

# Giải thuật di truyền

---

TS. NGUYỄN ĐÌNH HÓA

# Giới thiệu

---

*Giải thuật di truyền (Genetic Algorithm: GA)* là kỹ thuật chung giúp giải quyết vấn đề bằng cách mô phỏng sự tiến hóa của con người hay của sinh vật nói chung trong một số điều kiện ràng buộc nào đó.

Mục tiêu của GA không đưa ra lời giải chính xác mà đưa ra lời giải tương đối tối ưu.

Cách tiếp cận của GA:

- Trừu tượng hoá và mô phỏng quá trình thích nghi trong hệ thống tự nhiên
- Thiết kế phần mềm, chương trình mô phỏng, nhằm duy trì các cơ chế quan trọng của hệ thống tự nhiên

# Một số khái niệm cơ bản

Trong tự nhiên	Trong giải thuật	Hình thức thể hiện
Quần thể (population)	Tập hợp các giải pháp cho bài toán	Tập hợp NST ((011001, 00000, 11111)
Cá thể (individual)	Một giải pháp cho bài toán	
Sự thích nghi (fitness)	Chất lượng của giải pháp	
Nhiễm sắc thể (NST)	Mã hóa cho một cá thể, là một chương trình thực hiện giải pháp	Tập hợp các bit, ví dụ [101100]
Gen (gene)	Một phần của chương trình thực hiện	Hệ đếm (nhị phân, thập phân,...)
Thế hệ (generation)	Một vòng lặp của chương trình	

# Một số khái niệm cơ bản

---

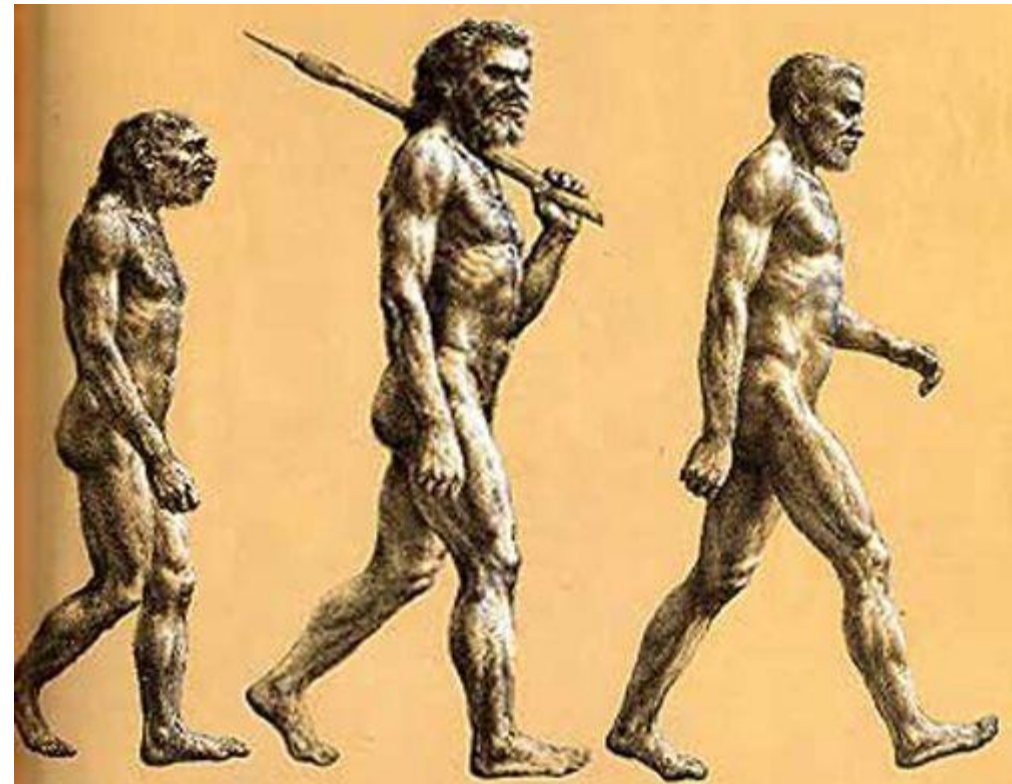
Thế hệ (generation) là một quần thể (population) của các cá thể (individuals) đã được tiến hóa đến một thời điểm nào đó.

Giải thuật di truyền mô phỏng sự tiến hóa trong tự nhiên, cho phép chọn lọc các cá thể của một thế hệ và tái sinh chúng để tạo ra các thế hệ mới.

# Chọn lọc tự nhiên

---

*"Select The Best, Discard The Rest"*



# Ví dụ - chọn lọc tự nhiên

---

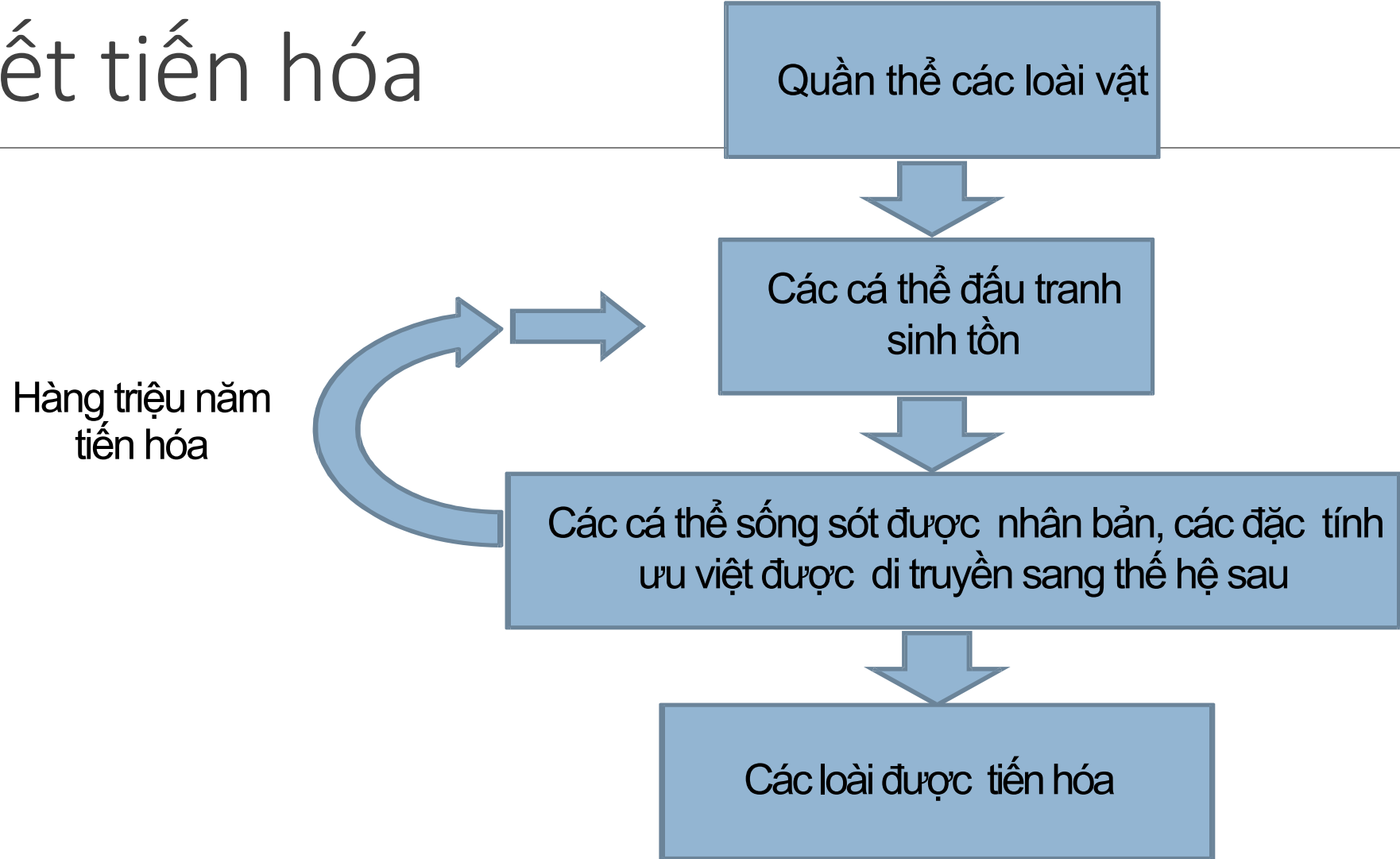
Hươu cao cổ:

- Cổ dài giúp chúng ăn được lá cây trên cao
- Chúng có nhiều cơ hội sống sót trong môi trường khắc nghiệt
- Những đặc tính ưu việt có xu hướng di truyền từ đời này sang đời khác, và ngày càng hoàn thiện một cách tự nhiên



