**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỂN THÔNG**



**BÁO CÁO ĐỒ ÁN**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI**

**THUẬT TOÁN DI TRUYỀN (BÀI TOÁN TÌM CHU TRÌNH ĐƯỜNG ĐI**

**CẢ NGƯỜI GIAO HÀNG)**

**Giảng viên hướng dẫn Sinh viên thực hiện**

Võ Thi Thức Trần Hoàng Kim – B2007245

Nguyễn Văn A – Bzzzaaaa

Nguyễn Văn B – Bvvvccc

**MỤC LỤC**

[**I.** **Bài toán đặt ra** 1](#_Toc119978848)

[**II.** **Hướng giải quyết bài toán** 1](#_Toc119978849)

[**1.** **Initialization** 2](#_Toc119978850)

[**2.** **Fitness Assignment** 2](#_Toc119978851)

[**3.** **Selection** 2](#_Toc119978852)

[**4.** **Crossover** 3](#_Toc119978853)

[**5.** **Mutation** 3](#_Toc119978854)

[**6.** **Termination** 3](#_Toc119978855)

[**III.** **Áp dụng thuật toán vào bài toán** 5](#_Toc119978856)

[**1.** **Initialization** 5](#_Toc119978857)

[**2.** **Fitness Assignment** 5](#_Toc119978858)

[**3.** **Selection** 6](#_Toc119978859)

[**4.** **Crossover** 6](#_Toc119978860)

[**5.** **Mutation** 7](#_Toc119978861)

[**6.** **Termination** 7](#_Toc119978862)

[**IV.** **Kết luận và hướng phát triển** 7](#_Toc119978863)

1. **Bài toán đặt ra**

Bài toán người bán hàng du lịch (TSP) là một bài toán nổi tiếng trong khoa học máy tính. Vấn đề có thể được tóm tắt như sau: hãy tưởng tượng chúng ta là một nhân viên bán hàng cần đến thăm một số thành phố. Vì ta muốn giảm thiểu chi phí đi lại, ta muốn tìm ra con đường hiệu quả nhất, con đường sẽ tốn ít thời gian di chuyển nhất. Ta được cung cấp một tọa độ của các thành phố để truy cập trên bản đồ. Làm thế nào ta có thể tìm thấy tuyến đường tối ưu?

Giải pháp tốt nhất sẽ là phương pháp sử dụng thuật toán brute force (thuật toán vét cạn), trong đó ta xem xét tất cả các khả năng khác nhau, tính toán khoảng cách ước tính cho từng khả năng và chọn một khả năng là con đường ngắn nhất. Mặc dù đây là một cách để giải quyết TSP, nhưng vấn đề với cách tiếp cận này là nó đòi hỏi rất nhiều tính toán - thời gian chạy của thuật toán brute force này sẽ là O(n!) với dữ liệu lớn thì sẽ là vấn đề lớn.

1. **Hướng giải quyết bài toán**

Chúng ta sẽ xem xét một cách khác để tiếp cận TSP là sử dụng giải thuật di truyền **thuật toán di truyền**. Như tên gọi của nó, **thuật toán di truyền** phần nào mô phỏng một quá trình tiến hóa, trong đó nguyên tắc về sự sống sót của kẻ thích nghi nhất đảm bảo rằng chỉ những gen tốt nhất mới sống sót sau một số lần lặp lại các chu kỳ tiến hóa qua một số thế hệ. Các thuật toán di truyền có thể được coi là một loại thuật toán ngẫu nhiên trong đó chúng ta sử dụng lấy mẫu ngẫu nhiên để đảm bảo rằng chúng ta thăm dò toàn bộ không gian tìm kiếm trong khi cố gắng tìm giải pháp tối ưu. Mặc dù các thuật toán di truyền không phải là phương pháp hiệu quả nhất hoặc được đảm bảo nhất để giải quyết TSP, nhưng đó là một cách tiếp cận tốt để giải quyết tình trạng với dữ liệu lớn.

