

Thuật giải di truyền

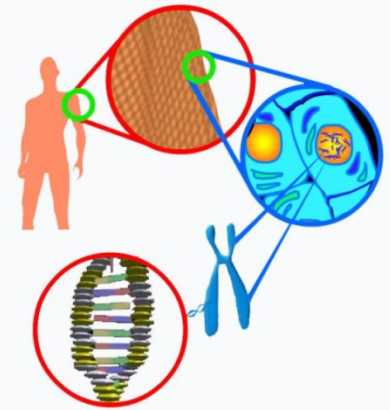
Giới thiệu

Các nguyên lý cơ bản của giải thuật di truyền được tác giả J.H.Holland công bố lần đầu tiên vào năm 1962.

Thuật giải di truyền (GA-Genetic Algorithms) là giải thuật tìm kiếm, chọn lựa các giải pháp tối ưu để giải quyết các bài toán thực tế khác nhau, dựa trên cơ chế chọn lọc của tự nhiên: từ tập lời giải ban đầu, thông qua nhiều bước tiến hoá, hình thành tập lời giải mới phù hợp hơn, và cuối cùng dẫn đến lời giải tối ưu toàn cục.

Nền tảng sinh học

- Tất cả các sinh vật sống bao gồm các tế bào
- Trong mỗi tế bào có cùng một bộ **nhễm sắc thể**.
- Nhiễm sắc thể gồm có **các gen**, các khối của DNA. Về cơ bản, nó có thể nói rằng mỗi gen mã hóa một **đặc điểm**, các tính trạng của gen được gọi là **allele**.
- Tập hợp các gen trong bộ gen được gọi là **kiểu gen**.
- Từ kiểu gen, với sự phát triển của sinh vật sẽ hình thành **kiểu hình**



Nền tảng sinh học

- **Quá trình tiến hóa**

- Cá thể mới sinh ra trong quá trình tiến hóa nhờ sự lai ghép ở thế hệ cha-mẹ, một cá thể mới có thể những tính trạng của cha-mẹ (di truyền), cũng có thể mang những tính trạng hoàn toàn mới (đột biến).

- **Độ thích nghi của sinh vật**

- Được đo bằng thành công của các sinh vật trong cuộc sống của nó

Vấn đề tối ưu hóa

Cho 1 hàm

$$g : \mathcal{V} \mapsto \mathbb{R}$$

Tìm

$$\arg \min_{v \in \mathcal{V}} g.$$

Mã hóa:

$$c : \mathcal{A}^\ell \mapsto \mathcal{V}$$

Do đó, vấn đề tối ưu hóa là tìm $\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x})$ với $\mathcal{X} \subseteq \mathcal{A}^\ell$

Khi

$$g(\mathbf{x}) = g(c(\mathbf{x}))$$

Mã hóa

Mã hóa nhiệm sắc thể là câu hỏi đầu tiên khi bắt đầu dùng GA giải quyết một vấn đề

Gồm 2 bước:

- Xây dựng cấu trúc gen cho mỗi lời giải của bài toán, từ đó mã hoá thành một NST (chuỗi các gen).
- Giải mã các NST để nhận được lời giải.

Mã hóa

Tùy thuộc vào nội dung của mỗi bài toán mà ta có cách mã hoá khác nhau.

- Mã hoá dạng chuỗi nhị phân
- Mã hóa dạng chuỗi q-phân
- Mã hoá thứ tự
- Mã hoá theo giá trị
- Mã hoá dạng cây:

Mã hóa

1. Mã hóa nhị phân

- Là phổ biến nhất vì đơn giản nhất. Mỗi nhiễm sắc thể là một chuỗi các **bit 0** hoặc **1**.

Mã hóa

1. Mã hóa nhị phân

Bài toán ví dụ:: có n mặt hàng, trong đó mặt hàng thứ i có trọng lượng (c_i) và trị giá (w_i), có một cái ba lô có sức chứa về trọng lượng tối đa là C . Vậy nên bỏ vào những món nào để đạt giá trị cao nhất trong khả năng chứa của ba lô.