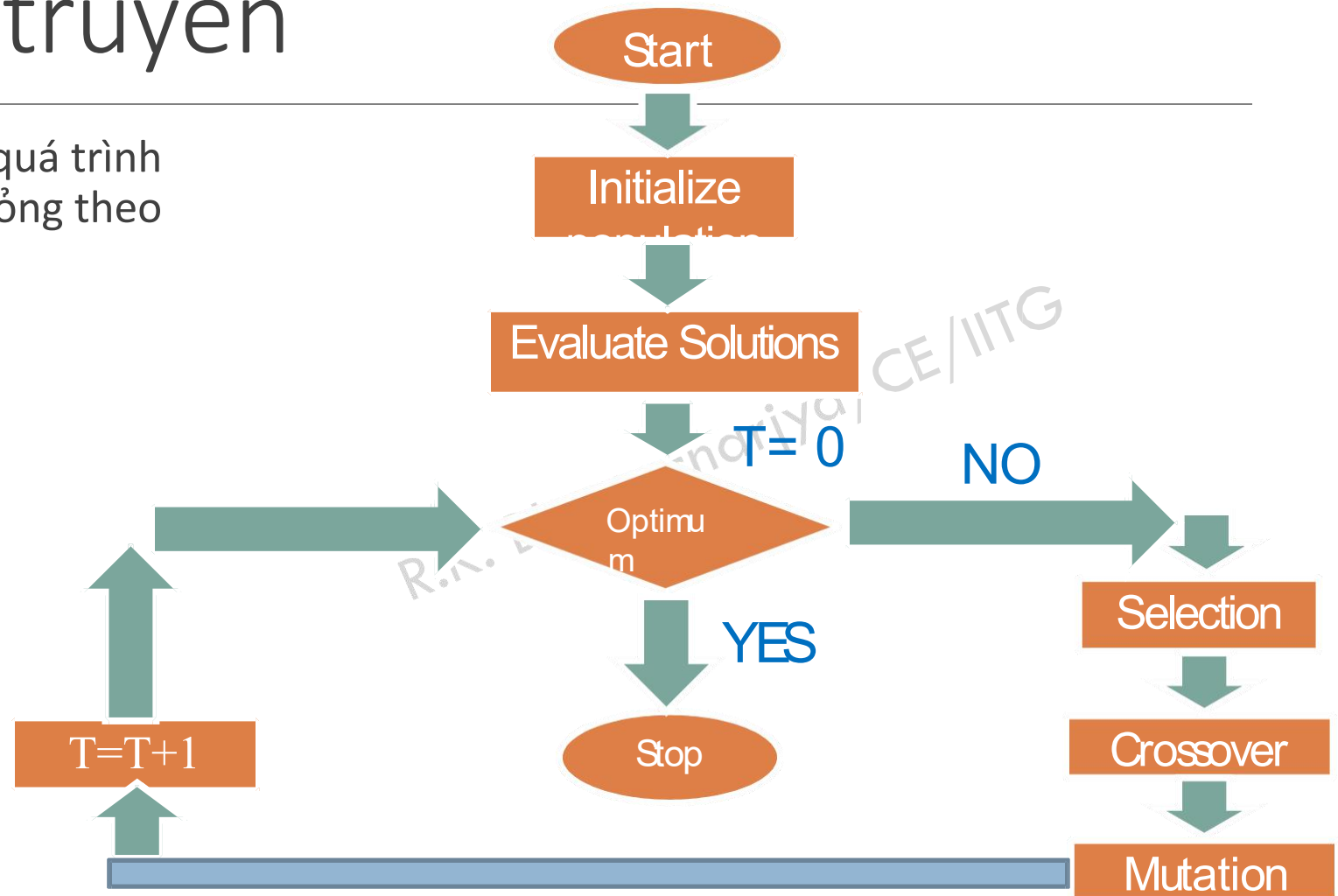
# Thuyết tiến hóa

---



# Giải thuật di truyền

Giải thuật di truyền thực hiện quá trình tối ưu hóa theo hướng mô phỏng theo thuyết tiến hóa tự nhiên



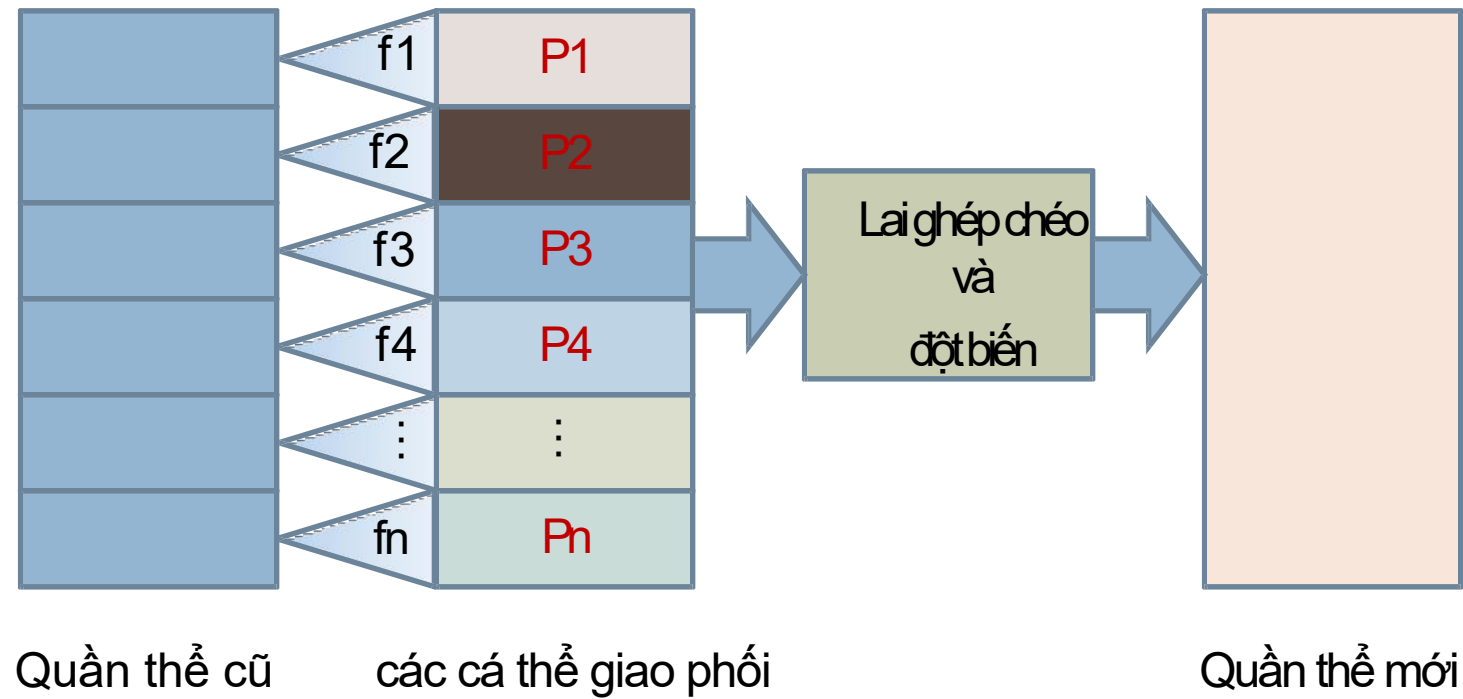


# Giải thuật di truyền

---

```
function sga(){  
    Initialize population;  
    Calculate fitness function;  
    While(fitness value != termination criteria)  
    {  
        Selection;  
        Crossover;  
        Mutation;  
        Calculate fitness function;  
    }  
}
```

# Giải thuật di truyền



# Chọn lọc

---

Quá trình lựa chọn cá thể / giải pháp phù hợp để tồn tại, và nhân bản các giải pháp tốt, loại bỏ các giải pháp tồi

Nguyên tắc: đảm bảo kích thước quần thể không đổi

“Lựa chọn những thứ tốt nhất, loại bỏ những thứ tồi nhất”

# Hàm thích nghi

---

Hàm thích nghi là công cụ đánh giá độ thích nghi của từng giải pháp (cá thể) đối với vấn đề cần giải quyết.

Các giải pháp (cá thể) được xếp hạng dựa trên độ thích nghi của chúng.

Độ thích nghi càng cao thì giải pháp (cá thể) càng dễ được lựa chọn để tồn tại.

$$eval(v_i) = \frac{f(v_i)}{\sum_{i=1}^{kích\,thước\,quần\,thể} f(v_i)}$$

Trong đó:  $f(v_i)$  là giá trị hàm mục tiêu của giải pháp  $v_i$

Tổng giá trị thích nghi của quần thể:  $F = \sum_{i=1}^{Kích\,thước\,quần\,thể} eval(v_i)$

# Hàm thích nghi

---

Xác suất được lựa chọn  $p_i$  của mỗi cá thể  $v_i$

$$p_i = \frac{eval(v_i)}{\sum_{i=1}^{Kích\ thước\ quần\ thể} eval(v_i)}$$

Xác suất tích lũy:

$$q_i = \sum_{j=1}^i p_j$$

# Hàm thích nghi

---

Độ thích nghi tuyến tính:  $f(v_i)' = a * f(v_i) + b$

Với các tham số a, b được lựa chọn dựa vào quần thể.

Độ thích nghi sigma:  $f(v_i)' = f(v_i) + (\overline{f(v_i)} - c * \sigma)$

Với tham số c là số nguyên, thường chọn từ 1 đến 5,  $\sigma$  là độ lệch chuẩn của quần thể. Nếu  $f'$  âm thì được thiết lập bằng 0.

Độ thích nghi theo lũy thừa:  $f(v_i)' = f(v_i)^k$

Với tham số k thường được chọn gần bằng 1.

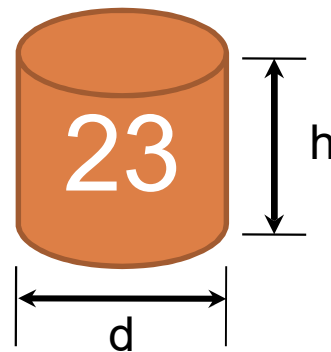
# Ví dụ về hàm thích nghi

Tìm các kích thước của hình trụ tròn (đường kính đáy  $d$  và chiều cao  $h$ ) sao cho diện tích bề mặt là nhỏ nhất, và thể tích không được nhỏ hơn 300 đơn vị thể tích.

$$\begin{aligned} \text{Minimize} \quad & f(d, h) = c((\pi d^2/2) + \pi dh), \\ \text{Subject to} \quad & g_1(d, h) \equiv (\pi d^2 h/4) \geq 300, \\ \text{Variable bounds} \quad & d_{\min} \leq d \leq d_{\max}, \\ & h_{\min} \leq h \leq h_{\max}. \end{aligned}$$

Considering  $c = 0.0654$

$$\begin{aligned} F(s) &= 0.0654(\pi(8)^2/2 + \pi(8)(10)), \\ &= 23, \end{aligned}$$



# Các hàm lựa chọn

---

Có nhiều cách lựa chọn khác nhau trong giải thuật di truyền

- Tournament selection
- Roulette wheel selection
- Proportionate selection
- Rank selection
- Steady state selection
- ...



# Tournament selection

---

Một nhóm các cá thể (giải pháp) được lựa chọn một cách ngẫu nhiên vào nhóm.

