định nghĩa thuật toán ở trên, ta nắm được một số thông tin mà sẽ được dùng để áp dụng kết nối chủ thể như "**hướng dẫn cho máy tính**", "**giải quyết vấn đề/bài toán**"

### 3 Thuật toán trong góc nhìn sinh học

Khi tìm hiểu về chủ thể này, sự thâm nhập của thuật toán vào lĩnh vực sinh học nói chung là cực kỳ thú vị. Trong mục này, bài viết xem thuật toán dưới góc nhìn di truyền nên những kiến thức có liên quan đến di truyền cũng sẽ được đề cập.

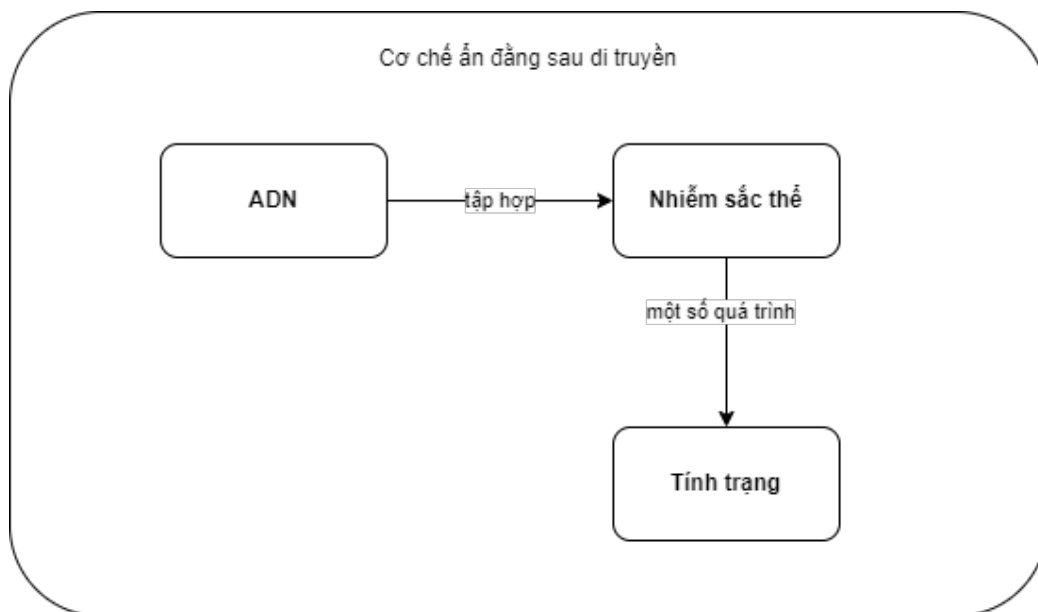
#### 3.0.1 Di truyền, ADN và nhiễm sắc thể

Di truyền là hiện tượng truyền đạt các tính trạng của các (bố mẹ, tổ tiên) cho các thế hệ con cháu theo [3].

ADN là phân tử mang thông tin di truyền quy định mọi hoạt động sống theo [1]

Nhiễm sắc thể là bào quan chính chứa bộ gen của sinh vật, là cấu trúc quy định sự hình thành protein, có vai trò quyết định trong di truyền theo [18]. Có thể hiểu nhiễm sắc thể (NST) là một tập hợp của các gen.

Như vậy, mối quan hệ giữa các khái niệm có thể được mô tả theo sơ đồ sau:



Hình 2: Sơ đồ mối quan hệ giữa các khái niệm. Về cơ bản ADN/NST qua một số giai đoạn trung gian để có thể biểu thị nên tính trạng. Và quá trình đó có thể hiểu là cơ chế biểu thị của di truyền

Việc đề cập đến kiến thức di truyền nhằm khẳng định vai trò của di truyền là mang **tính hướng dẫn**. Điều này tương tự với nội dung đã đề cập trong định nghĩa của thuật toán 2.

Về bản chất của áp dụng trong mục này là nhận ra ADN/NST là có tính hướng dẫn tương tự như thuật toán. Sự suy luận này xuất phát từ việc đúc kết những đối tượng mang tính tương tự thường sẽ có một số tính chất chung.

Vậy câu hỏi mà mục này đặt ra là: "**Mình khai thác tính hướng dẫn mà sinh học nói chung mang lại cho thuật toán như thế nào?**"

## 4 Tìm hiểu về thuật toán di truyền và trả lời câu hỏi trên

Việc khai thác câu hỏi trên chính là việc cần đưa tính hướng dẫn đáp ứng tính chất thứ hai mà định nghĩa thuật toán ( 2) đã nêu. Đó là tính chất giải quyết vấn đề/bài toán.

Trong thực tế, nhiều thuật toán đã dựa trên các đối tượng sinh học để giải quyết yêu cầu đặt ra. Một trong số những thuật toán có thể kể đến như:

- Thuật toán tối ưu đàn kiến (Ant Colony Optimization) [6]. Một thuật toán dựa trên hành vi tìm đường của quần thể kiến trong tự nhiên.
- Thuật toán tối ưu bầy hạt ((Particle Swarm Optimization) [24]. Thuật toán cũng dựa trên hành vi di chuyển và tìm kiếm thức ăn của các đàn chim và đàn cá.
- Thuật toán sinh học miễn dịch (Artificial Immune System) [14] [19]. Một thuật toán lấy ý tưởng từ hệ thống miễn dịch của con người.
- và còn nhiều thuật toán khác nữa...

Về cơ bản, những dẫn chứng em cung cấp ở trên là một phần minh chứng cho câu hỏi lớn mà bài viết của em muốn khai thác. Tuy nhiên, hệ quả của câu hỏi đó mới là quan trọng. Ở mục này em xin bàn luận thêm sâu hơn về thuật toán di truyền xem như câu trả lời cho câu hỏi làm sao để áp dụng ý tưởng đó vào lập trình.