Tức là tính tổng $\sum_{i \in I} w_i$

Để

$$\sum_{i \in I} c_i \leq C.$$

Mã hóa

Tính tổng $\sum_{i \in I} w_i$

Để $\sum_{i \in I} c_i \leq C.$

Nếu ta định nghĩa $x_i = \begin{cases} 1 & \text{if item } i \text{ is packed} \\ 0 & \text{otherwise} \end{cases}$

Khi đó bài toán được quy về việc tìm $\text{maximize } \sum_{i=1}^n w_i x_i$
 $\text{such that } \sum_{i=1}^n c_i x_i \leq C$

Mã hóa: nhiệm sắc thể là một chuỗi nhị phân có độ dài n

Mã hóa nhị phân

1. Mã hóa nhị phân

Bài toán ví dụ: Cần tối ưu hóa hàm $f(x) = x^3 - 60x^2 + 900x + 100$

Trên không gian tìm kiếm $\mathcal{V} = \{x : x \in \mathbb{Z}; x \in \{0, 31\}\}$,

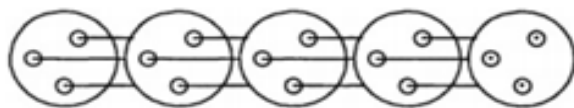
Mã hóa: chuỗi 5 chữ số nhị phân tương ứng với số nguyên $x \in \{0, 31\}$,

$$\left\{ \begin{array}{l} 00000 = 0 \\ 00001 = 1 \\ \dots\dots\dots \\ 11111 = 31 \end{array} \right.$$

Mã hóa

Mã hóa q-phân: được dùng trong trường hợp giá trị của alen không là lũy thừa của 2

Bài toán ví dụ: Một bộ gồm n rotor có sẵn, mỗi cái có q lỗ khoan, trong đó các rotor có thể lắp ráp thành một thiết bị đơn vị bằng cách chồng và đóng lại với nhau (thông qua các lỗ khoan như trong hình 2.1). Bởi vì các rotor không hoàn toàn phẳng, lắp ráp sẽ thấy lệch so với trục, tùy vào hiệu quả lắp ráp mà khi thiết bị khi hoạt động (quay) mà các rotor có thể sẽ lung lay. Mục tiêu là tìm tất cả các tổ hợp có thể định hướng cách lắp ráp sao cho độ lệch là ít nhất.

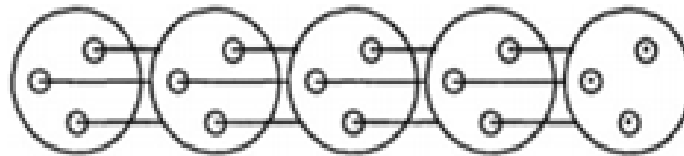


Mã hóa

Mã hóa: nhiễm sắc thể là 1 chuỗi chiều dài n , mỗi gen tương ứng với một rotor và chứa các alen $\{1, \dots, q\}$, đại diện các định hướng (so với một mốc cố định) của lỗ khoan.

VD: Chuỗi đại diện cho một giải pháp cho 3-rotor problem là (1 3 2)

Có nghĩa là: lỗ 1 của rotor đầu tiên được xếp với lỗ 3 của rotor thứ hai và nối tiếp với lỗ 2 của rotor thứ ba.



Chuỗi (132) (213) (321) là đại diện cho 3 giải pháp?

Mã hóa

3. Mã hoá thứ tự: được sử dụng trong bài toán có sắp xếp thứ tự. Ở đây mỗi NST là một chuỗi các số nguyên thể hiện thứ tự phân bố lời giải của bài toán.

Bài toán ví dụ: bài toán TSP: Cho một danh sách n thành phố và khoảng cách giữa chúng, nhiệm vụ là phải tìm đường đi ngắn nhất có thể để người bán hàng đi hết tất cả các thành phố và mỗi thành phố chỉ thăm đúng 1 lần.

Mã hóa: Nhiệm sắc thể mô tả thứ tự của thành phố, trong đó các nhân viên bán hàng sẽ đến thăm họ.

VD: Trường hợp $n = 9$, các thành phố được gán nhãn là 1,2,3,4,5,6,7,8,9 chuỗi NST: 947385612

Mã hóa

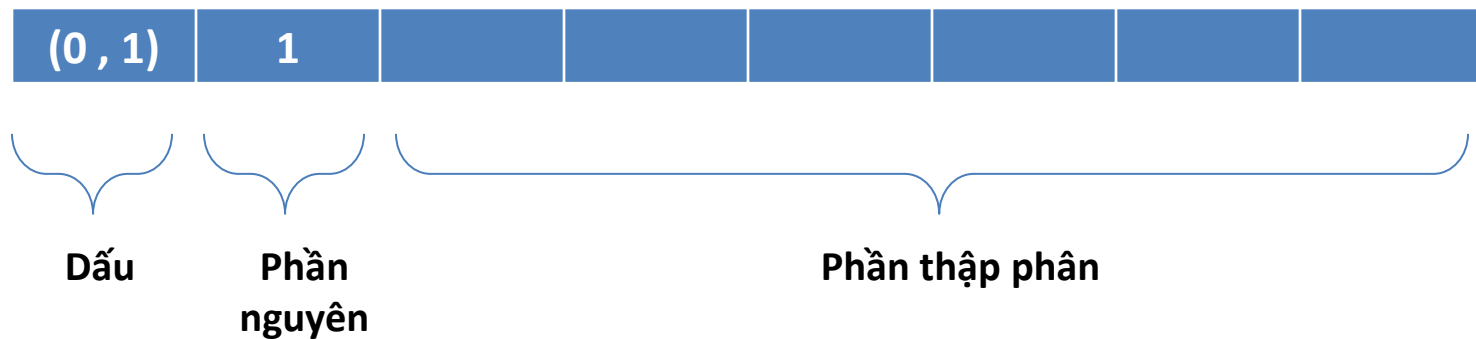
4. **Mã hoá theo giá trị:** được sử dụng trong các bài toán mà mỗi lời giải là tập các giá trị (ví dụ tập số thực). Các giá trị có thể được bất cứ điều gì kết nối với vấn đề, chẳng hạn như con số (**thực tế**) , ký tự hoặc bất kỳ đối tượng.

