# GIẢI THUẬT DI TRUYỀN VÀ CHU TRÌNH HAMILTON

**Vũ Đình Hòa, Nguyễn Hữu Mùi**

*Khoa CNTT ĐHSP Hà Nội*

***Tóm tắt:*** *Bài toán xác định chu trình Hamilton trong một đồ thị vô hướng cho trước được biết là bài toán NP-complete. Điều đó có nghĩa là khó có một thuật toán tốt (thuật toán với thời gian đa thức đối với máy tính Turing tất định) để nhận biết liệu một đồ thị vô hướng cho trước có một chu trình Hamilton hay không. Chúng tôi trình bày phương pháp giải thuật di truyền quen biết để xác định lời giải gần đúng bài toán người du lịch vào việc xác định xem một đồ thị vô hướng cho trước có chu trình Hamilton hay không.*

## 1. GIỚI THIỆU THUẬT TOÁN DI TRUYỀN CỔ ĐIỂN

### 1. 1 Cấu trúc của thuật toán di truyền cổ điển

**• Cấu trúc nhiễm sắc thể và kiểu gien**

Trong GA (Genetic Algorithms) cổ điển, các cá thể (hay còn gọi là nhiễm sắc thể viết tắt là NST) được mã hoá bởi các chuỗi nhị phân, mỗi vị trí trên chuỗi chỉ nhận một trong hai giá trị "1" hoặc "0". Một NST trong GA cổ điển có dạng như sau:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

Mỗi NST cụ thể biểu diễn một lời giải có thể của bài toán, một quá trình tiến hoá được thực hiện trên một quần thể (một tập hợp NST) tương đương với sự tìm kiếm trong một không gian các lời giải có thể. Sự tìm kiếm này đòi hỏi sự cân bằng giữa hai mục đích: tìm lời giải tốt nhất và khám phá không gian tìm kiếm mới.

GA cổ điển thực hiện tìm kiếm theo nhiều hướng bằng cách duy trì một tập lời giải có thể, khuyến khích sự hình thành và trao đổi thông tin giữa các hướng. Tập hợp lời giải trải qua các quá trình tiến hoá và cuối cùng cho ta một lời giải đủ tốt tuỳ theo yêu cầu. Tại mỗi thế hệ các lời giải tương đối tốt được chọn để tái sinh, trong khi đó các lời giải tương đối tồi bị loại bỏ. Để phân biệt mức độ tốt, xấu giữa các lời giải khác nhau, người ta dùng một hàm được gọi là "hàm mục tiêu" hay "hàm thích nghi" vì hàm này tương đương với môi trường sống trong thuyết tiến hoá.

GA cổ điển được J. H. Holland giới thiệu để giải bài toán tối ưu: $\max\{ f(x) / x \in M \}$, trong đó M là hình hộp trong không gian số thực n-chiều, $f(x) > 0$ với mọi x thuộc M. Cấu trúc của GA cổ điển như sau:

```
Procedure GA
Begin
        t ← 0
        khởi tạo P(t)
        đánh giá P(t)
while ( not điều_kiện_dừng ) do
    begin
```

t ← t + 1

chọn P(t) từ P(t-1)

thay đổi P(t)

đánh giá P(t)

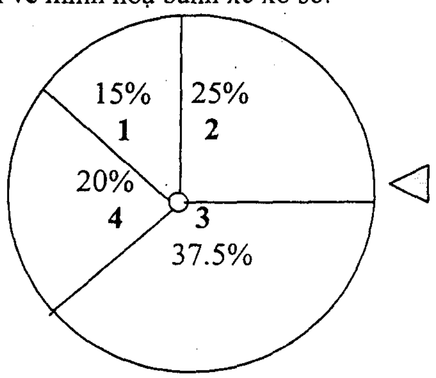
end

End

Quá trình tiến hoá được diễn ra trong vòng lặp while, tại thế hệ thứ t, thuật toán duy trì một tập lời giải P(t) = {$xt_1$,...,$xt_n$ }. Mỗi lời giải xti được đánh giá "độ thích nghi". Một tập lời giải mới được xây dựng (vòng lặp t+1) bằng cách *"chọn lọc"* các cá thể thích nghi hơn ta được tập lời giải trung gian. Tiếp theo một số cá thể trong tập lời giải đã được chọn này bị biến đổi bằng phương pháp *"lai ghép"* và *"đột biến"* để tạo thành các lời giải mới cho thế hệ thứ t + 1. Chúng ta sẽ tìm hiểu nội dung của của các phép ***chọn lọc***, ***lai ghép*** và ***đột biến*** qua các mục dưới đây:

- **Phép chọn lọc**

Phép chọn lọc là một quá trình trong đó các cá thể được chọn để tham gia vào các pha tiếp theo của quá trình tiến hoá, việc lựa chọn này tuỳ thuộc độ thích nghi của cá thể đó. nghĩa là những cá thể nào có giá trị hàm thích nghi cao hơn thì sẽ có khả năng có nhiều con cháu hơn trong các thế hệ tiếp theo. Phép chọn lọc có thể được biểu diễn dưới dạng một bánh xe xổ số, đó là một hình tròn trong đó mỗi cá thể trong thế hệ hiện hành chiếm một phần tương ứng với giá trị hàm thích nghi của nó. Các giá trị này tương ứng với các xác suất chọn lọc của mỗi cá thể vi được tính theo công thức pi = eval(vi) / F (trong đó eval(vi) là giá trị của hàm thích nghi của mỗi cá thể vi, F là tổng các giá trị của các hàm thích nghi của quần thể, Pi là xác suất chọn lọc của mỗi cá thể). Sau đây là hình vẽ minh hoạ bánh xe xổ số:




Hình 1 - Bánh xe xổ số

Trong hình trên, cá thể 1 có xác suất chọn lọc là 15%, mỗi lần quay bánh xe xổ số nó có khả năng được chọn là 0.15. Tương tự như vậy đối với các cá thể 2, 3, 4.

- **Phép trao đổi chéo**

Phép trao đổi chéo (còn gọi là phép lai ghép), kết hợp các đặc tính trên NST của bố và mẹ để tạo thành hai cá thể mới, bằng cách tráo đổi các đoạn gien tương ứng trên các NST của bố và mẹ. Ví dụ:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Parent 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| Parent 2 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Thì việc trao đổi chéo các NST sau gien thứ năm sẽ tạo ra hai con:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Child 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| Child 2 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

- **Phép đột biến**

Phép đột biến là sự sửa đổi một vài gien của một NST được chọn, bằng cách thay đổi ngẫu nhiên với xác suất là tỷ lệ đột biến. Ví dụ:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $v_1$ | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| $v'_1$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

NST $v_1$ được chọn để đột biến tại vị trí gen số năm, gien này hiện tại là 0 sau khi đột biến sẽ chuyển thành 1. Khi này NST $v_1$ trở thành $v_1$'.

Thuật toán di truyền cổ điển dùng phương pháp mã hoá nhị phân truyền thống, do vậy khi áp dụng cho các bài toán trong không gian nhiều chiều và có độ chính xác cao thì mỗi NST sẽ có độ dài rất lớn, nên việc áp dụng GA cổ điển cho các bài toán này là rất khó khăn. Mặt khác, đối với những bài toán có nhiều ràng buộc phức tạp thì các toán tử di truyền truyền thống tỏ ra kém hiệu quả. Chính vì vậy, trong những năm vừa qua rất nhiều hướng tiếp cận dựa trên các nguyên lý về tiến hoá và chọn lọc tự nhiên được nghiên cứu và phát triển. Phần lớn các nhà nghiên cứu về thuật toán di truyền đã cải tiến thuật toán bằng cách dùng các biểu diễn không thuộc dạng chuỗi nhị phân, hay thiết kế các toán tử di truyền đặc biệt để phù hợp với bài toán cần giải.