## 4.1 Nền tảng Genetic Algorithm (GA)

Thuật toán di truyền là một thuật toán lấy ý tưởng từ quá trình tiến hoá trong tự nhiên. Chính xác hơn, đây là thuật toán mô phỏng quá trình ấy và là một thuật toán tối ưu.

Quá trình tiến hoá trong tự nhiên còn hiểu là quá trình chọn lọc tự nhiên [17]. Đây là quá trình mà những cá thể mang khả năng thích nghi (fitness) cao với môi trường sẽ có nhiều khả năng tồn tại và duy trì nòi giống để tạo ra thế hệ sau. Kết quả của quá trình này là qua nhiều thế hệ, thế hệ sau có khả năng cao sẽ mang những gen thích nghi tốt với môi trường.

Diễn giải thêm cho quá trình trên, **môi trường** và **di truyền** là hai yếu tố chi phối chủ đạo.

Phân tích thêm cho quá trình, ngoài hai yếu tố chi phối trên, quá trình này hoạt động trên một quần thể (một tập hợp các cá thể).

Như vậy, từ ý tưởng thô sơ là quá trình chọn lọc tự nhiên. Sau quá trình phân tích để xác định các yếu tố, bước tiếp theo là mình cần chuyển các yếu tố ấy thành mã lập trình dưới góc nhìn của **toán học**.

## 4.2 Triển khai ở góc nhìn thuật toán

Do bản chất của máy tính là tính toán nên mình phải cần nhìn vấn đề dưới góc nhìn của một người lập trình. Ở góc nhìn này, ta sẽ trả lời cho khía cạnh thứ hai của định nghĩa thuật toán (giải quyết vấn đề).

Máy tính còn mang tính tất định. Những yếu tố mô tả ở trên còn tương đối mơ hồ. Như vậy, sự rõ ràng của những khái niệm trên phải được xác lập. Dưới đây là những câu hỏi dùng để làm rõ thêm thuật toán di truyền.

- Làm sao phản ánh được môi trường vào trong di truyền?
- Di truyền còn tương đối mơ hồ, làm sao để đảm bảo sự rõ ràng ở đây?
- Việc cài đặt quần thể là cài đặt như thế nào?
- Phản ánh giữa thuật toán và di truyền sẽ ra sao?

## 4.3 Câu trả lời của thuật toán

Trong GA, việc phản ánh môi trường vào trong di truyền được trả lời thông qua hàm fitness (hàm đánh giá độ thích nghi).

Ngoài phản ánh qua hàm fitness, môi trường còn phản ánh thông qua Selection. Selection hoạt động dựa trên giá trị thích nghi. Những cá thể có điểm thích nghi cao sẽ có khả năng giữ được phần gen của mình và truyền cho thế hệ sau. Selection còn có thể coi là một toán tử trong môi trường.

Di truyền ở mức độ chi tiết hơn ngoài 2 còn có thêm một số yếu tố sau để có thể triển khai thuật toán:

- Lai ghép (Crossover). Hiểu là trao đổi thông tin di truyền ở giữa hai cá thể. Còn có thể hiểu thêm, việc tiến hành lai ghép là việc yêu cầu hai cá thể trong quần thể thực hiện sinh sản để cho ra cá thể con. Cá thể con sẽ mang một phần/toàn phần thông tin di truyền từ hai cá thể tổ tiên (có thể xem như cha và mẹ).
- Đột biến (Mutation). Trong di truyền, quá trình đột biến xảy ra tương đối. Ở quá trình chọn lọc tự nhiên, đột biến đem đến sự đa dạng cho nguồn gen di truyền.
- ADN/NST. Đây chính là mã di truyền. Đối với giải quyết vấn đề/bài toán thì ADN/NST chính là **lời giải tiềm năng**.

Câu hỏi về cài đặt quần thể. Trong lập trình, điều này có thể được mô phỏng dưới dạng một tập hợp các mã di truyền. Ở góc độ lập trình, điều này đồng nghĩa với việc mình đang duy trì một tập lời giải.

Phản ánh giữa thuật toán và di truyền là mối quan hệ phản ánh đến từ việc nhìn các mã di truyền là lời giải cho vấn đề được nêu. Điều này dẫn đến hệ quả các phép toán như Crossover, Mutation sẽ đóng vai trò như việc trao đổi lời giải và đột ngột phát sinh ý tưởng mới.

## 4.4 Mã giả thuật toán

Qua những mục được đề cập trên, mã giả của thuật toán sẽ như sau:

---

**Algorithm 1** Thuật toán di truyền (GA)

---

- 1: Khởi tạo quần thể  $P$
  - 2: Đánh giá độ thích nghi cho từng cá thể trong  $P$
  - 3: **while** Thỏa điều kiện dừng **do**
  - 4:   Chọn cá thể trong  $P$  dựa trên fitness
  - 5:   Tiến hành lai ghép (Crossover) để tạo con cháu
  - 6:   Thêm đột biến (Mutation)
  - 7:   Tính thích nghi (dùng hàm Fitness) cho con cháu
  - 8:   Chọn (Selection) thế hệ tiếp theo từ  $P$  và con cháu
  - 9: **end while**
  - 10: Trả về cá thể tốt nhất
- 

Thuật toán di truyền chỉ cố định ở mô hình thao tác (mã giả minh hoạ ở trên). Còn các chi tiết còn lại như hàm Fitness, cách chọn thế hệ, phương pháp lai ghép hay cách đột biến, chúng ta có thể hoàn toàn linh động. Tùy vào từng trường hợp mà sẽ dùng phương pháp khác nhau.

Và ngoài ra, thuật toán di truyền cũng là một phần của lập trình tiến hoá (Evolution Programming) [21] cùng với một số thuật toán khác như Chiến lược tiến hoá (Evolution Strategy) [7], thuật toán di truyền vi phân (Differential Evolution) [4] [5], ...