Mã hóa giá trị là một lựa chọn tốt cho một số vấn đề đặc biệt. Tuy nhiên, để mã hóa , ta cần phát triển phương pháp lai và đột biến cho từng vấn đề cụ thể

Mã hóa

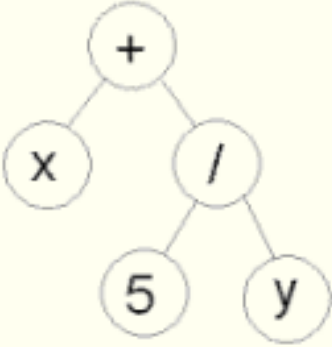
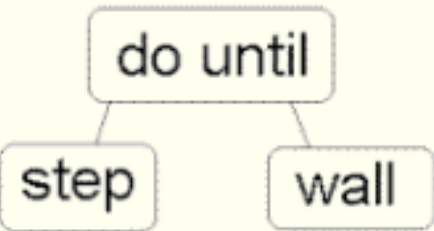
4. Mã hoá theo giá trị:

- **Bài toán ví dụ:** Tìm giá trị lớn nhất của hàm $f(x) = x \cdot \sin(10 \cdot \pi \cdot x) + 10$, với ràng buộc $-1 < x < 2$
- **Mã hóa:** chuỗi nhị phân thể chứa n số



Mã hóa

5. Mã hóa cây:: dùng trong trường hợp xử lý dữ liệu số, chuỗi, và các biểu thức đồng thời, chủ yếu trong các biểu thức toán học, trong phương pháp mã hoá này mỗi NST là một cây của một nhóm đối tượng nào đó.

| Nhiệm sắc thẻ A | Nhiệm sắc thẻ B |
|--|---|
|  |  |
| (+ X (/ 5 y)) | (Do_until bước tường) |

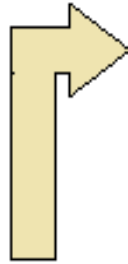
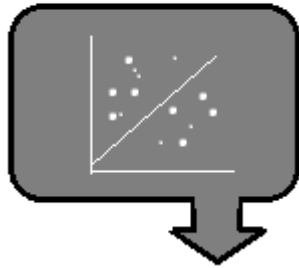
Ví dụ về nhiệm sắc thẻ với mã hóa cây

Thuật giải di truyền

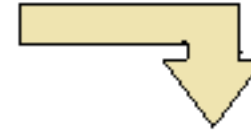
```
Choose an initial population of chromosomes;  
while termination condition not satisfied do  
  repeat  
    if crossover condition satisfied then  
      {select parent chromosomes;  
       choose crossover parameters;  
       perform crossover};  
    if mutation condition satisfied then  
      {select chromosome(s) for mutation;  
       choose mutation points;  
       perform mutation};  
    evaluate fitness of offspring  
  until sufficient offspring created;  
  select new population;  
endwhile
```

Sơ đồ thực hiện GA

Không gian tìm kiếm



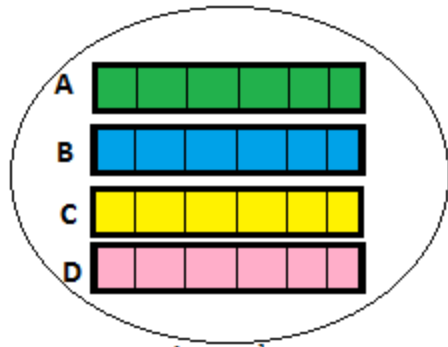
Chọn lọc



Lai ghép



Đột biến



Quần thể



Xác định độ thích nghi

Các yếu tố trong thuật giải di truyền

- Quần thể ban đầu
- Điều kiện dừng
- Hàm thích nghi (fitness)
- Các phép toán di truyền
 - Phép chọn lọc (selection)
 - Phép lai (crossover)
 - Phép đột biến (mutation)