Các cá thể trong mỗi nhóm sẽ ganh đua nhau, chọn ra cá thể mạnh nhất (giải pháp tốt nhất)

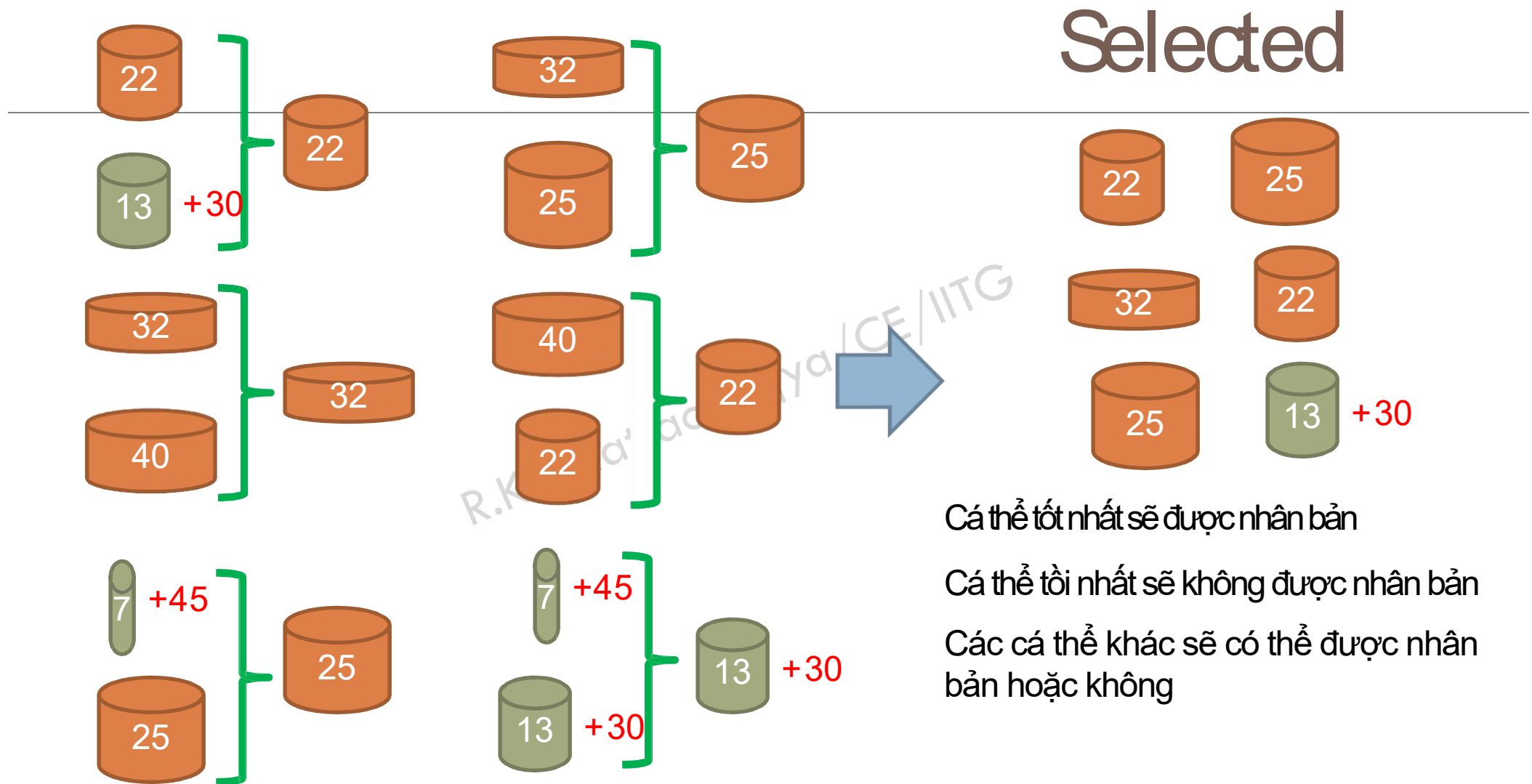
Kích thước nhóm, và hình thức ganh đua trong mỗi nhóm có thể được tùy chỉnh

Các cá thể chiến thắng trong mỗi nhóm sẽ được lựa chọn để tiến hóa trong các giai đoạn tiếp theo

Các cá thể yếu trong mỗi nhóm sẽ ít có cơ hội chiến thắng nếu kích thước nhóm là lớn

Hai cách thực hiện cơ bản

- Lựa chọn có thay thế
- Lựa chọn không thay thế



# Roulette wheel

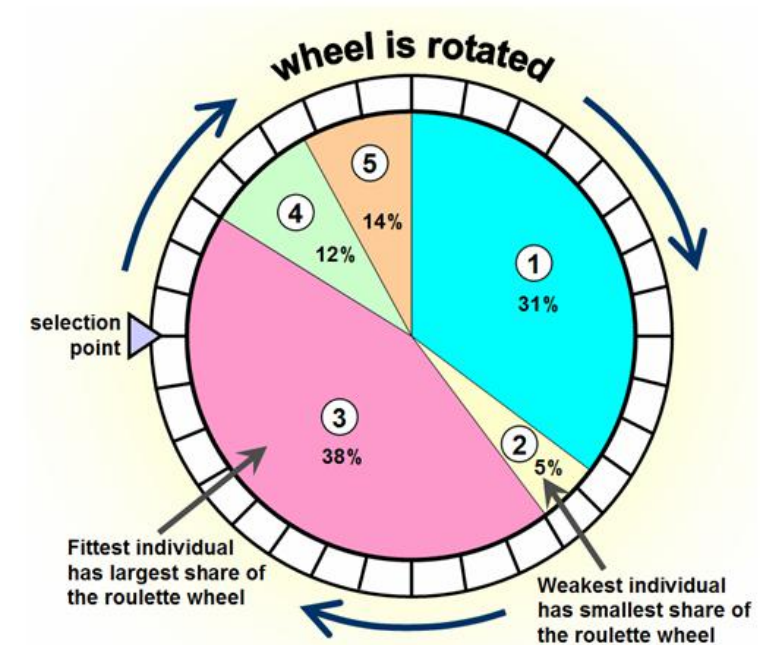
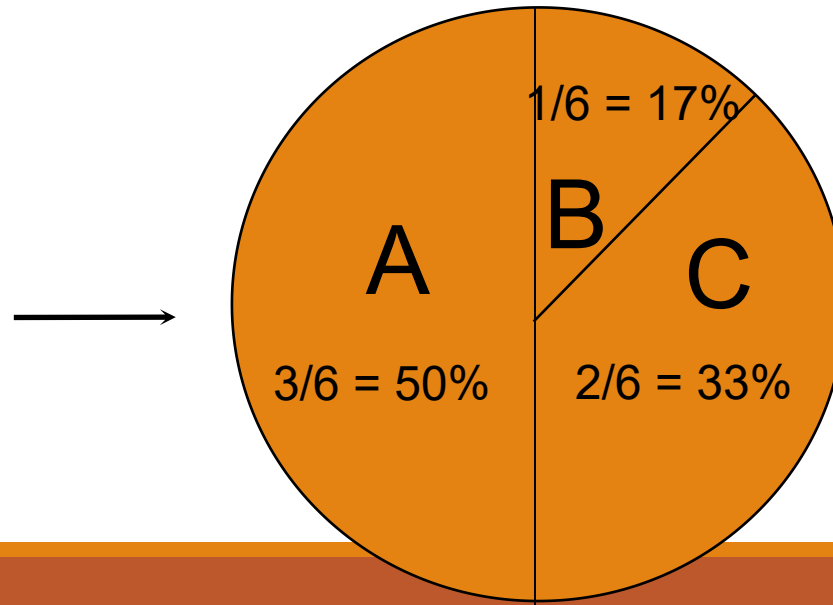
Nguyên tắc: các cá thể tốt sẽ có nhiều cơ hội được chọn lựa hơn

- Cơ hội lựa chọn tỉ lệ thuận với độ thích nghi
- Kỹ thuật Roulette wheel:
  - Mỗi cá thể được gán một phần diện tích trên Roulette wheel
  - Quay Roulette wheel n lần để chọn ra n cá thể

fitness(A) = 3

fitness(B) = 1

fitness(C) = 2



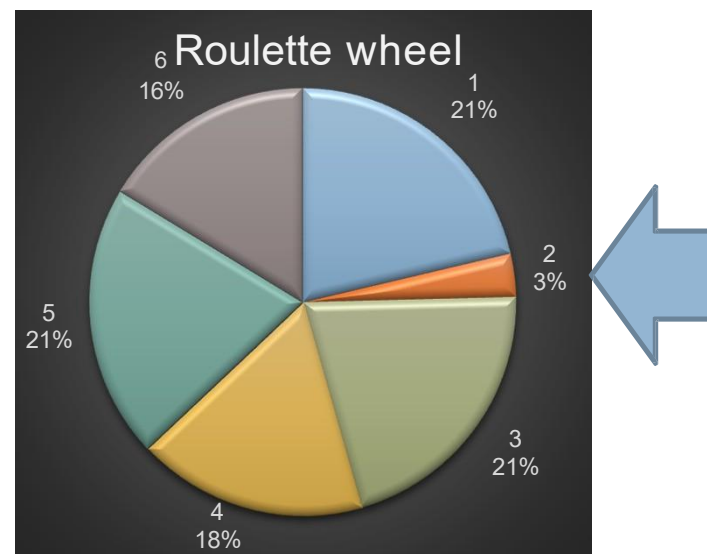
# Roulette wheel và Proportionate Selection

Mỗi cá thể sẽ tương ứng với một giá trị thích nghi

Các cá thể có giá trị thích nghi cao sẽ dễ được chọn lựa làm “cha” “mẹ” để tạo ra các cá thể mới

Lựa chọn các cá thể “cha” “mẹ” dựa vào Roulette wheel

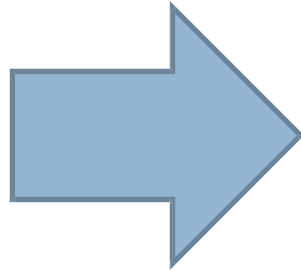
NST	Giá trị thích nghi	% trên RW	EC	AC
1	50	26.88	1.61	2
2	6	3.47	0.19	0
3	36	20.81	1.16	1
4	30	17.34	0.97	1
5	36	20.81	1.16	1
6	28	16.18	0.90	1
Tổng	<b>186</b>	<b>100.00</b>	<b>6</b>	<b>6</b>