Mục nội dung này được viết là dựa trên các nguồn [12] [20] [9] [10] [15]



## 5 Mạng Feedforward Neural Network (FNN) và mạng Artificial Neural Network (ANN)

Tìm hiểu trên về thuật toán di truyền đã phần nào mở ra và chứng minh cho câu hỏi chủ đề qua lĩnh vực sinh học. Ở mục này, câu hỏi sẽ tiếp tục được trả lời nhưng sẽ dưới góc nhìn của lĩnh vực triết học và khoa học thần kinh.

Mạng Feedforward Neural Network còn được hiểu là mạng lan truyền thẳng. Đây là một mạng lưới thần kinh nhân tạo. Để hiểu hơn về mạng này, hãy khảo sát sơ qua về lịch sử phát triển của AI thông qua tìm hiểu một số khuynh hướng phát triển của Deep Learning.

### 5.1 Lịch sử phát triển của AI

Đây là phần nội dung được tổng hợp qua [11, Chương 1, mục 1.2.1, trang 12–26]

AI mang trong mình một lịch sử phát triển dài hạn và qua nhiều khuynh hướng phát triển khác nhau. Những khuynh hướng phát triển ấy là kết quả của sự giao thoa nhiều lĩnh vực có thể kể đến như bio-learning (học tập lấy cảm hứng từ tự nhiên), cybernetic (điều khiển học) và connectionism (hiểu là sự kết nối).

Những đóng góp khác nhau từ các lĩnh vực đã góp phần và làm tiền đề quan trọng cho sự nổi dậy của AI trong giai đoạn xã hội hiện nay. Và chính thực, ví như các mô hình ngôn ngữ lớn (Large Language Models) đều dựa trên những ý tưởng cốt lõi và thêm du nhập của sự chú ý (Attention [22]).

### 5.2 Ý nghĩa của AI trong bài viết này

Trong quá trình tìm tòi về Trí tuệ nhân tạo, em nhận ra tính kết nối và tổng hợp của nhiều lĩnh vực ở trong AI. Song song với điều đó, AI cũng là một thuật toán và do đó việc tiến hành ngâm cứu thêm về chủ thể này cũng sẽ góp phần trả lời cho câu hỏi chủ đề.

### 5.3 Mối quan hệ giữa AI và thuật toán

Khẳng định AI là một **biểu hiện của thuật toán**, là một khẳng định hợp lí.

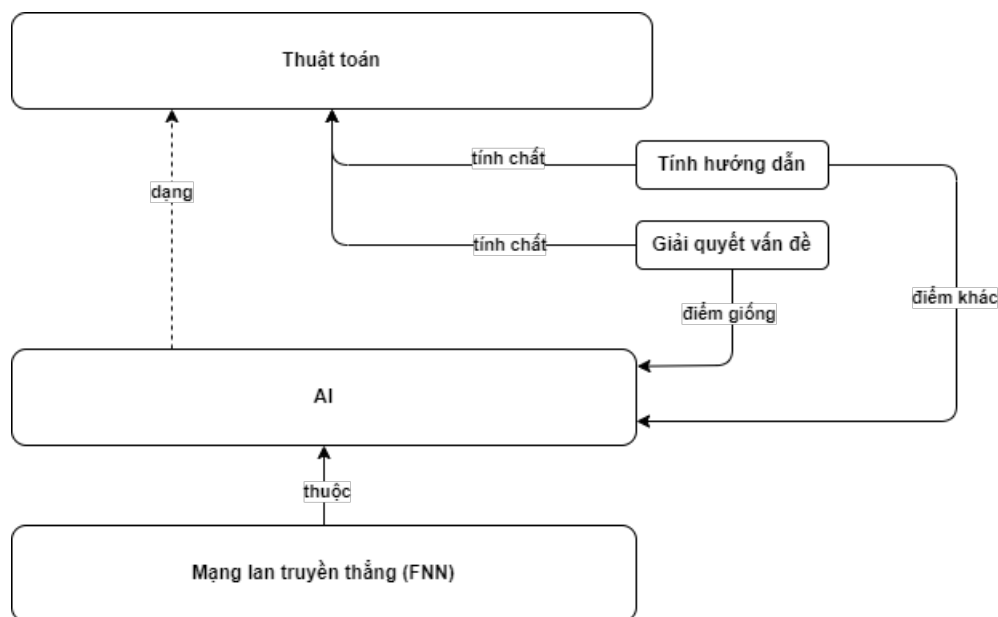
Hãy xét đến đối tượng thực hiện định nghĩa tập hữu hạn các bước. Nếu lấy con người làm trung tâm đối chiếu thì sẽ phát sinh một số câu hỏi sau:

- Nếu đối tượng thực hiện định nghĩa là con người thì sao?
- Nếu đối tượng thực hiện không là con người?

Tiến hành trả lời cho hai câu hỏi mang tính bổ trợ nhau ở trên, ta nhận ra AI là một dạng thuật toán nhưng việc thực hiện định nghĩa các bước giải quyết (tính hướng dẫn) phụ thuộc rất ít vào con người.

## 5.4 AI, thuật toán và FNN

Khảo qua mối quan hệ kế thừa của các đối tượng như AI, thuật toán và FNN. Ta có sơ đồ sau:



Hình 3: Mối quan hệ giữa AI, thuật toán và mạng lan truyền thẳng (FNN). AI giống thuật toán là đều dùng để giải quyết vấn đề nhưng khác nhau ở bước hướng dẫn. Mặt khác FNN là một con của AI, AI có những thuộc tính trên thì mạng FNN cũng sẽ có.

Như vậy thông qua sơ đồ trên, ta biết được FNN sẽ có những tính chất

mà AI có. Nhờ vậy, tìm hiểu thêm về mạng này cũng sẽ góp phần hiểu thêm về thuật toán.