***Những phát triển của GA cổ điển có tên gọi chung là phương pháp tính toán tiến hoá (Evolutionary Computation) viết tắt là EC.***

## 2. ÁP DỤNG THUẬT TOÁN DI TRUYỀN GIẢI BÀI TOÁN CHU TRÌNH HAMILTON

### 2.1. Nội dung bài toán

Cho trước một đồ thị đơn vô hướng. Một chu trình Hamilton là một đường đi dọc theo các cạnh qua tất cả các đỉnh của đồ thị và trở về đỉnh ban đầu sao cho mỗi đỉnh chỉ đi qua đúng một lần mà thôi. Bài toán quyết định:

INstance: Cho trước một đồ thị đơn vô hướng G.

Question: Tồn tại hay không trong G một chu trình Hamilton?

Được biết đây là một bài toán NPC.

Bài toán chu trình Hamilton có thể giải được nhờ sử dụng các thuật toán gien giải bài toán người du lịch. Một người du lịch sẽ đi qua tất cả các thành phố trong một chuyến du lịch và lại trở về nơi xuất phát. Biết khoảng cách đi lại giữa hai cặp thành phố bất kỳ, người du lịch phải lập lịch trình đi sao cho tổng đoạn đường của chuyến đi là ngắn nhất.

Không gian tìm kiếm của bài toán người du lịch là tập các hoán vị của n thành phố. Như vậy, không gian lời giải có kích thước là n!, mỗi hoán vị cho ta một lời giải, lời giải tối ưu là một hoán vị cho ta tổng đoạn đường của chuyến đi là ngắn nhất.

Bài toán người du lịch thuộc loại NP khó. Bản chất của nó chính là tìm một chu trình Hamilton với trọng số (được hiểu là tổng số độ dài/trọng số của các cạnh) nhỏ nhất có thể trong một đồ thị đầy đủ có n đỉnh và các cạnh được gán các trọng số dương cho trước. Trong những năm gần đây người ta đã áp dụng một số thuật toán để tìm lời giải gần tối ưu cho bài toán này. Bài viết này sẽ trình bày việc áp dụng "phương pháp tính toán tiến hoá" để tìm lời giải tối ưu cho bài toán này và bài toán quyết định tìm chu trình Hamilton trong đồ thị cho trước.

**2.2. Chọn hàm đánh giá và biểu diễn gien**

**2.2.1. Chọn hàm đánh giá**

Mỗi lời giải của bài toán người du lịch là một hoán vị của n thành phố. Phương pháp đơn giản nhất để đánh giá độ tốt xấu của một lời giải là tính tổng khoảng cách giữa các thành phố.

**2.2.2. Biểu diễn gien và các toán tử gien**

Trong những năm gần đây, có một số cách biểu diễn gien cho bài toán người du lịch hay được áp dụng đó là: biểu diễn *liên thuộc*, biểu diễn *thứ tự* và biểu diễn *đường đi*.

**a) Biểu diễn liên thuộc**

**$a_1$) Biểu diễn gien**

Một lời giải hay một hành trình được biểu diễn bằng một dãy các thành phố. Thành phố j được đặt tại vị trí i khi và chỉ khi có đường đi trực tiếp từ i đến j. Ví dụ, xét một lời giải cho một hành trình gồm 9 thành phố: (2 4 8 3 9 7 1 5 6), lời giải trên biểu diễn hành trình sau đây: 1 - 2 - 4 - 3 - 8 - 5 - 9 - 6 - 7 - 1.

Nếu một biểu diễn liên thuộc đại diện cho một hành trình, mà hành trình đó không đi qua tất cả các thành phố thì đó là một hành trình không hợp lệ hay hành trình không đầy đủ. Ví dụ, (2 4 8 1 9 3 5 7 6) đại diện cho hành trình: 1 -2 - 4 - 1, đây là một hành trình không đầy đủ.

**$a_2$) Toán tử trao đổi chéo**

Có nhiều phép trao đổi chéo đã được thử nghiệm cho cách biểu diễn liên thuộc, sau đây sẽ trình bày hai cách:

- **Phép trao đổi chéo "luân phiên cạnh - alternating-edges":**

Phép trao đổi chéo này xây dựng một cá thể con (offspring) bằng cách chọn ngẫu nhiên một cạnh từ một cá thể cha/mẹ (parents), sau đó chọn một cạnh thích hợp từ cá thể cha/mẹ kia. (tức là việc mở rộng hành trình bằng cách chọn các cạnh luân phiên từ cha và mẹ). Nếu cạnh mới từ một trong hai các thể cha/mẹ tạo một chu trình cho một hành trình không đầy đủ, toán tử sẽ chọn một cạnh ngẫu nhiên trong số các cạnh còn lại mà không tạo chu trình.