# Rank Selection

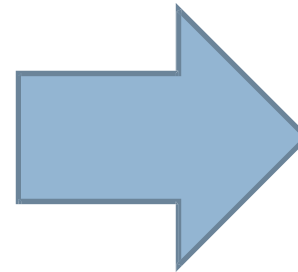
Chrom #	Fitness
1	37
2	6
3	36
4	30
5	36
6	28

Sắp xếp dựa  
vào giá trị  
thích nghi



Chrom #	Fitness
1	37
3	36
5	36
4	30
6	28
2	6

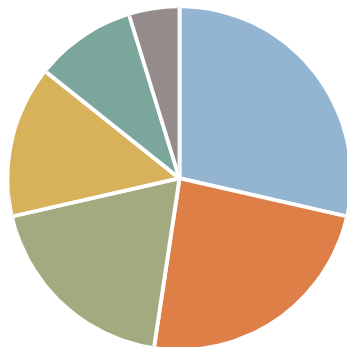
Xếp  
hạng



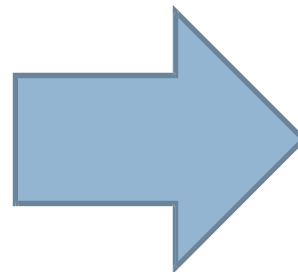
Chrom #	Rank
1	6
3	5
5	4
4	3
6	2
2	1

Chrom #	% of RW
1	29
3	24
5	19
4	14
6	10
2	5

Roulette wheel



6 5 4 3 2 1



Chrom #	EC	AC
1	1.714	2
3	1.429	1
5	1.143	1
4	0.857	1
6	0.571	1
2	0.286	0

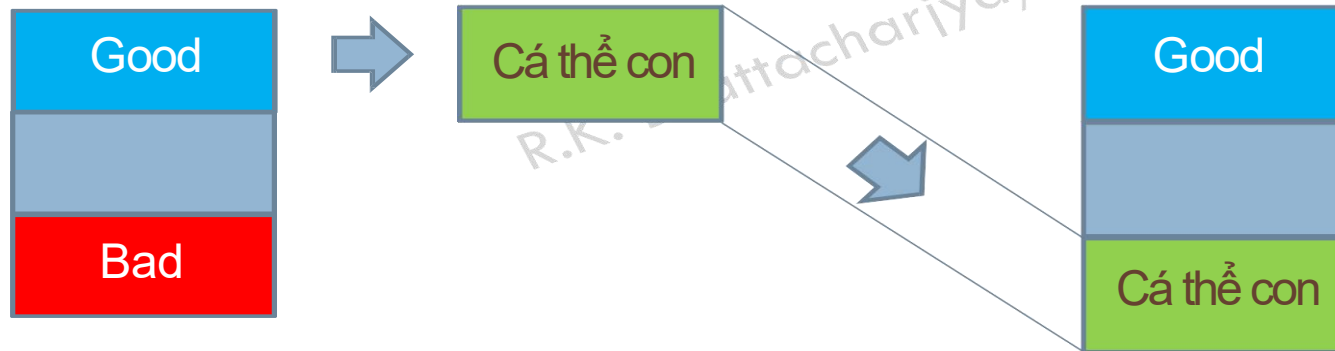
# Steady State selection

---

Mỗi chu kỳ sẽ sinh ra một vài cá thể con từ các NST tốt

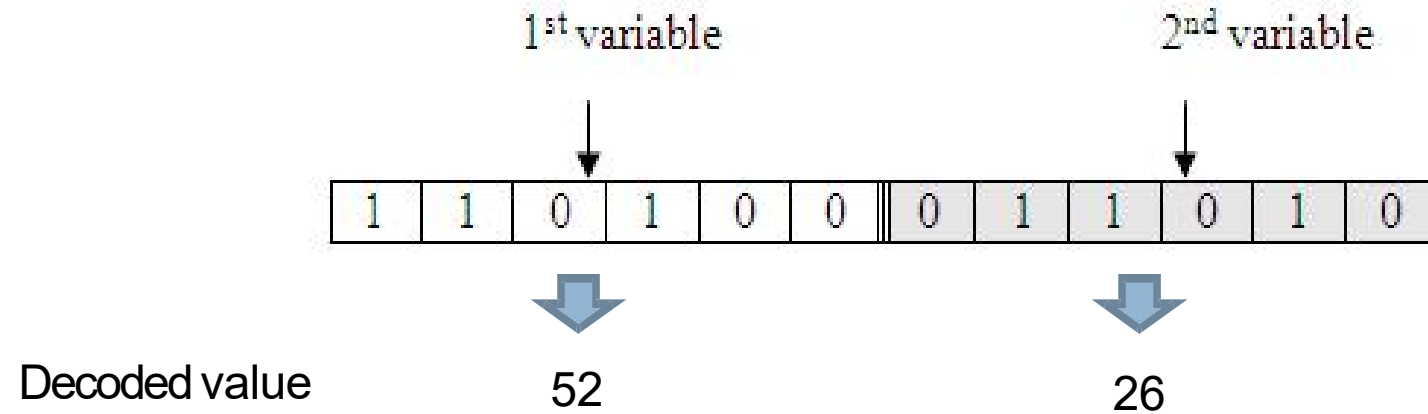
Các cá thể tồi sẽ bị loại bỏ, và thay thế bằng các cá thể con

Các cá thể này tiếp tục được tiến hóa theo các chu kỳ tiếp theo



# Mã hóa NST

Sử dụng mã nhị phân: các NST được mã hóa thành các chuỗi bit, mỗi bit đại diện cho một đặc tính nào đó



Mapping between decimal  
and binary value

$$x_i = x_i^{\min} + \frac{x_i^{\max} - x_i^{\min}}{2^{l_i} - 1} \text{DV}(s_i)$$

# Mã hóa nhị phân

---

Minimize  $f(d, h) = c((\pi d^2 / 2) + \pi dh)$ ,

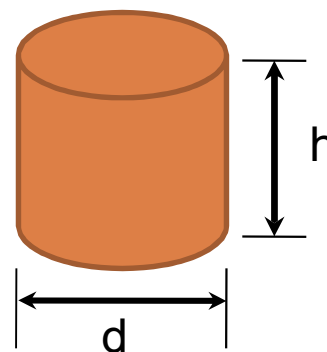
Subject to  $g_1(d, h) \equiv (\pi d^2 h / 4) \geq 300$ ,

Variable bounds  $d_{\min} \leq d \leq d_{\max}$ ,

$h_{\min} \leq h \leq h_{\max}$ .

Định nghĩa chuỗi bit: [0100001010]

d h



$(d, h) = (8, 10) \text{ cm}$

NST = [0100001010]

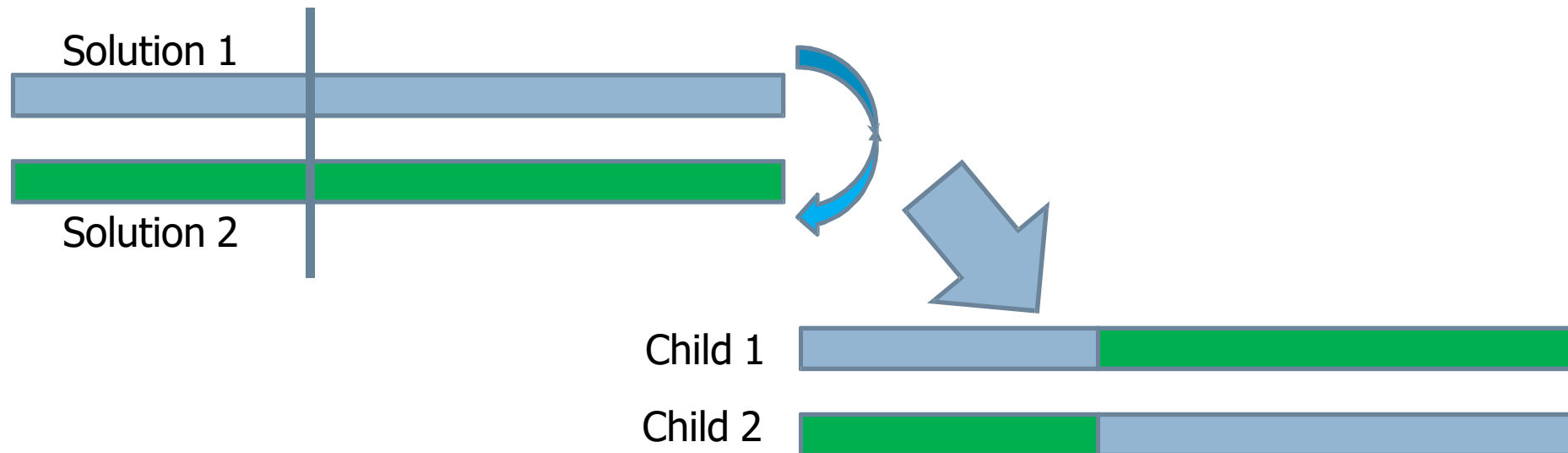


# Lai ghép chéo 1 điểm

Khi các cá thể “cha” “mẹ” được lựa chọn, chúng được lai ghép với nhau để tạo ra các cá thể mới

Lai ghép chéo: xác suất thực hiện lai ghép chéo là  $P_c$ , thông thường nằm trong khoảng (0.6, 0.9)

- Chọn một điểm ngẫu nhiên trên NST của cha, mẹ.
- Trao đổi các phần NST của cha – mẹ sau điểm lựa chọn đó
- Kết quả tạo ra các cá thể con



# Lai ghép chéo 1 điểm

---

Các gen nằm sát nhau có xu hướng được giữ nguyên trên cá thể con sau khi lai ghép

Các gen nằm ngược đầu nhau trên bộ NST sẽ có xu hướng nằm ở các cá thể con khác nhau

2 vấn đề trên được gọi là thiên vị về vị trí (positional bias)

Đặc điểm này có thể được khai thác trong các bài toán mà chúng ta biết rõ về cấu trúc, giúp tìm ra các giải pháp tiên định phù hợp