## 5.5 FNN

Để cho câu trả lời cho câu hỏi chủ đề được trọn vẹn. Hãy tìm hiểu sơ qua về mạng FNN.

FNN là một cấu trúc cơ bản và quan trọng trong học sâu (Deep Learning). Dựa trên tiền đề này mà nhiều mạng với kiến trúc tiên tiến hơn được ra đời. Ngoài ra tính chất của FNN là thông tin chỉ lan truyền theo một chiều duy nhất xuyên suốt và không có vòng lặp.

FNN là một dạng của mạng thần kinh nhân tạo (Artificial Neural Network). Tất có nghĩa, FNN là một tập hợp các neuron và sự kết nối giữa chúng.

Mặt khác, do mang trong mình tính chất của một thuật toán nên để có thể diễn giải sang mã lập trình được thì cần thông qua một cây cầu mang tên **toán học**. Trong bối cảnh này, toán học dùng để mô phỏng hành vi của một đơn vị neuron và hành vi của một tập các neuron được kết nối với nhau. Về cụ thể:

- một đơn vị neuron có hành vi: nhận tín hiệu từ nhiều nguồn, tiến hành tổng hợp và đưa ra kết quả. Ở mặt này có thể dùng hàm số để mô phỏng hành vi. Cụ thể hơn, trong trường hợp này là hàm phi tuyến với công thức:

$$f(x) = \sigma(w_0 + \sum_{i=1}^n w_i * x_i) \quad (1)$$

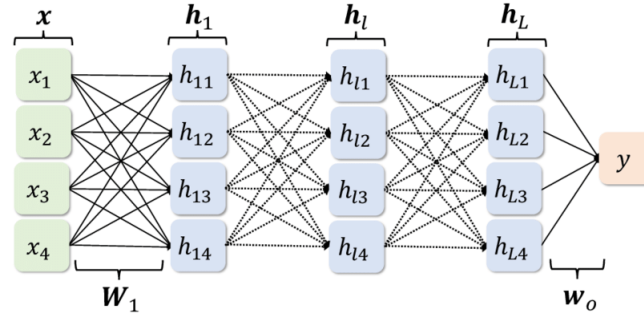
Hệ số  $w_0$  là hệ số đền bù (bias) và  $w_i$  là các trọng số kích hoạt ứng với đầu vào  $x_i$ , và  $n$  là số lượng đầu vào.

- một tập hợp neuron. Mang hành vi kế thừa của các neuron nhưng sẽ nhận nhiều đầu vào và cho ra nhiều kết quả. Để làm được điều này, ma trận được dùng để mô phỏng hành vi. Ngoài ra việc dùng ma trận thay vì tập trung vào mô phỏng từng đơn vị sẽ đem đến hiệu quả tính toán tốt hơn.

$$f(X) = \sigma(W^T X + B) \quad (2)$$

Với  $X$  là một tập đầu vào với kích thước  $\text{data\_points} \times \text{input\_dim}$ ,  $X$  là một ma trận.  $W$  là tập trọng số với kích thước  $\text{output\_dim} \times \text{input\_dim}$ .  $W$  cũng là một ma trận và  $W^T$  là ma trận chuyển vị.  $B$  đóng vai trò như hệ số bù với kích thước  $1 \times \text{output\_dim}$ .

Ở mặt hình tượng, đây là sơ đồ của mạng FNN.



Hình 4: Ảnh minh họa cho mạng FNN [8]. Gồm các trọng số và các đơn vị. Mạng này có 5 tầng gồm 1 tầng đầu vào, 1 tầng đầu ra và 3 tầng ẩn.

Qua quá trình phân tích và tìm hiểu về mạng FNN, bài viết làm rõ thêm về mối quan hệ của các đối tượng. Qua đó khẳng định thêm về vị trí đứng của **toán học** trong câu hỏi "Người lập trình lấy ý tưởng từ đâu?"

## 6 Tính kết nối giữa GA và FNN

### 6.1 Ở phạm vi con người

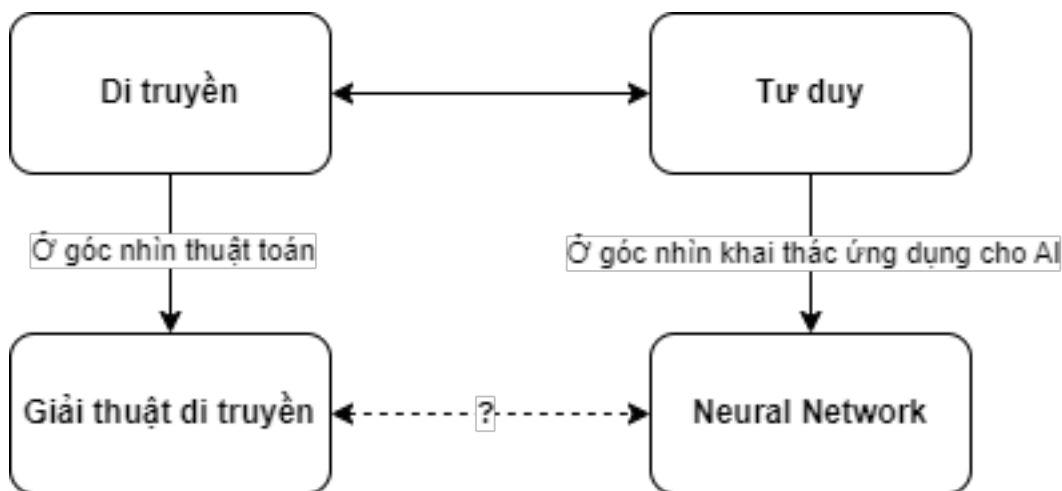
Trước khi trả lời câu hỏi này, hãy khảo sát sơ qua ý tưởng ứng dụng thuật toán di truyền vào mạng Neural Network bằng cách đặt câu hỏi trong bối cảnh con người chúng ta.