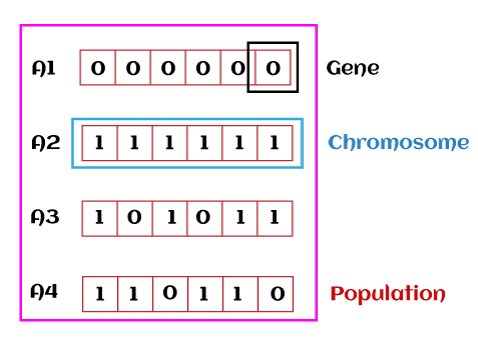
**Thuật toán di truyền**

Thuật toán di truyền hoạt động theo chu kỳ thế hệ tiến hóa để tạo ra các cá thể chất lượng cao. Thuật toán này sử dụng các hoạt động khác nhau nhằm nâng cao hoặc thay thế tổng thể để đưa ra giải pháp phù hợp được cải thiện. Nó bao gồm năm giai đoạn để giải quyết các vấn đề tối ưu hóa phức tạp:

* **Initialization**
* **Fitness Asignment**
* **Selection**
* **Crossover**
* **Mutation**
* **Termination**

1. **Initialization**

Quá trình của thuật toán di truyền bắt đầu bằng cách tạo ra một tập hợp các cá thể, được gọi là quần thể. Ở đây mỗi cá thể là giải pháp cho vấn đề nhất định. Một cá thể chứa hoặc được đặc trưng bởi một tập hợp các thông số được gọi là **gen**. Các **gen** được kết hợp thành một chuỗi và tạo ra các nhiễm sắc thể, đó là giải pháp cho vấn đề. Một trong những kỹ thuật phổ biến nhất để khởi tạo là sử dụng các chuỗi nhị phân ngẫu nhiên.



1. **Fitness Assignment**

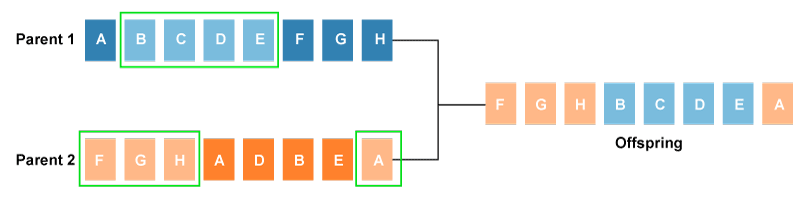
Giá trị **Fitness** được sử dụng để xác định mức độ phù hợp của một cá thể? Nó có nghĩa là khả năng của một cá thể để cạnh tranh với các cá thể khác. Trong mỗi lần lặp lại, các cá thể được đánh giá dựa trên giá trị **Fitness**. Giá trị **Fitness** cung cấp điểm số cho mỗi cá thể. Điểm số này xác định thêm xác suất được chọn để sinh sản. Giá trị **Finess** càng cao thì càng có nhiều cơ hội được chọn để sinh sản.

1. **Selection**

Giai đoạn chọn lọc liên quan đến việc lựa chọn các cá thể có gen tốt để sinh sản con cái. Tất cả các cá thể được chọn sau đó được sắp xếp thành từng cặp để sinh sản. Sau đó, những cá thể này chuyển gen của nó cho thế hệ tiếp theo.

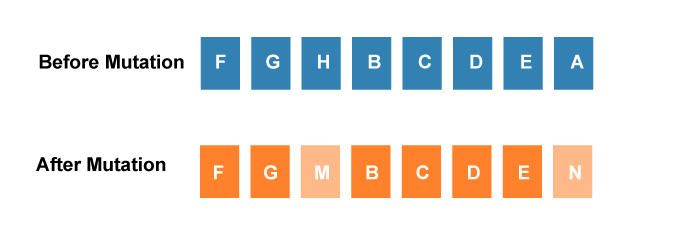
1. **Crossover**

Sự trao đổi chéo đóng một vai trò quan trọng nhất trong giai đoạn sao chép của thuật toán di truyền. Trong quá trình này, một điểm trao đổi chéo được chọn ngẫu nhiên trong các gen. Sau đó, toán tử lai hoán đổi thông tin di truyền của hai bố mẹ từ thế hệ hiện tại để tạo ra một cá thể mới đại diện cho thế hệ con.



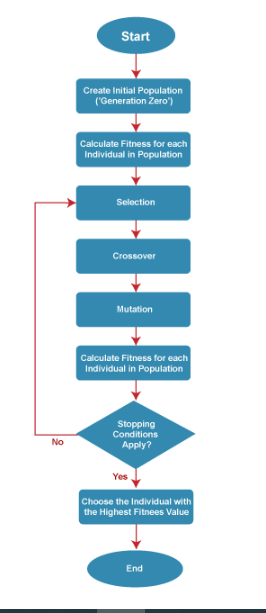
1. **Mutation**

Đột biến chèn các gen ngẫu nhiên vào đời con (đứa con mới) để duy trì tính đa dạng trong quần thể. Nó có thể được thực hiện bằng cách lật một số bit trong nhiễm sắc thể. Đột biến giúp giải quyết vấn đề hội tụ sớm và tăng cường đa dạng hóa.