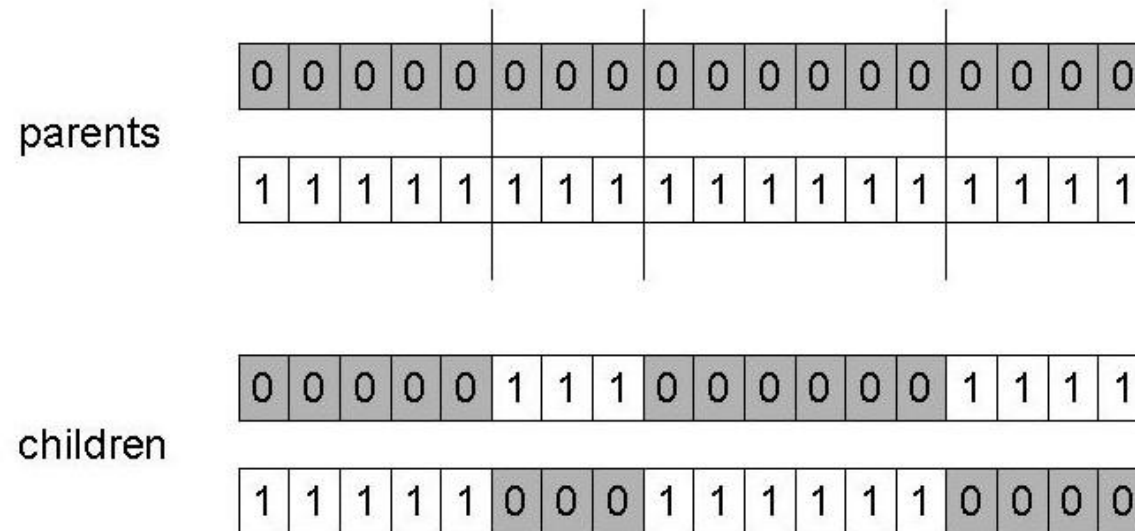
# Lai ghép chéo nhiều điểm

Chọn ngẫu nhiên n điểm để thực hiện ghép chéo

Chia NST các cá thể cha/mẹ tại các điểm này

Ghéo chéo các đoạn NST cha/mẹ giữa các điểm đã chọn

Quá trình này vẫn có tính thiên vị theo vị trí



# Lai ghép chéo ngẫu nhiên

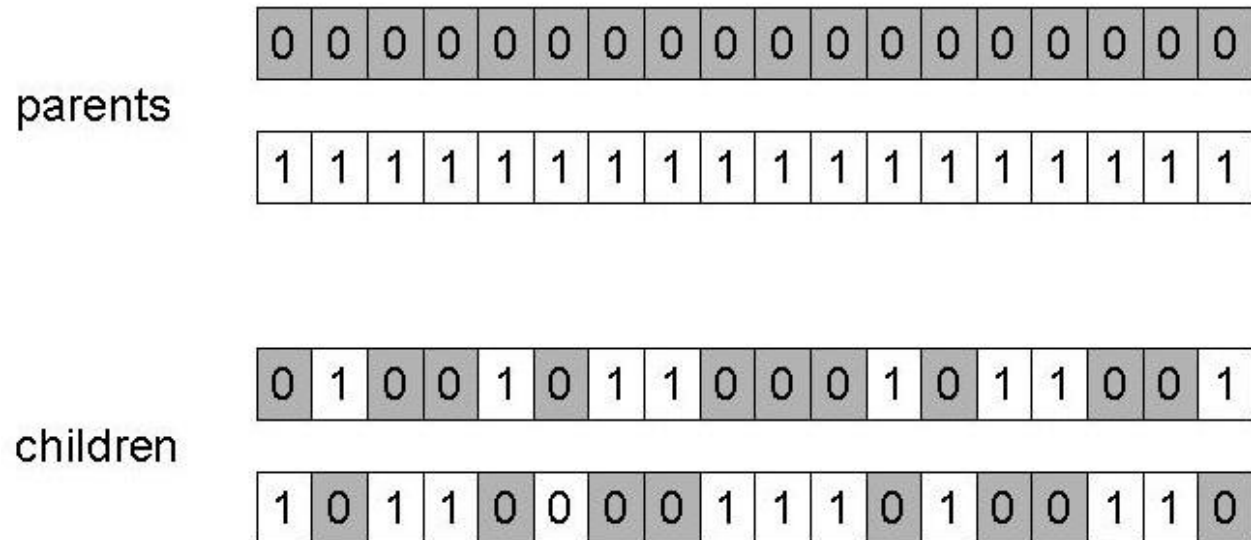
---

Lấy điểm đầu của NST cha và điểm cuối của NST mẹ để làm mốc tạo các thể con.

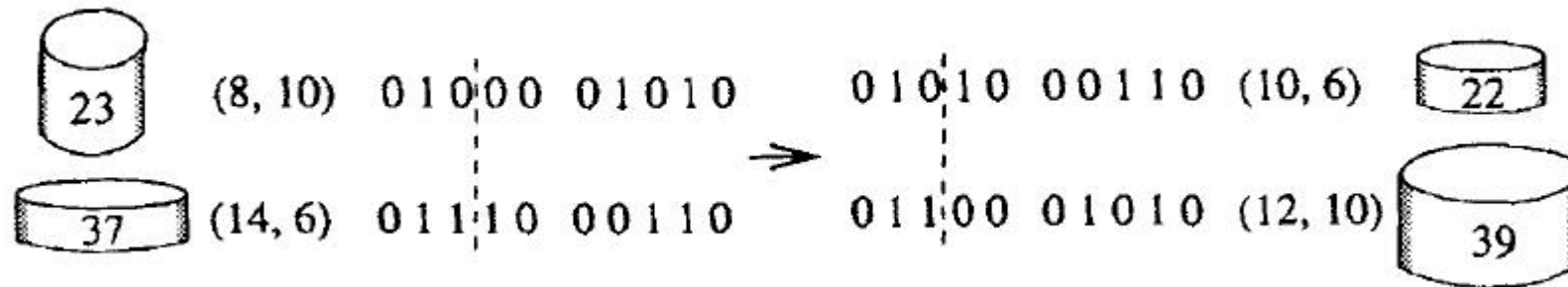
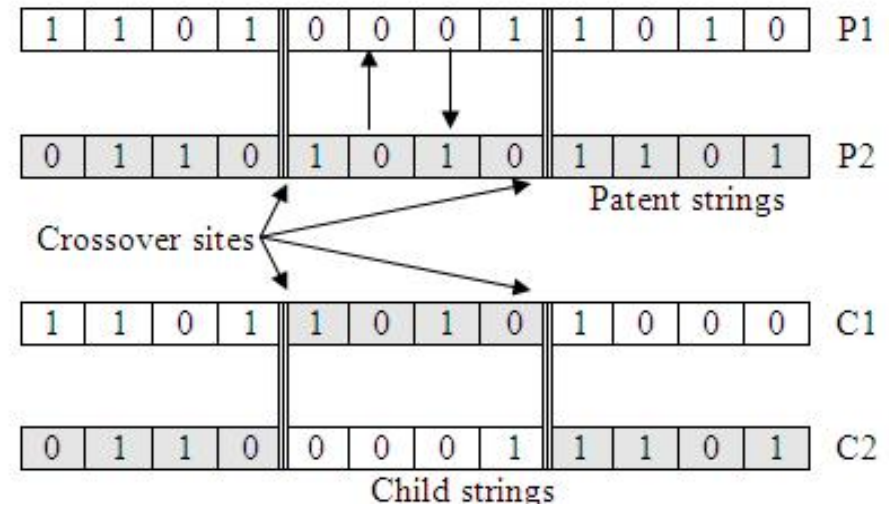
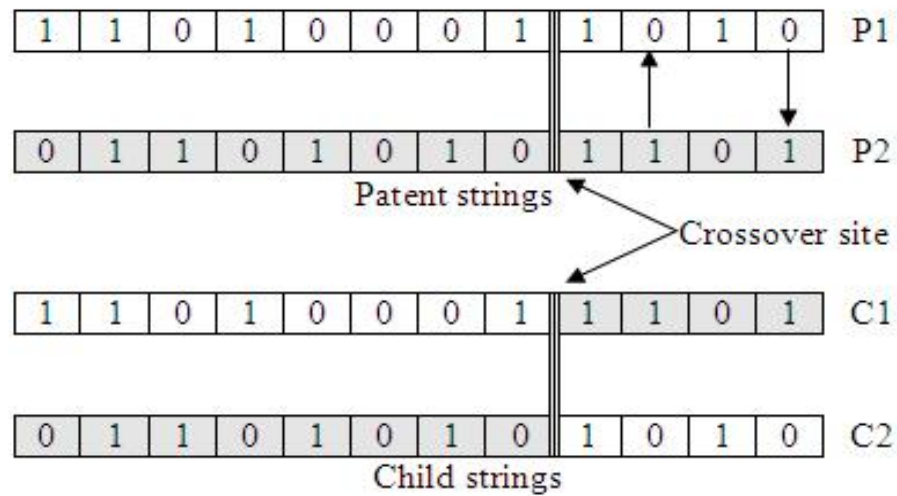
Các phần NST còn lại của cá thể con được lai ghép một cách ngẫu nhiên giữa cha và mẹ

Cá thể con thứ 2 được tạo ra theo các phần ghép còn lại của cha và mẹ

Quá trình kế thừa được thực hiện độc lập, không xoay quanh điểm cố định nào



# Ví dụ về lai ghép chéo



# Lai ghép chéo với các giá trị thực

---

## Rời rạc:

- Mỗi giá trị gen của cá thể con  $z$  được lấy từ một trong số các giá trị của cha mẹ  $(x, y)$  với xác suất lựa chọn bằng nhau:  $z_i = x_i$  or  $y_i$
- Quá trình có thể được thực hiện trên  $n$ -điểm hoặc thực hiện đồng đều