Không thể phủ nhận rằng, con người là một bản thể sinh học có sự sống, mỗi cá thể trong quần thể con người đều có những đặc điểm riêng đầy thú vị. Dựa trên lí giải sinh học, ta biết rằng chính gen là bộ khung tạo nên và làm cho quần thể con người tồn tại nhiều cá thể đầy thú vị.

Và thú vị hơn nữa, mỗi chúng ta đều có một tư duy khác nhau và độc lập với người khác. Ở điểm này, vậy có phải di truyền cũng là thứ làm cho

chúng ta khác biệt ở nhận thức? Đây là một câu hỏi mà để trả lời nó cần giao thoa quan điểm của nhiều thứ (di truyền, môi trường, xã hội, biến cố, ...), song không thể gạt bỏ di truyền ra khỏi tư duy.

Theo dòng chảy của suy luận trên, ta biết di truyền có kết nối với tư duy. Dựa trên mối kế thừa quan hệ ở lập trình, ta cũng phần nào đoán nhận giải thuật di truyền phải có một kết nối gì đó đến Neural Network.



Hình 5: Sơ đồ trên thể hiện Di truyền và tư duy có mối quan hệ với nhau. Thông qua góc nhìn thuật toán đối với di truyền làm phát sinh giải thuật di truyền, góc nhìn khai thác ứng dụng AI làm cho phát sinh mạng thần kinh nhân tạo (ANN). Mặt khác di truyền và tư duy có mối quan hệ, vậy thì giải thuật di truyền và ANN cũng phải có quan hệ tương tự. Dấu ? tượng trưng cho việc chưa xác định được mối quan hệ đó là gì.

## 6.2 Thử khai thác dấu ?

Đặt mục tiêu khai thác trong bối cảnh mạng FNN, ta thấy FNN có một số điểm đáng lưu tâm như sau:

- Kiến trúc FNN. Đó là số lượng số đơn vị neuron, số tầng, hàm kích hoạt, số tham số.
- Tham số FNN. Là về các trọng số kết nối giữa các tầng.

Mặt khác, một quy trình ứng dụng của mạng FNN gồm hai pha cơ bản sau:

- Pha đào tạo (Training). Là pha học của mạng. Là pha thay đổi các tham số sao cho việc chuyển đổi giữa X (Input) sang Y (Output) là tốt nhất có thể. [13]
- Pha ứng dụng. Là pha khai thác mạng đã qua đào tạo, đem vào dùng với dữ liệu có thể chưa qua đào tạo.

Xét đến, thuật toán di truyền (Genetic Algorithm) là một thuật toán tối ưu. Tuân theo quy trình, ta nắm được GA sẽ tham gia vào pha đào tạo (Training). Xét đến những điểm đáng lưu tâm, ta khẳng định GA có thể dùng để tối ưu các thành phần của mô hình như kiến trúc và tham số. [16]

Đối với các mô hình AI hiện tại, thường việc tối ưu sẽ dựa trên các phương pháp liên quan đến Gradient. Thuật toán di truyền nói riêng và họ thuật toán tối ưu không dùng Gradient vẫn đang phát triển nhưng không nổi trội bằng nhánh trên.

## **7 Ứng dụng GA, FNN vào tìm kiếm chiến lược sinh tồn của sinh vật**

### **7.1 Mô tả vấn đề**

Cho sự tồn tại của một môi trường với các thông số như nhiệt độ/độ ẩm/lượng thức ăn/mức năng lượng tối đa/ánh sáng/lượng mưa/độ Ph/tốc độ gió/số lượng kẻ thù.

Mỗi cá thể là một mạng Feedforward Neural Network với kiến trúc cố định (mô phỏng chiến lược sinh tồn). Mạng này nhận đầu vào là các thông số môi trường và trả kết quả đầu ra là chiến lược tìm kiếm thức ăn và cách phản ứng với kẻ thù.

Hãy ứng dụng thuật toán di truyền nhằm tìm kiếm ra chiến lược sinh tồn phù hợp đối với môi trường được cho.

### **7.2 Biện luận một số thành phần trong mô tả vấn đề**

Chiến lược sinh tồn của sinh vật là cách mà sinh vật tương tác với môi trường. Ở đây, bài viết đặt bối cảnh trong sự tương tác với môi trường để biểu lộ

cách mà thuật toán di truyền hoạt động cũng như gắn kết với quá trình chọn lọc tự nhiên.

Việc chọn mạng FNN là dùng để mô tả thêm sâu cách mà sinh vật tương tác để cho ra chiến lược tối ưu.

Sự chi phối của thuật toán di truyền đến mạng FNN là một ví dụ biểu thị cho mối quan hệ giữa GA và FNN, là cách di truyền ảnh hưởng đến tư duy.

### 7.3 Ý tưởng triển khai

Trong vấn đề này, với kiến trúc mô hình là cố định, ta xác định yếu tố cần tối ưu là các tham số của mạng. Như vậy, hãy xem tập tham số tượng trưng như một cá thể.

Mỗi bước thực hiện tìm kiếm, ta duy trì một tập các quần thể như vậy. Qua một số thế hệ hoạt động, ta có được những cá thể tiềm năng (mang chiến lược sinh tồn hợp lý), đây là lời giải cho vấn đề được nêu.

Hàm đánh giá độ thích nghi cá thể được thực thi qua các tiêu chí sau:

- Nếu chiến lược sinh tồn đưa ra mức năng lượng cao, khả năng nhận diện thức ăn tốt thì giá trị fitness cao (tăng cao khả năng tồn tại)
- Nếu môi trường có các điều kiện khắc nghiệt như mưa nhiều, áp suất thấp, độ Ph không ổn định thì fitness giảm
- Nếu chiến lược có khả năng chiến đấu tốt sẽ làm cho có lợi trong môi trường cạnh tranh (tùy vào môi trường)

### 7.4 Chi tiết một số thành phần thuật toán GA

Quần thể: là tập hợp các cá thể mang mã di truyền là các tham số của mạng FNN.