Ví dụ, cho hai cá thể cha/mẹ:

$p_1 = (2\ 3\ 8\ 7\ 9\ 1\ 4\ 5\ 6)$

$p_2 = (7\ 5\ 1\ 6\ 9\ 2\ 8\ 4\ 3)$.

Thì cá thể con có thể là: $c_1 = (2\ 5\ 8\ 7\ 9\ 1\ 6\ 4\ 3)$.

Trong đó quá trình tạo cá thể con bắt đầu từ cạnh (1,2) - $p_1$, (2,5) - $p_2$, (5,9) - $p_1$, (9,3) - $p_2$, (3,8) - $p_1$, (8,4) - $p_2$, 4,7) - $p_1$, (7,6) - $p_2$ thay cho (7,8) - $P_2$ (vì tạo ra chu trình cho hành trình không đầy đủ), (6,1) - $p_1$.

• **Phép trao đổi chéo "heuristic":**

Phép trao đổi chéo heuristic tạo ra cá thể con bằng cách chọn ngẫu nhiên một thành phố làm điểm xuất phát cho hành trình của cá thể con. Tiếp theo đó, xét hai cạnh thuộc hành trình của hai cá thể cha/mẹ xuất phát từ thành phố này chọn lấy một cạnh ngắn hơn cho hành trình của cá thể con. Thành phố thuộc đầu kia của cạnh được chọn trở thành điểm xuất phát,... Nếu tại thời điểm nào đó, cung ngắn hơn tạo thành chu trình cho hành trình không đầy đủ mà cung dài hơn không tạo thành chu trình thì chọn cung dài hơn. Nếu cả hai đều tạo thành chu trình thì hành trình sẽ được mở rộng bằng một cạnh ngẫu nhiên trong số các cạnh còn lại mà không tạo ra chu trình.

**b) Biểu diễn thứ tự**

**$b_1$) Biểu diễn gien**

Phương pháp này biểu diễn một hành trình bằng một dãy n thành phố. Phần tử thứ i của dãy là một số trong khoảng từ 1 đến n-i+1. ý tưởng của biểu diễn thứ tự như sau: giả sử C là một dãy các thành phố xếp thứ tự nào đó. Ví dụ, cho dãy thứ tự các thành phố C = (1 2 3 4 5 6 7 8 9). Một hành trình 1 - 2 - 4 - 3 - 8 - 5 - 9 - 6 - 7 được biểu diễn bằng một dãy l = (1 1 2 1 4 1 3 1 1), và được giải thích như sau:

- Số đầu tiên trong dãy l là 1, do đó thành phố đầu tiên trong dãy C là thành phố đầu tiên của hành trình (thành phố số 1) và di chuyển nó khỏi C. Hành trình cục bộ là: 1; C = (2 3 4 5 6 7 8 9).
- Số tiếp theo trong dãy l cũng là 1, do đó thành phố đầu tiên trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 2) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2; C = (3 4 5 6 7 8 9).
- Số tiếp theo trong dãy l là 2, do đó thành phố thứ 2 trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 4) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2 - 4; C = (3 5 6 7 8 9).
- Số tiếp theo trong dãy l là 1, do đó thành phố đầu tiên trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 3) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2 - 4 - 3; C = (5 6 7 8 9).
- Số tiếp theo trong dãy l là 4, do đó thành phố thứ tư trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 8) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2 - 4 - 3 - 8; C = (5 6 7 9).