## Trung bình

- Tạo ra các NST con mang giá trị nằm giữa các giá trị NST của cha mẹ (kết hợp theo đại số)
- $z_i = \alpha x_i + (1 - \alpha) y_i$  với  $\alpha : 0 \leq \alpha \leq 1$ .
- Tham số  $\alpha$  có thể được xác định theo các cách:
  - Hằng số cho trước: tạo ra lai ghép chéo đồng đều
  - Biến số (phụ thuộc vào tuổi của các thế hệ)
  - Lấy ngẫu nhiên

# Lai ghép chéo số học trên 1 điểm

- NST cha:  $\langle x_1, \dots, x_n \rangle$  và  $\langle y_1, \dots, y_n \rangle$
- Chọn ngẫu nhiên 1 gen ( $k$ ),
- Con<sub>1</sub> có NST là:  $\langle x_1, \dots, x_k, \alpha \cdot y_k + (1 - \alpha) \cdot x_k, \dots, x_n \rangle$
- Con còn lại được tạo ra từ giá trị ngược lại.
- Ví dụ: với  $\alpha = 0.5$

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.5	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

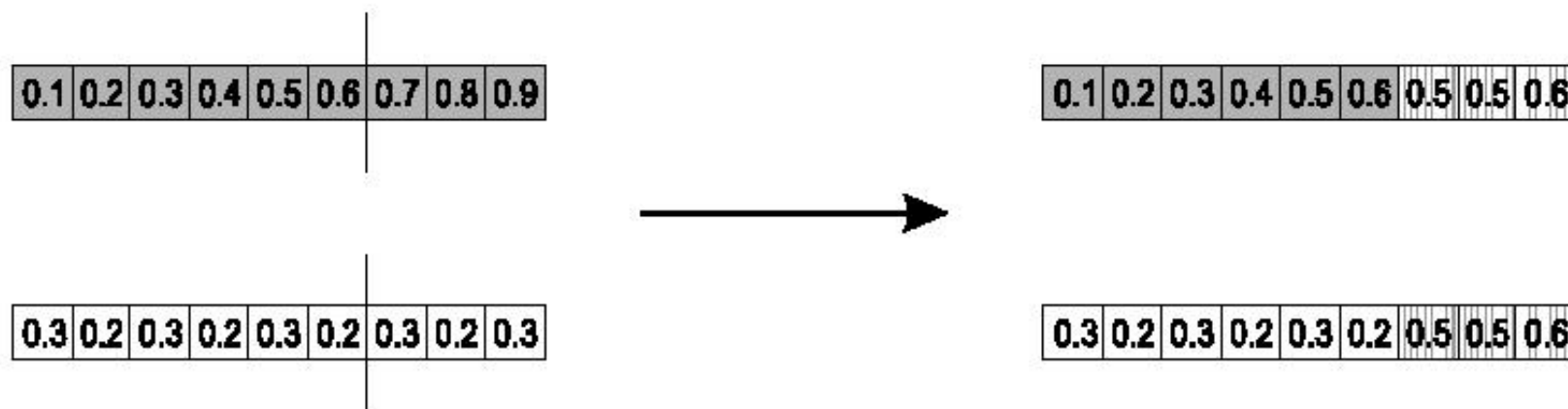


0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.5	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

# Lai ghép chéo số học đơn thuần

- Các NST cha mẹ:  $\langle x_1, \dots, x_n \rangle$  và  $\langle y_1, \dots, y_n \rangle$
- Chọn một gen ngẫu nhiên ( $k$ ), các giá trị gen của NST con được tạo ra bằng cách trộn NST cha mẹ sau điểm  $k$  này
- Con<sub>1</sub> có NST là:  $\langle x_1, \dots, x_k, \alpha \cdot y_{k+1} + (1 - \alpha) \cdot x_{k+1}, \dots, \alpha \cdot y_n + (1 - \alpha) \cdot x_n \rangle$
- Con còn lại có NST được tạo ra theo các giá trị ngược lại
- Ví dụ: với  $\alpha = 0.5$





# Lai ghép số học toàn bộ NST

---

- Đây là cách hay thực hiện nhất
- NST cha mẹ:  $\langle x_1, \dots, x_n \rangle$  và  $\langle y_1, \dots, y_n \rangle$
- Con<sub>1</sub> có NST là:  $a \cdot \bar{x} + (1 - a) \cdot \bar{y}$
- NST của con còn lại được thực hiện với giá trị ngược lại
- Ví dụ: với  $\alpha = 0.5$

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----



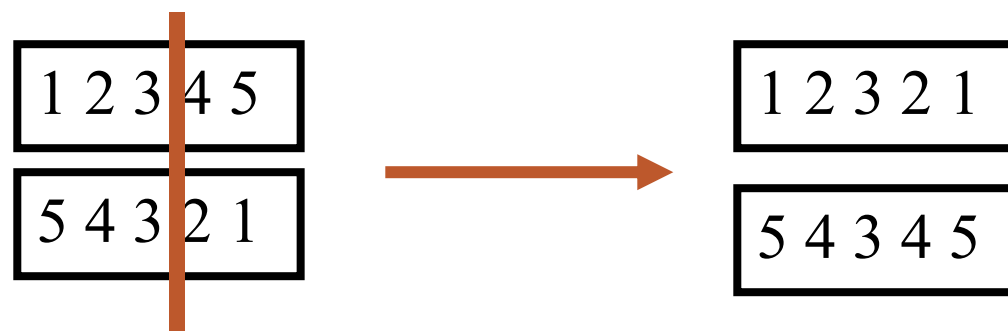
0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----

# Lai ghép chéo hoán vị

---

Lai ghép chéo thường có thể tạo ra các giải pháp bất ngờ



Các phương pháp lai ghép chéo mới có thể được sử dụng, trong đó có sự kết hợp thông tin về các gen liền kề của NST cha mẹ

# Lai chéo chéo Order 1

---

Ý tưởng là bảo toàn thứ tự của các gen

Thủ tục thực hiện:

1. Chọn một phần ngẫu nhiên của NST cha mẹ
2. Copy phần này của NST cha vào NST của con thứ nhất
3. Copy các giá trị gen chưa có trong phần NST nói trên vào phần còn lại của NST con:
  - Bắt đầu từ điểm cắt của phần NST đã chọn,
  - Sử dụng thứ tự sắp xếp của NST mẹ
  - Ghép tuần tự đến hết phần NST còn trống
4. Làm tương tự với con thứ hai, đảo vai trò của NST cha và mẹ

# Ví dụ về lai ghép chếp Order 1

---

Copy đoạn NST ngẫu nhiên của cha vào NST con thứ nhất

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



			4	5	6	7		
--	--	--	---	---	---	---	--	--

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

Copy phần còn lại của NST mẹ theo thứ tự 1,9,3,8,2

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



3	8	2	4	5	6	7	1	9
---	---	---	---	---	---	---	---	---

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

# Lai ghép chéo chiếu từng phần

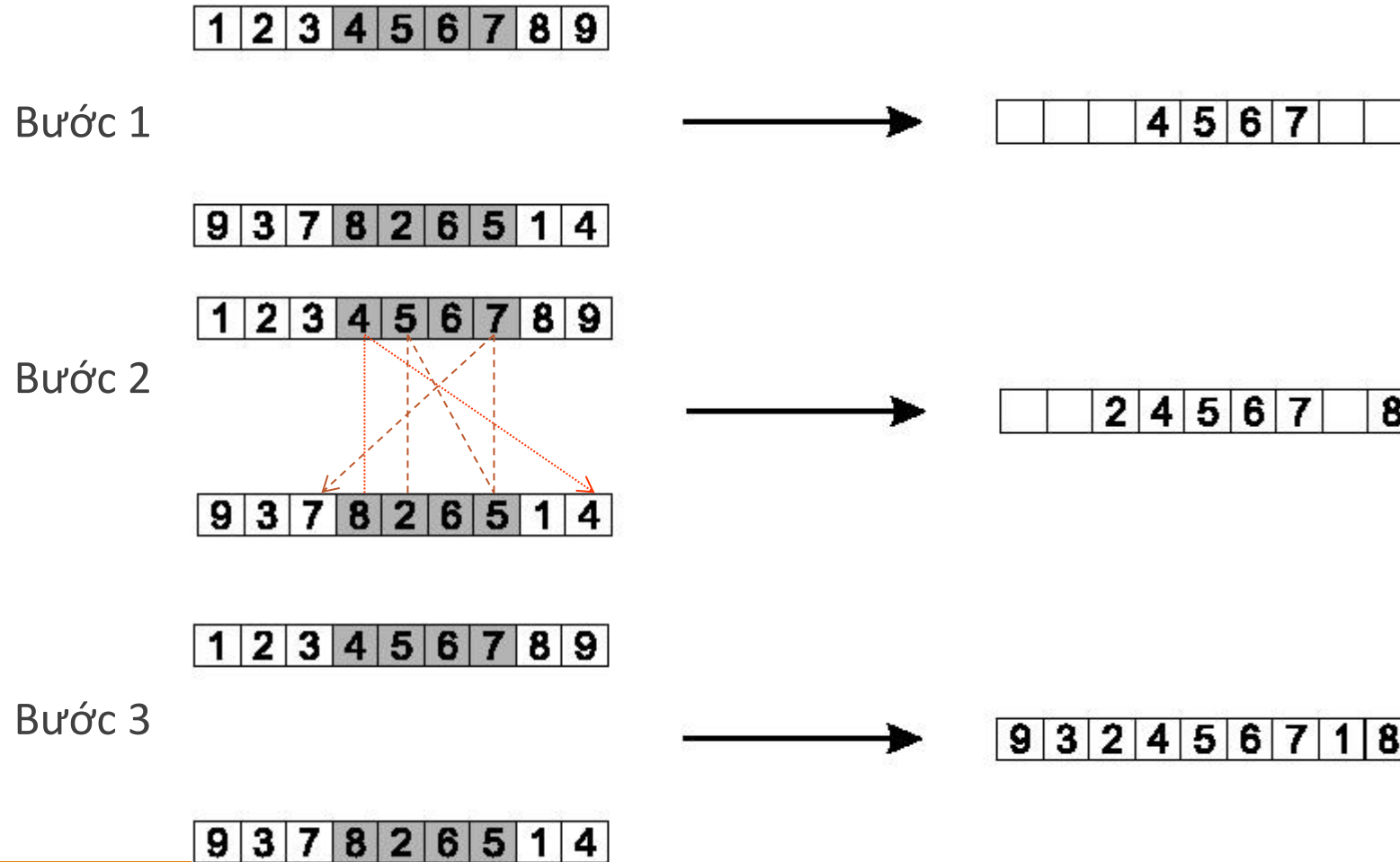
---

Gọi NST của cha và mẹ lần lượt là P1 và P2:

1. Chọn ngẫu nhiên một đoạn NST của P1 rồi copy đoạn đó vào NST con
2. Bắt đầu từ điểm ghép chéo, tìm các gen có trong P2 mà chưa có trong NST con
3. Với mỗi gen  $i$  của P2: kiểm tra trong NST con xem gen  $j$  nào đã được copy vào vị trí của nó trong đoạn NST của P1
4. Gán gen  $i$  vào vị trí của gen  $j$  đã bị chiếm vị trí trong P2, vì gen  $j$  đã nằm ở vị trí  $i$  trong NST con rồi
5. Nếu vị trí của gen  $j$  trong P2 đã được sử dụng trong NST của con bởi gen  $k$ , gán gen  $i$  vào vị trí  $k$  trong P2
6. Các gen còn lại của NST P2 sẽ lần lượt được điền nốt vào các vị trí còn trống của NST con.