Hàm Fitness: nhận đầu vào là cá thể và môi trường. Kết quả trả về sẽ là điểm đánh giá mức độ thích nghi của chiến lược.

$$fitness = w_1 \times E + w_2 \times F - w_3 \times R - w_4 \times P - w_5 \times A + w_6 \times S \quad (3)$$

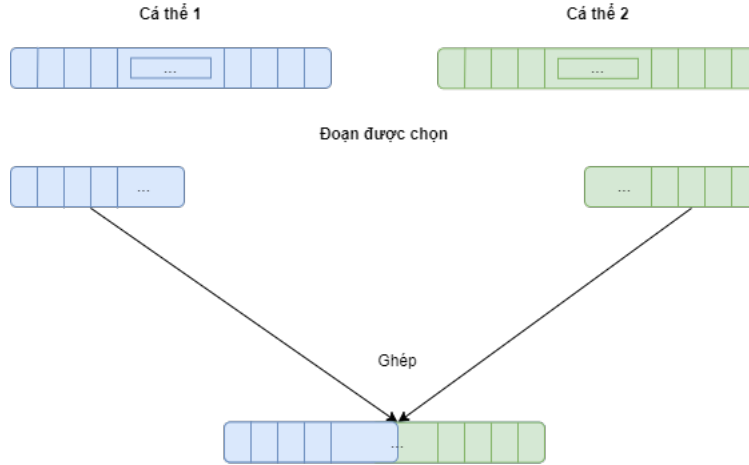
Với  $w_i$  là hệ số điểm đánh giá cho thành phần.  $E$  là mức năng lượng tiêu thụ của sinh vật.  $F$  là khả năng cảm nhận thức ăn của sinh vật.  $R$  là RainFall (lượng mưa), giá trị được chuẩn hoá về khoảng 0,1.  $P$  là ảnh hưởng của áp

suất, giá trị cũng được chuẩn hoá về khoảng 0,1.  $S$  là giá trị phản mức độ chiến đấu của sinh vật. Còn  $A$  là phản kháng lại môi trường lí tưởng với công thức:

$$A = \frac{1}{5} \left[ \left( \frac{\text{Temp} - 25}{25} \right)^2 + \left( \frac{\text{Humid} - 50}{50} \right)^2 + \left( \frac{\text{Wind\_speed}}{200} \right)^2 + \left( \frac{\text{pH} - 7}{7} \right)^2 + \left( \frac{\text{Energy} - 5000}{10000} \right)^2 \right] \quad (4)$$

Selection (Chọn lọc): được tiến hành dựa trên việc sắp xếp giá trị Fitness tương ứng với sinh vật. Số lượng cá thể bị loại đi sẽ chiếm một tỉ lệ cho trước so với tổng cá thể.

Crossover (Lai ghép): được tiến hành hoàn toàn ngẫu nhiên trên những cá thể được giữ lại. Cách lai ghép đến từ việc lắp ghép các đoạn gen từ hai cha con. Cá thể mới được bổ sung thay thế cá thể bị loại bỏ mang id tương ứng.



Hình 6: Minh hoạ mô hình lai ghép được dùng cho bài toán.

Mutation (Đột biến): được tiến hành hoàn toàn ngẫu nhiên trên tập quần thể. Chọn một cá thể, rồi chọn vị trí đột biến trên mã di truyền của cá thể. Rồi tiến hành đột biến.

**Mã giả của Crossover và Mutation ở mục minh hoạ bổ sung**

## 7.5 Kết quả thực thi thuật toán

Mã nguồn thuật toán được tham chiếu ở [2].