- Số tiếp theo trong dãy l là 1, do đó thành phố đầu tiên trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 5) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2 - 4 - 3 - 8 - 5; C = (6 7 9).
- Số tiếp theo trong dãy l là 3, do đó thành phố thứ ba trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 9) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2 - 4 - 3 - 8 - 5 - 9; C = (6 7).
- Số tiếp theo trong dãy l là 1, do đó thành phố đầu tiên trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 6) và di chuyển nó khỏi C. Hành trình cục bộ là: 1 - 2 - 4 - 3 - 8 - 5 - 9 - 6; C = (7)
- Số cuối cùng trong dãy l là 1, do đó thành phố đầu tiên trong dãy C còn lại là thành phố tiếp theo của hành trình (thành phố số 7, thành phố còn lại cuối cùng) và di chuyển nó khỏi C. Hành trình cuối cùng là: 1 - 2 - 4 - 3 - 8 - 5 - 9 -7.

**$b_2$) Phép trao đổi chéo**

Phép trao đổi chéo truyền thống có thể áp dụng được đối với biểu diễn thứ tự. Với một cặp hành trình cha/mẹ, chọn một vị trí trao đổi chéo và bắt chéo tại vị trí đó sẽ cho ta hai cá thể con. Ví dụ, cho hai cá thể cha/mẹ:

$p_1$ = (1 1 2 1 | 4 1 3 1 1), $p_2$ = (5 1 5 5 | 5 3 3 2 1). Hai hành trình tương ứng là: 1 - 2 - 4 - 3 - 8 - 5 - 9 - 6 - 7 và 5 - 1 - 7 - 8 - 9 - 4 - 6 - 3 - 2.

Điểm bắt chéo là sau vị trí thứ tư của dãy, sẽ tạo ra hai cá thể con:

$c_1$ = (1 1 2 1 5 3 3 2 1) và $c_2$ = (5 1 5 5 4 1 3 1 1). Hai hành trình tương ứng với hai thể con là: 1 - 2 - 4 - 3 - 9 - 7 - 8 - 6 - 5 và 5 - 1 - 7 - 8 - 6 - 2 - 9 - 3 - 4.

Chúng ta nhận thấy phần đường đi ở bên trái điểm bắt chéo không thay đổi, còn phần đường đi ở bên phải lại bị xáo trộn một cách tuỳ tiện. Các kết quả thực nghiệm cho thấy cách biểu diễn thứ tự cộng với phép trao đổi chéo truyền thống không thật phù hợp cho bài toán người du lịch.

**c) Biểu diễn đường đi**

**$c_1$) Biểu diễn gien**

Biểu diễn đường đi là phương pháp biểu diễn tự nhiên nhất của một hành trình. Ví dụ, một hành trình: 5 - 1 - 7 - 8 - 9 - 4 - 6 - 3 - 2, được biểu diễn bằng dãy sau: (5 1 7 8 9 4 6 3 2).

**$c_2$) Phép trao đổi chéo**

Người ta đã đưa ra ba phép trao đổi chéo cho biểu diễn đường đi đó là: trao đổi chéo ánh xạ không đầy đủ (partially mapped crossover), trao đổi chéo thứ tự (order crossover), trao đổi chéo chu trình (cycle crossover).

- **Phép trao đổi chéo ánh xạ không đầy đủ:** cá thể con được tạo ra bằng cách:

1. Lấy hai điểm bắt chéo từ cá thể cha/mẹ, tráo đổi hai đoạn nằm giữa hai điểm bắt chéo cho nhau, các vị trí còn lại tạm thời chưa xác định. Ví dụ:

$p_1$ = (1 2 3 | 4 5 6 7 | 8 9), $p_2$ = (4 5 2 | 1 8 7 6 | 9 3). Ta có hai các thể con sau:

$o_1$ = (x x x | 1 8 7 6 | x x) và $o_2$ = (x x x | 4 5 6 7 | x x).

Sự tráo đổi này còn xác định một ánh xạ:

1<—> 4, 8<—>5, 7<—> 6, 6<—>7.

2. Tiếp theo chúng ta có thể điền thêm các thành phố vào các vị trí "x" sao cho không tạo mâu thuẫn: $o_1$ = (x 2 3 | 1 8 7 6 | x 9) và $o_2$ = (x x 2 | 4 5 6 7 | 9 3).