NST của con thứ hai sẽ được hình thành tương tự với sự hoán đổi vai trò P1 và P2

# Ví dụ về lai ghép chéo chiều từng phần



# Lai ghép chéo xoay vòng

---

Mỗi gen của cha mẹ được sao chép cùng thông tin về vị trí của nó

Thủ tục thực hiện:

1. Tạo một vòng các gen từ cha P1:

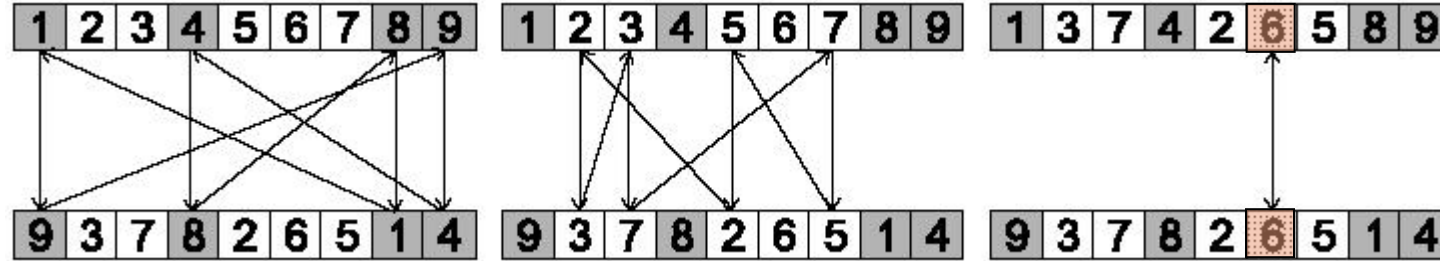
- (a) Bắt đầu với gen đầu tiên của P1.
- (b) Soi đến gen có cùng vị trí trong P2.
- (c) Nhảy đến vị trí có cùng gen với P2 ở trong P1
- (d) Điền gen này vào vòng gen.
- (e) Lặp lại từ bước b đến d đến khi gặp được gen đầu tiên của P1.

2. Điền các gen trong vòng nói trên vào NST con thứ nhất theo thứ tự của các gen đó trong P1.

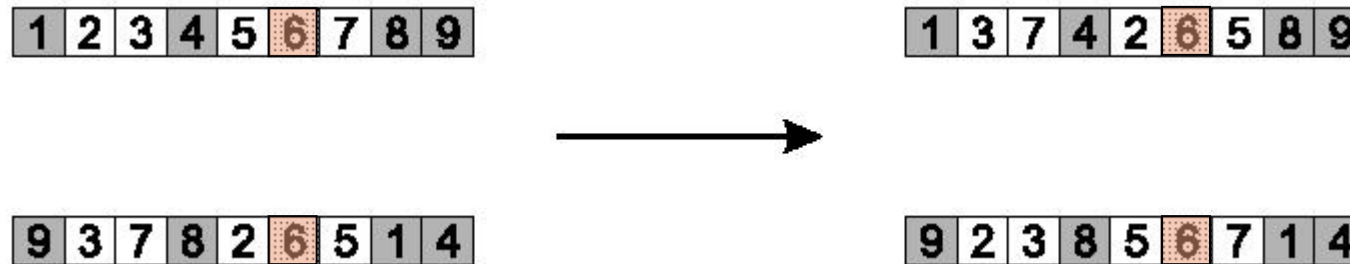
3. Tạo vòng gen thứ hai từ P2 rồi điền tiếp vào NST con

# Lai ghép chéo xoay vòng

Bước 1: xây dựng các vòng gen



Bước 2: copy các vòng gen tương ứng vào NST con





# Đột biến

---

Là hiện tượng tự biến đổi trong gen, thường hiếm gặp trong tự nhiên.

Đột biến giúp cải thiện khả năng tương thích với môi trường của các cá thể con

Đột biến giúp tạo ra sự đa dạng trong giống loài (đa dạng trong giải pháp)

Xác suất xảy ra đột biến là  $P_m$  với giá trị nằm giữa  $1/(\text{kích thước quần thể})$  và  $1/(\text{độ dài NST})$

parent

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

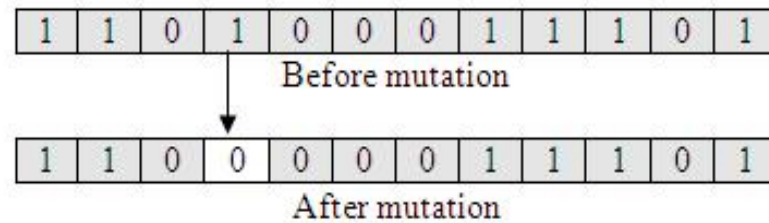
child

0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# Đột biến nhị phân

Là sự chuyển đổi giá trị của gen (0 thành 1 hoặc ngược lại) trong NST với một xác suất xảy ra  $P_m$

Ví dụ:



$$\text{22} \quad (10, 6) \quad 010\bar{1}10 \quad 00110 \Rightarrow 010\bar{0}10 \quad 00110 \quad (8, 6) \quad \text{16}$$

# Đột biến hoán vị

---

Đột biến bình thường có thể tạo ra các giải pháp bất ngờ

- Đột biến theo bit : đổi giá trị gen từ  $i$  thành  $j$
- Đổi giá trị gen từ  $i$  sang giá trị  $k$  có thể dẫn tới  $k$  có thể xuất hiện 2 lần, trong khi một giá trị  $j$  nào đó không bao giờ xuất hiện

Để đột biến hoán vị thực hiện được, cần ít nhất 2 giá trị bị thay đổi

Tham số hoán vị biểu diễn xác suất một toán tử đột biến nào đó có thể được thực hiện trên toàn bộ chuỗi NST chứ không xảy ra trên 1 vị trí gen nào đó

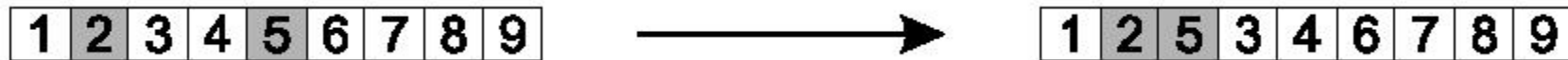
# Đột biến hoán vị - phép chèn

---

Chọn một cặp gen ngẫu nhiên

Chuyển vị trí gen phía sau lên sát gen phía trước, dịch toàn bộ các gen còn lại về phía sau

Phép chèn này bảo toàn hầu hết thứ tự của các gen cùng thông tin lân cận của chúng trong NST



# Đột biến hoán vị - phép đảo

---

Chọn ngẫu nhiên hai gen bất kỳ rồi đảo vị trí của chúng cho nhau

Phép đảo bảo toàn hầu hết các thông tin liên kề của các gen (chỉ có 4 liên kết bị phá vỡ), cho phép đảo thứ tự nhiều hơn phép chèn



# Đột biến hoán vị - phép nghịch đảo

---

Chọn ngẫu nhiên hai gen, rồi đảo ngược chuỗi gen nằm giữa hai vị trí đó.

Phép nghịch đảo bảo toàn hầu hết các thông tin gen liên kề (chỉ phá vỡ 2 liên kết) nhưng tạo ra nhiều sự lộn xộn hơn



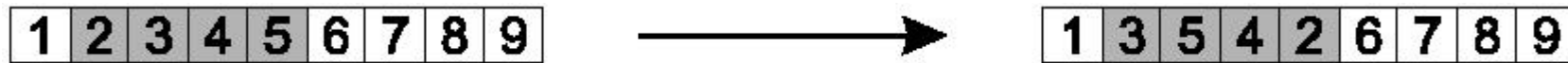
# Đột biến hoán vị - phép trộn

---

Chọn ngẫu nhiên một đoạn NST

Hoán đảo vị trí một cách ngẫu nhiên toàn bộ gen trên đoạn NST đó

Phép trộn có thể tiến hành trên nhiều đoạn NST không liền kề nhau



# Lai ghép chéo và đột biến – cái nào tốt?

---

Lai ghép chéo và đột biến là hai quá trình độc lập, có các vai trò khác nhau trong tiến hóa

Cái nào tốt hơn trong hai quá trình này phụ thuộc vào từng vấn đề cụ thể

Cả hai quá trình cùng được thực hiện sẽ thường có kết quả tốt hơn trong tiến hóa. Tuy nhiên,

**Tiến hóa tự nhiên sẽ khó thực hiện được nếu chỉ có lai ghép chéo**

- Lai ghép chéo là quá trình mang tính thăm dò / tìm kiếm sự kết hợp tốt giữa các cá thể cha mẹ
- Lai ghép chéo không làm thay đổi tỉ lệ các gen đã có (VD. tỉ lệ giữa 0 và 1)

**Nếu chỉ có đột biến thì vẫn tiến hóa được**

- Đột biến là quá trình mang tính khai phá, tạo ra các cá thể hoàn toàn mới gần với cha mẹ của chúng
- Đột biến có thể làm thay đổi tỉ lệ giữa các gen, tạo ra các NST hoàn toàn mới.
- Đột biến là cơ hội tạo ra các giải pháp tối ưu.