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  POLYGLOT NOTEBOOK

Hãy nhập số lượng cá thể: 100
Hãy nhập số lượng thế hệ: 500
Nhập giá trị cho nhiệt độ: 24
Nhập giá trị cho độ ẩm: 80
Nhập giá trị cho mức độ thức ăn: 0.7
Nhập giá trị cho lượng mưa: 850
Nhập giá trị cho áp suất khí quyển: 1000
Nhập giá trị cho tốc độ gió: 200
Nhập giá trị cho pH: 8
Nhập giá trị cho năng lượng tối đa của môi trường: 2000
```

Hình 7: Thông tin môi trường. Được nhập từ màn hình Console với thông tin được mô tả ở mục trên

Kết quả được chạy ở local với các thông số môi trường sau:

Với thông số trên, tiến hành đào tạo quần thể gồm 100 cá thể, qua 500 thế hệ, ta có kết quả:

Thực thi trên mã nguồn, sẽ hiển thị được thêm một số kết quả khác như thông số cài đặt thuật toán, mối quan hệ tổ tiên của một cá thể, ...

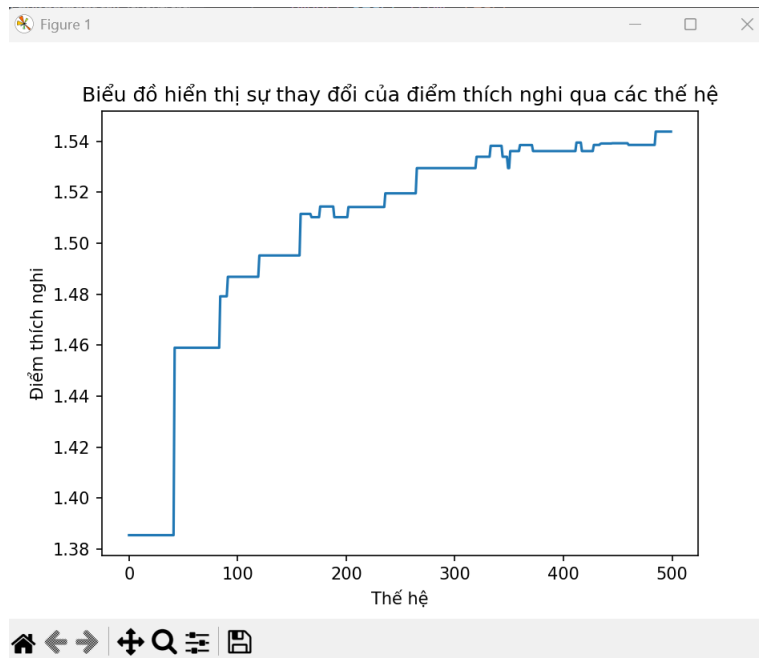
## 8 Kết luận

Thông qua phân tích và trả lời câu hỏi, bài viết đem đến một góc nhìn rõ ràng hơn về vai trò của người lập trình, quy trình tiếp thu và xử lý ý tưởng cũng như thiết lập vai trò của toán học trong lập trình.

Bài viết triển khai từ những ý tưởng chung nhất đến ý tưởng riêng và cô lập hơn. Mẫu hình triển khai này có thể được dùng cho việc tìm lời giải cho vấn đề.

Ngoài ra, qua quá trình phân tích, bài viết đi đến kết luận là AI cũng là một dạng thuật toán nên với vai trò là một người lập trình, chúng ta nên khai thác AI hiệu quả hơn nữa.

Bài viết tập trung nhiều vào việc chứng minh câu hỏi chủ đề và xác lập những hệ quả có ý nghĩa của câu hỏi đó nên những nội dung được cập sẽ có những mục chưa đủ độ sâu nên bài viết dùng thêm phần Tài liệu để mở rộng thêm.



Hình 8: Biểu đồ ghi nhận giá trị Fitness cao nhất qua các thế hệ. Tuy có những thế hệ mà điểm Fitness tăng rồi lại giảm (do tác động của tính ngẫu nhiên) nhưng xu hướng chung là tăng theo thời gian.

## 9 Tham khảo thêm

### Tài liệu

- [1] ADN. URL: <https://vi.wikipedia.org/wiki/DNA>.
- [2] Nguyễn Đức Bảo Lâm. *Mô phỏng thuật toán*. URL: <https://github.com/baolam/techaway-genetic-algo-mn>.
- [3] Di truyền. URL: [https://vi.wikipedia.org/wiki/Di\\_truy%E1%BB%81n](https://vi.wikipedia.org/wiki/Di_truy%E1%BB%81n).
- [4] *Differential Evolution*. URL: [https://en.wikipedia.org/wiki/Differential\\_evolution](https://en.wikipedia.org/wiki/Differential_evolution).
- [5] *Differential Evolution Algorithm*. URL: <https://www.sciencedirect.com/topics/computer-science/differential-evolution-algorithm#>:

~:text=Differential%20Evolution%20(DE)%20is%20another, of%20agents%20in%20the%20population.

- [6] Marco Dorigo **and** Christian Blum. “Ant colony optimization theory: A survey”. in *Theoretical Computer Science*: 344.2 (2005), **pages** 243–278. ISSN: 0304-3975. DOI: <https://doi.org/10.1016/j.tcs.2005.05.020>. URL: <https://www.sciencedirect.com/science/article/pii/S0304397505003798>.
- [7] *Evolution Strategy*. URL: [https://en.wikipedia.org/wiki/Evolution\\_strategy](https://en.wikipedia.org/wiki/Evolution_strategy).
- [8] *Feedforward neural network*. URL: [https://www.researchgate.net/figure/Feedforward-Neural-Network-Feedforward-Neural-Network-with-L-hidden-layers-four-units\\_fig2\\_349025766](https://www.researchgate.net/figure/Feedforward-Neural-Network-Feedforward-Neural-Network-with-L-hidden-layers-four-units_fig2_349025766).
- [9] *Genetic Algorithm*. URL: [https://en.wikipedia.org/wiki/Genetic\\_algorithm](https://en.wikipedia.org/wiki/Genetic_algorithm).
- [10] *Genetic Algorithms*. URL: <https://www.geeksforgeeks.org/genetic-algorithms/>.
- [11] Ian Goodfellow, Yoshua Bengio **and** Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [12] ThS. Lê Thị Ngọc Hiếu. *Giải bài toán tối ưu bằng giải thuật di truyền*. 2016. URL: [https://dnpu.edu.vn/upload/elfinder/T%E1%BA%A1p%20ch%C3%AD%20khoa%20h%E1%BB%8Dc/TCKH%20xu%E1%BA%A5t%20b%E1%BA%A3n/TCKH%20s%E1%BB%91%201/10.\\_85-93.pdf](https://dnpu.edu.vn/upload/elfinder/T%E1%BA%A1p%20ch%C3%AD%20khoa%20h%E1%BB%8Dc/TCKH%20xu%E1%BA%A5t%20b%E1%BA%A3n/TCKH%20s%E1%BB%91%201/10._85-93.pdf).