3. Cuối cùng, ký hiệu x đầu tiên của $o_1$ lẽ ra là 1 nhưng tạo ra mâu thuẩn nên thay bởi 4 (vì có ánh xạ 1<—> 4). Tương tự, ký hiệu x thứ hai của $o_1$ được thay bởi 5, hai ký hiệu x trong $o_2$ lần lượt được thay bởi 1 và 8, ta có:

$o_1$ = (4 2 3 | 1 8 7 6 | 5 9) và $o_2$ = (1 8 2 | 4 5 6 7 | 9 3).

• **Phép trao đổi chéo thứ tự:** được đề xuất bởi Davis - tạo ra các cá thể con bằng cách:

1. Chọn một dãy con của một hành trình từ mỗi cá thể cha/mẹ và bảo toàn thứ tự tương đối của các thành phố trong hành trình của cá thể cha/mẹ kia. Ví dụ, cho hai cá thể cha/mẹ sau:

$p_1$ = (1 2 3 | 4 5 6 7 | 8 9), $p_2$ = (4 5 2 | 1 8 7 6 | 9 3). Các đoạn nằm giữa hai điểm bắt chéo được copy sang các cá thể con:

$o_1$ = (x x x | 4 5 6 7 | x x), $o_2$ = (x x x | 1 8 7 6 | x x).

2. Tiếp theo, một hành trình được tạo ra bắt đầu từ điểm trao đổi chéo thứ hai của một cá thể cha/mẹ. Ví dụ: 9 - 3 - 4 - 5 - 2 - 1 - 8 - 7 - 6.

3. Sau đó loại bỏ các thành phố 4, 5, 6, và 7 có mặt trong cá thể con $o_1$ ta được: 9 - 3 - 2 - 1 - 8. Dãy này được đặt vào cá thể con thứ nhất bắt đầu từ điểm cắt thứ hai:

$o_1$ = (2 1 8 | 4 5 6 7 | 9 3). Làm tương tự như vậy ta có:

$o_2$ = (3 4 5 | 1 8 7 6 | 9 2).

Như vậy, trong phép trao đổi chéo thứ tự, chúng ta chỉ quan tâm tới thứ tự của các thành phố chứ không quan tâm tới vị trí của chúng. Tức là, hai hành trình sau là một: 9 - 3 - 4 - 5 - 2 - 1 - 8 - 7 - 6 và 4 - 5 - 2 - 1 - 8 - 7 - 6 - 9 - 3.

• **Phép trao đổi chéo chu trình:**

Trong phép trao đổi chéo chu trình, việc xây dựng các cá thể con theo nguyên tắc: mỗi thành phố và vị trí của nó được lấy từ một trong hai cá thể cha/mẹ. Chúng ta sẽ làm rõ cơ chế trao đổi chéo chu trình thông qua ví dụ dưới đây:

Cho hai cá thể cha/mẹ: $p_1$ = (1 2 3 4 5 6 7 8 9) và

$p_2$ = (4 1 2 8 7 6 9 3 5).

Tạo cá thể con thứ nhất:

- Lấy thành phố thứ nhất từ $p_1$: $o_1$ = (1 x x x x x x x x).

- Thành phố tiếp theo phải là số 4, vì thành phố này nằm ngay dưới thành phố 1 vừa được chọn và thành phố này ở vị trí số 4 nên ta có: $o_1$ = (1 x x 4 x x x x x).

- Tương tự như vậy, lần lượt các thành phố 8, 3, 2 được chọn: $o_1$ = (1 2 3 4 x x x 8 x).

- Khi thành phố 2 được chọn kéo theo sự lựa chọn thành phố 1 đã có trong hành trình. Ta đã hoàn thành một chu trình, các thành phố còn khuyết của $o_1$ sẽ được sao chép nguyên si từ $p_2$: $o_1$ = (1 2 3 4 7 6 9 8 5). Tương tự, ta có: : $o_2$ = (4 1 2 8 5 6 7 3 9).

Như vậy, phép trao đổi chéo chu trình bảo toàn được vị trí tuyệt đối của các thành phố trong các hành trình của các cá thể cha/mẹ.