# Ví dụ

---

Tìm giá trị lớn nhất của hàm số:  $f(x) = x^2$  với  $x$  mang các giá trị  $\{0,1,...,31\}$

Cài đặt thuật toán GA:

- Biểu diễn nhị phân các kết quả, VD:  $01101 \leftrightarrow 13$
- Chọn kích thước quần thể: giả sử chọn 4 cá thể
- Lai ghép chéo 1 điểm, đột biến ngẫu nhiên
- Lựa chọn bằng Roulette wheel
- Khởi tạo ngẫu nhiên

Các thể hệ giải pháp / cá thể được tạo ra lần lượt như sau

# Thế hệ khởi tạo

---

Khởi tạo ngẫu nhiên thế hệ đầu

Lựa chọn các cá thể cha, mẹ để lai ghép tạo ra thế hệ tiếp theo

String no.	Initial population	$x$ Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

# Lai ghép để tạo ra thế hệ tiếp theo

---

String no.	Mating pool	Crossover point	Offspring after xover	$x$ Value	Fitness $f(x) = x^2$
1	0 1 1 0   1	4	0 1 1 0 0	12	144
2	1 1 0 0   0	4	1 1 0 0 1	25	625
2	1 1   0 0 0	2	1 1 0 1 1	27	729
4	1 0   0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

# Đột biến để tạo thế hệ tiếp theo

---

String no.	Offspring after xover	Offspring after mutation	$x$ Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

# Elitism

---

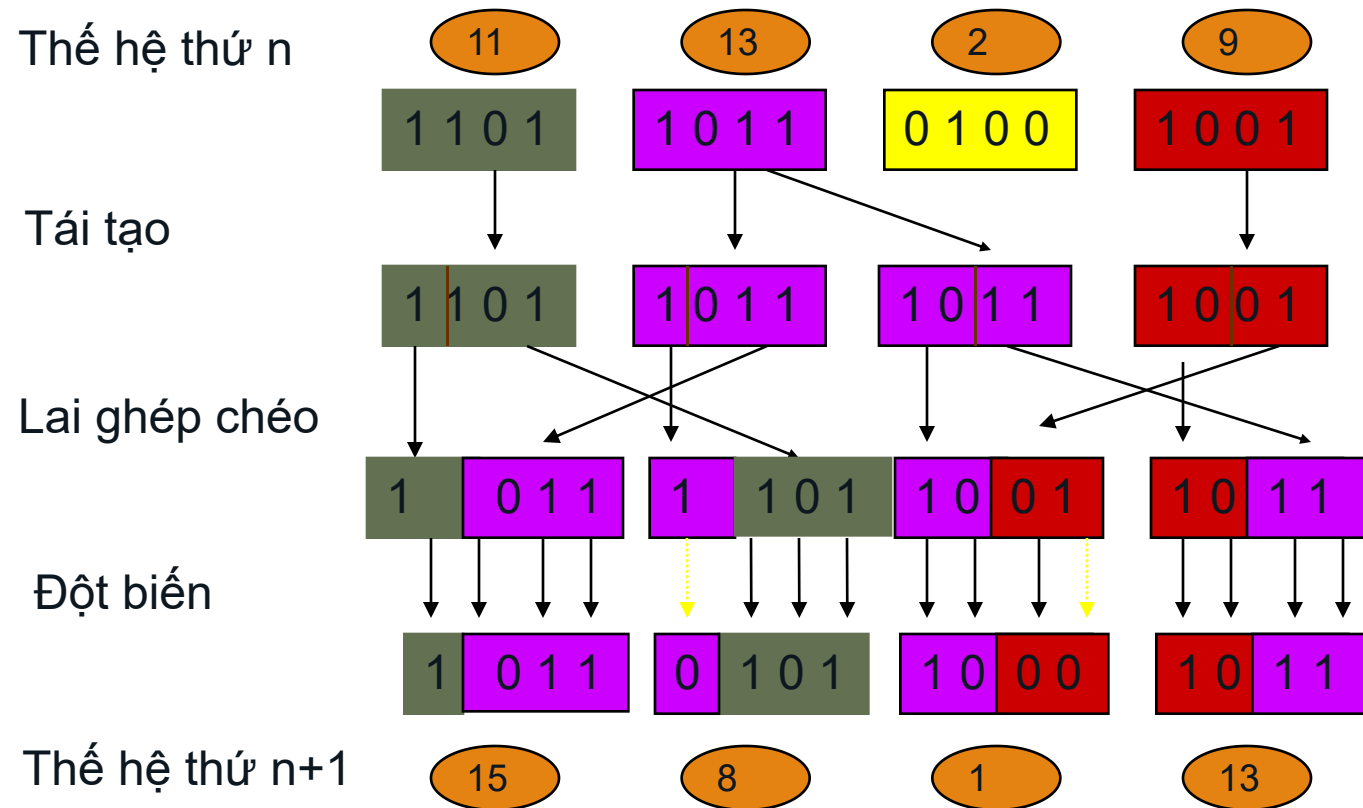
Lai ghép chéo và đột biến có thể làm hỏng đi các giải pháp tốt nhất trong quần thể

Elitism là phương pháp nhân bản các giải pháp tốt nhất trong quần thể

Elitism là giải pháp cho phép bảo tồn những giải pháp tốt nhất hiếm có trong quần thể

Elitism được thực hiện dựa trên phần trăm hoặc một con số cụ thể.

# Ví dụ về quá trình tiến hóa qua 2 thế hệ



# Một vài ứng dụng cơ bản

---

Xếp lịch

Vận tải hàng hóa

Thiết kế mạch

# Bài toán tìm đường

Bài toán:

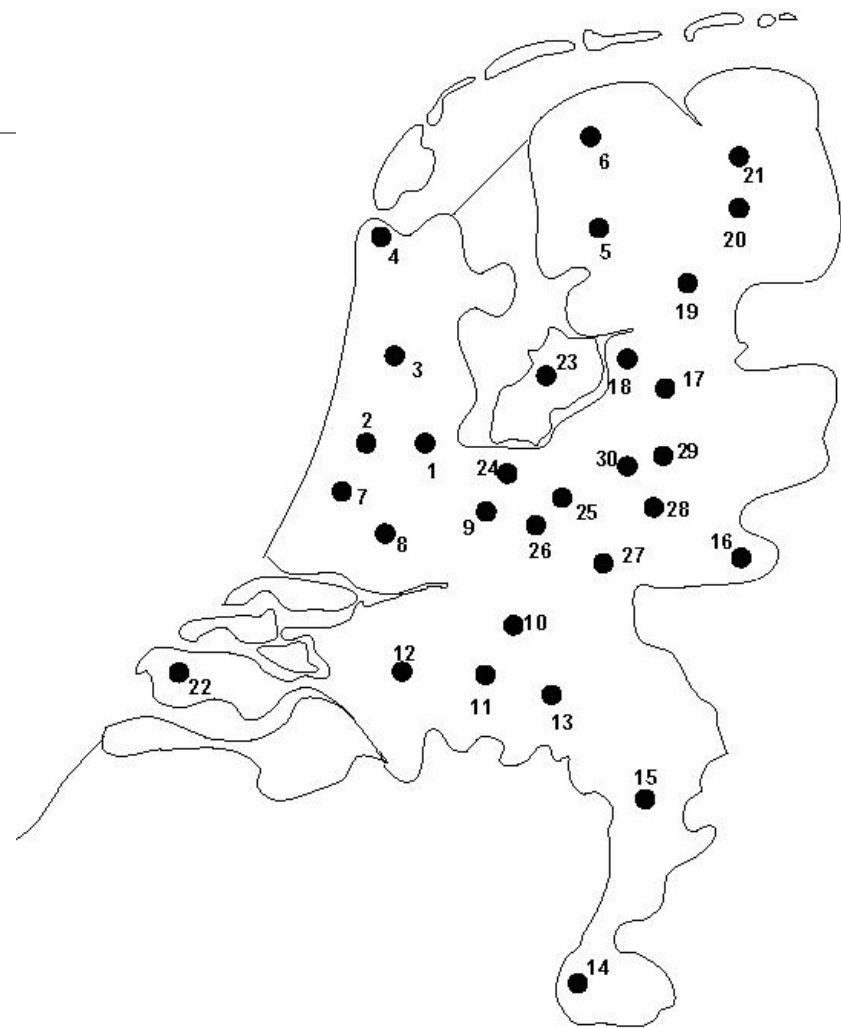
- Cho  $n$  thành phố
- Tìm tuyến đường đi qua tất cả các thành phố mà có độ dài ngắn nhất

Giải pháp:

- Gán nhãn mỗi thành phố là  $1, 2, \dots, n$
- Mỗi tuyến đường khác nhau là một sự hoán đổi vị trí các thành phố

Không gian tìm kiếm sẽ rất lớn.

- Với  $n=30$ , số tổ hợp phải thực hiện là  $30! \approx 10^{32}$





# Bài toán lập lịch làm việc

---

$J$  tập các công việc cần thực hiện.

$O$  là tập các thao tác

$M$  là tập các thiết bị cần sử dụng

$Able \subseteq O \times M$  chứa thông tin về các thiết bị nào có thể thực hiện các thao tác nào

$Pre \subseteq O \times O$  chứa thông tin về các thao tác nào cần thực hiện trước thao tác nào

$Dur : \subseteq O \times M \rightarrow \mathbb{R}$  chứa thông tin về thời gian mà máy  $m \in M$  thực hiện thao tác  $o \in O$

Mục tiêu là xây dựng lịch làm việc để:

Hoàn thành toàn bộ các công việc

Điều kiện: toàn bộ các điều kiện trong  $Able$  và  $Pre$  đều được thỏa mãn

Tối ưu: Tổng thời gian thực hiện là nhỏ nhất

---