- [13] Subhash Kak. “On training feedforward neural networks”. in *Pramana*: 40 (1993), **pages** 35–42.
- [14] Ozlem Kilic **and** Quang M Nguyen. “Application of artificial immune system algorithm to electromagnetics problems”. in *Progress in Electromagnetics Research B*: 20 (2010), **pages** 1–17.
- [15] Melanie Mitchell. “Genetic algorithms: An overview.” in *Complex*. **volume** 1. 1. Citeseer. 1995, **pages** 31–39.
- [16] David J Montana, Lawrence Davis **and others**. “Training feedforward neural networks using genetic algorithms.” in *IJCAI*: **volume** 89. 1989, **pages** 762–767.
- [17] *Natural Selection*. URL: [https://en.wikipedia.org/wiki/Natural\\_selection](https://en.wikipedia.org/wiki/Natural_selection).

- [18] *Nhiễm sắc thể*. URL: [https://vi.wikipedia.org/wiki/Nhi%E1%BB%85m\\_s%E1%BA%AFc\\_th%E1%BB%83](https://vi.wikipedia.org/wiki/Nhi%E1%BB%85m_s%E1%BA%AFc_th%E1%BB%83).
- [19] Brian Schmidt **and others**. “Optimizing an artificial immune system algorithm in support of flow-Based internet traffic classification”. in *Applied Soft Computing*: 54 (2017), **pages** 1–22.
- [20] Le Duc Tien. *Thuật toán di truyền - Ứng dụng giải một số bài toán kinh điển (phần 1)*. URL: <https://viblo.asia/p/thuat-toan-di-truyen-ung-dung-giai-mot-so-bai-toan-kinh-dien-phan-1-RQqKLxJzK7z>.
- [21] Thúc TS.Nguyễn Đình. *Lập trình tiến hoá*. URL: <https://thuvienso.dau.edu.vn:88/handle/DHKTDN/6070>.
- [22] Ashish Vaswani **and others**. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: <https://arxiv.org/abs/1706.03762>.
- [23] vietcv. *Thuật toán là gì? Học thuật toán làm quái gì?* Truy cập lúc 12:14AM ngày 5/3/2025. URL: <https://codelearn.io/sharing/thuat-toan-la-gi-hoc-thuat-toan-lam-quai-gi>.
- [24] Dongshu Wang, Dapei Tan **and** Lei Liu. “Particle swarm optimization algorithm: an overview”. in *Soft computing*: 22.2 (2018), **pages** 387–408.
- [25] *What is Algorithm / Introduction to Algorithms*. URL: <https://www.geeksforgeeks.org/introduction-to-algorithms/>.

## 10 Minh họa bổ sung

---

**Algorithm 2** Lai ghép giữa các sinh vật được giữ lại

---

**Require:** Danh sách sinh vật được giữ lại  $kept\_creatures$ , độ dài ADN  $adn\_length$ , chỉ số bị loại  $removed\_idx$ , thế hệ hiện tại  $generation$

**Ensure:** Sinh vật con mới được tạo từ phép lai ghép

- 1: Chọn ngẫu nhiên  $parent1$  từ  $kept\_creatures$
  - 2: Chọn ngẫu nhiên  $parent2$  từ  $kept\_creatures$
  - 3:  $k \leftarrow$  Số nguyên ngẫu nhiên trong khoảng  $[0, adn\_length - 1]$
  - 4:  $adn \leftarrow parent1.adn()[0 : k] + parent2.adn()[k : adn\_length]$
  - 5: Tạo sinh vật con  $child$  với ID  $removed\_idx$ , thế hệ  $generation + 1$ , ADN  $adn$
  - 6: Thêm  $parent1.id$  vào danh sách tổ tiên của  $child$
  - 7: Thêm  $parent2.id$  vào danh sách tổ tiên của  $child$
  - 8: **return**  $child$
-

---

**Algorithm 3** Đột biến quần thể sinh vật

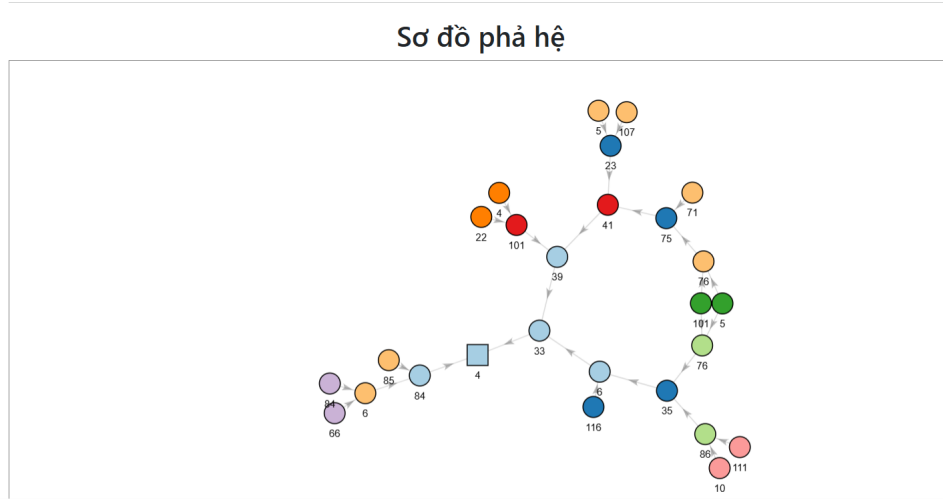
---

**Require:** Danh sách sinh vật *creatures*, tỷ lệ đột biến *mutation\_rate*, độ dài ADN *adn\_length*

**Ensure:** Quần thể sinh vật sau khi đột biến

```
1: population  $\leftarrow$  độ dài của creatures
2: mutated_length  $\leftarrow$  mutation_rate  $\times$  population
3: mutated_creatures  $\leftarrow$  Chọn ngẫu nhiên mutated_length sinh vật từ creatures
4: for each creature in mutated_creatures do
5:   gene_index  $\leftarrow$  Số nguyên ngẫu nhiên trong khoảng  $[0, \text{adn\_length} - 1]$ 
6:   value  $\leftarrow$  Số thực ngẫu nhiên trong khoảng  $[-1, 1]$ 
7:   adn  $\leftarrow$  creature.adn()
8:   adn[gene_index]  $\leftarrow$  value
9:   creature.update_adn(adn)
10:  creature.mutated_position.append(gene_index)
11: end for
12: return creatures
```

---



Hình 9: Sơ đồ thể hệ kế thừa của một cá thể. Hình vuông thể hiện cá thể mục tiêu, mã màu khác nhau biểu thị cho thể hệ. Con số ghi dưới mỗi hình tượng trưng cho Id của cá thể.