Bài toán chu trình Hamilton sẽ dụng thuật toán bằng cách sau:

1. Cho mỗi cạnh có sẵn của đồ thị G cho trước một độ dài 1.
2. Thêm vào đồ thị cho trước G các cạnh không có trong nó, và cho chúng độ dài 2. Gọi đồ thị thu được là đồ thị G'.
3. Thực hiện thuật toán đã mô tả trên để tìm một lời giải tối ưu cho bài toán người du lịch trong đồ thị G'.
4. So sánh độ dài l(C) của chu trình Hamilton thu được trong G' với n (số các đỉnh của G). Nếu l(C)=n thì C là chu trình Hamilton trong G, còn nếu l(C)>n thì C không phải là chu trình Hamilton trong G.

Ta biết rằng thuật giải di truyền áp dụng cho bài toán người du lịch cho kết quả trong thực tiễn khá sát. Trong trường hợp bài toán này thì các trọng số các cạnh đều là số tự nhiên , cụ thể là số 1 và 2 mà thôi. Như vậy, chỉ cần với một độ chính xác nhỏ đủ sai lệch giữa phí tổn tìm thấy so với phí tổn tối ưu thật sự không quá 1, thì phương pháp cho ta sự kết quả đúng là tồn tại hay không tồn tại chu trình Hamilton trong đồ thị đã cho. Các tác giả của báo cáo này đang hoàn thiện một chương trình máy tính cụ thể để thể hiện thuật toán nói trên và cũng sẽ mở rộng thuật toán này thành thuật toán tìm đường đi Hamilton nối 2 đỉnh tùy ý cho trước trong đồ thị và sẽ cho ra mắt độc giả một ngày gần đây.

## TÀI LIỆU THAM KHẢO


[1] De Jong K.A., "*On using genetic algorithms to seatch program spaces*", in Proc. 2nd Int. conf. On Genetic Alogrithms and Their Applications, Hillsdale, NJ: lawrence Erbaum, phương pháp 210-216 (1987).

[2] Vũ Đình Hòa & Do Như An, "*Recognizing Dominating Cycles is NP-hard*", Tạp chí tin học và Điều khiển học, T. 18, S. 3 (2002), 223-227.

[3] Vũ Đình Hòa, Đỗ Như An và Nguyễn Hữu Xuân Trường, "*Bao đóng của đồ thị và thuật toán tìm chu trình Hamilton*", Tạp chí tin học và Điều khiển học.

[4] Vu Dinh Hoa, "*A sufficient condition for hamiltonian cycles in tough graphs*", Vietnam Journal of Mathematics 23 (1995), 57-67.

[5] Vu Dinh Hoa, "*Path covering number and hamiltonicity in tough graphs*", Discrete Mathematics 164 (1997) 291-294.

[6] Vu Dinh Hoa, *"Long cycles and Neighborhood Union in 1-tough graphs with large degree sums"*, Discussiones Mathematicae Vol. 18, No. 1 (1998), 1-13.

[7] NICHOLL,D.S.T., *"Gentechnische Methoden"*, Spektrum, Heidelberg 1995 (engl. Original 1994)

[8] PAUN,G.;ROZENBERG,G.;SALOMAA,A., *"DNA Computing"*, New Computing Paradigms. Springer, Berlin 1998

[9] GUSFIELD,D., *"Algorithms on Strings, Trees, and Sequences"*, Computer Science and Computational Biology. Cambridge University Press, 1997

[10] Setubal,J.;Meidanis,J., *"Introduction to Computational Molecular Biology"*, PWS, Boston, 1997.

[11] WATERMAN,M.S., *"Introduction to Computational Biology"*, Chapman&Hall, London, 1995.

[12] Erbsubstanz DNA. Reihe, *"Spektrum der Wissenschaft: Verstọndliche Forschung"*, 3.Auflage, Heidelberg 1988.

[13] Digest (Nr.6), *"Gene und Genome. Spektrum der Wissenschaft"*, Heidelberg 1997.

[14] CIESLIK,D., *"Mathematische Probleme der Genomanalyse"*, Greifswald 1998