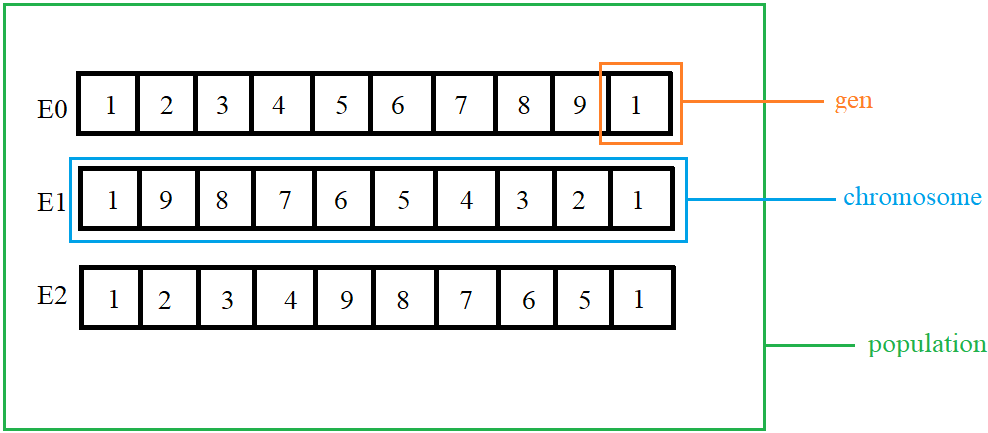
1. **Termination**

Tiêu chí dừng được áp dụng làm cơ sở để chấm dứt. Thuật toán kết thúc sau khi đạt được giải pháp phù hợp với ngưỡng. Nó sẽ xác định giải pháp cuối cùng là giải pháp tốt nhất trong quần thể.

**Sơ đồ**

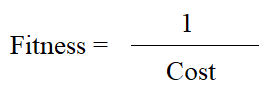
1. **Áp dụng thuật toán vào bài toán**
2. **Initialization**

Với các thành phố là các gen, ta sẽ khai báo 1 **class City** với thuộc tính x, y và mã thành phố để giữ các dữ kiện cần thiết, danh sách các thành phố sẽ tạo nên nhiễm sắc thể vì thế ta cũng cần 1 **class Chromosome** để tính giá trị **fitness** cho mỗi nhiễm sắc thể, do mỗi thành phố chỉ đi đúng 1 lần nên điểm bắt đầu và điểm kết thúc phải giống nhau do đó nếu ta muốn tạo **n** nhiễm sắc thể thì chỉ cần giữ vị trí tại thành phố ban đầu và **random** các **gen** (thành phố) còn lại để tạo ra **n** nhiễm sắc thể.



1. **Fitness Assignment**

Mục tiêu của mình là tìm được đường đi tới các thành phố đúng 1 lần và quay lại thành phố ban đầu sao cho quãng đường ngắn chấp nhận được. Vì thế ta cần sẽ tính tổng quãng đường đi qua các thành phố của mỗi **chromosome** quãng đường nào càng ngắn thì càng gần với mục tiêu của mình, do giá trị **Fitness** cần là giá trị lớn chứ không phải nhỏ như trường hợp này, cách đơn giản nhất ta sẽ nghịch đảo giá trị quãng đường của mỗi **chromosome,** ta có giá trị **Fitness** sẽ được tính là:

****

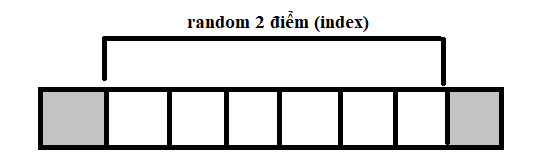
1. **Selection**

Ta tiến hành chọn 1 vài các cá thể trong quần thể một cách **ngẫu nhiên**,

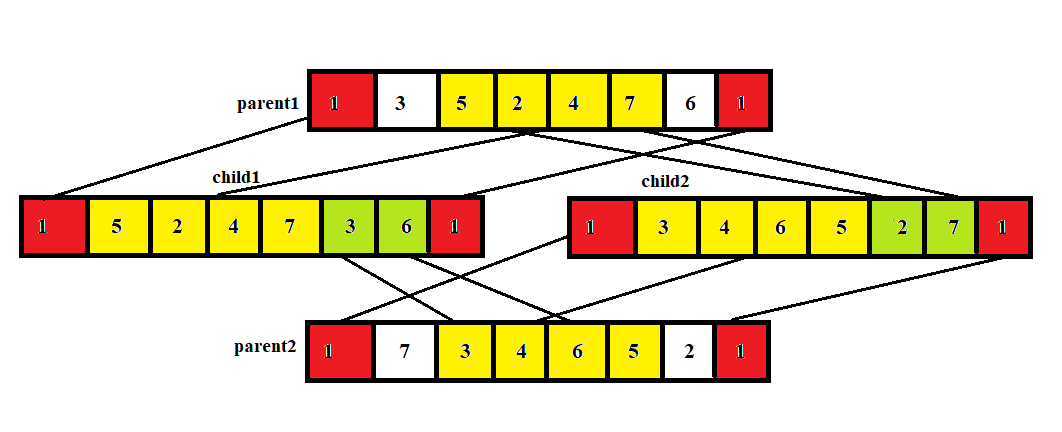
Sau đó tiến hành so sánh với nhau giữa các cá thể về giá trị **Fitness** mà đã được tính để tìm ra được cá thể tốt nhất trong vài cá thể được chọn đấy để chuyển **gen** của cá thể này vào thế hệ tiếp theo.

1. **Crossover**

Đầu tiên là **random** một cách ngẫu nhiên 2 điểm (index) để tiến hành lai tạo.

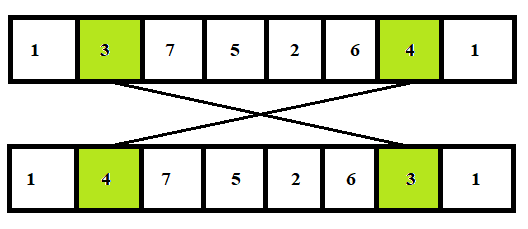


Khi có 2 điểm (index) ta sẽ tiến hành lấy ra **sublist** trong khoảng đấy của 2 **chromosome** và tiến hành lai tạo (điểm bắt đầu và kết thúc phải giữ nguyên không được lai tạo, thế hệ cha như thế nào thì thế con như thế đấy)**.**



1. **Mutation**

Ta sẽ tạo đột biến bằng cách đổi chỗ ngẫu nhiên với 2 thành phố tạo nên đường đi mới.



1. **Termination**

Tiêu chí dừng ở đây sẽ là số thế hệ cần lặp hoặc là 1 giá trị nào đó chấp nhận được, nhưng việc mà dùng 1 giá trị tạm chấp nhận nào đó thì nó sẽ không tối ưu bằng việc thiết lặp số thế hệ cần lặp để tìm ra giải pháp tốt chấp nhận được.

1. **Kết luận và hướng phát triển**

Thuật toán di truyền xuất hiện để tìm giải pháp tốt cho bài toán **TSP**, tuy nhiên

nó phụ thuộc vào các vấn đề được mã hóa, phương pháp lai ghép và đột biến

nào được sử dụng. Vấn đề của bài toán **TSP** là rất khó để duy trì cấu trúc từ các **NST** của cha mẹ và kết thúc ở thế hệ con, có lẽ lai ghép hoặc đột biến sẽ giữ lại 1 phần cấu trúc của **NST** cha mẹ tốt hơn. Vơi phương pháp coi mỗi thành phố như 1 **gen** và tập hợp các thành phố như là 1 bộ **NST** sẽ là phương pháp tốt với lượng dữ liệu lớn mà không thể giải quyết bằng thuật toán vét cạn.

**Giải thuật di truyền** sẽ có ưu thế trong những bài toán có sẵn lời giải trong một không gian tìm kiếm lớn như những bài toán cổ điển Người du lịch, Cái túi, … Ngoài ra, hiện nay với việc bùng nổ của AI, việc sử dụng mạng Neural cùng với giải thuật này đã giúp giải quyết một số bài toán về tự động, hành vi, …