Quần thể ban đầu

- Quần thể ban đầu được khởi tạo từ:
 - Các cá thể được tạo ngẫu nhiên.
 - Một tập các lời giải cung cấp bởi các chuyên gia
 - Một tập các lời giải cung cấp bởi các thuật toán tìm kiếm khác



Xác định kích thước quần thể

Kích thước quần thể

- Đối với cách biểu diễn nhị phân $P_2^* = (1 - (1/2)^{N-1})^\ell$.
- Sử dụng hàm mũ xấp xỉ $P_2^* \approx \exp(-\ell/2^{N-1})$
- Tính được $N \approx \lceil 1 + \log(-\ell / \ln P_2^*) / \log 2 \rceil$.
- Ví dụ: Đối với quần thể có NST có chiều dài $l=50$, để đạt xác suất 99.9% thì kích thước là $N=17$.

Các yếu tố trong thuật giải di truyền

- Quần thể ban đầu
- Điều kiện dừng
- Hàm thích nghi (fitness)
- Các phép toán di truyền
 - Phép chọn lọc (selection)
 - Phép lai (crossover)
 - Phép đột biến (mutation)
- Ví dụ minh họa

Điều kiện dừng (Termination)

- GA là 1 phương pháp tìm kiếm ngẫu nhiên mà về nguyên lý hoạt động thì nó sẽ không có kết thúc. Nên cần đặt điều kiện dừng cho nó.
- Thiết lập điều kiện dừng dựa trên:
 - Hàm thích nghi (fitness)
 - Thời gian chạy.
 - Sự đa dạng của quần thể (ngừng lại khi sự đa dạng giảm xuống dưới ngưỡng)

Các yếu tố trong thuật giải di truyền

- Quần thể ban đầu
- Điều kiện dừng
- Hàm thích nghi (fitness)
- Các phép toán di truyền
 - Phép chọn lọc (selection)
 - Phép lai (crossover)
 - Phép đột biến (mutation)
- Ví dụ minh họa

Hàm thích nghi

- Đầu vào: Chuỗi NST
- Đầu ra: Giá trị thích nghi của chuỗi NST, để đánh giá độ thích nghi của cá thể.
- Mục đích của hàm thích nghi là:
 - Chọn các cá thể thích nghi cho thế hệ sau.
 - Kiểm tra sự hội tụ của giải thuật.

Các yếu tố trong thuật giải di truyền

- Quần thể ban đầu
- Điều kiện dừng
- Hàm thích nghi (fitness)
- Các phép toán di truyền
 - Phép chọn lọc (selection)
 - Phép lai (crossover)
 - Phép đột biến (mutation)
- Ví dụ minh họa

Phép chọn lọc (selection)

- Phép chọn lọc bắt nguồn từ học thuyết của Darwin về “Sự sống sót của các cá thể thích nghi nhất ”
- Được thực hiện dựa trên giá trị của hàm thích nghi, các cá thể có độ thích nghi tốt sẽ được chọn
- **Mục đích của phép chọn lọc là tập trung sự tìm kiếm nghiệm trên miền thích nghi nhất.**

Phép chọn lọc (selection)

- Một số phép chọn lọc thường được sử dụng bao gồm:
 - **Roulette wheel Selection - Chọn lọc ngẫu nhiên theo vòng tròn Roulette**
 - Stochastic universal sampling Selection - Chọn lọc theo mẫu ngẫu nhiên.
 - Linear Ranking Selection- Chọn lọc theo thứ hạng tuyến tính
 - Tournament Selection- Chọn lọc theo cạnh tranh

Roulette wheel Selection (RWS)

- Đây được coi là phương pháp chọn lọc đơn giản nhất, ở đây mỗi chuỗi (cá thể) trong quần thể chiếm một vùng trong vòng tròn Roulette và có độ rộng tỷ lệ với giá trị hàm thích nghi. Mỗi lần quay vòng tròn Roulette chúng ta nhận được một chuỗi (cá thể) và coi như đó là cách lựa chọn chuỗi (cá thể) cho việc tái tạo.

Roulette wheel Selection (RWS)

- Các bước thực hiện:
 - Tính tổng các giá trị thích nghi của các cá thể trong một dân số và gán kết quả này vào biến Total fitness
 - Lấy một số ngẫu nhiên trong khoảng giữa 0 và Total fitness.
 - Trả về cá thể tập hợp đầu tiên dựa vào tổng giá trị thích nghi của nó và độ phù hợp của các thành viên trước đấy lớn hơn hoặc bằng n.

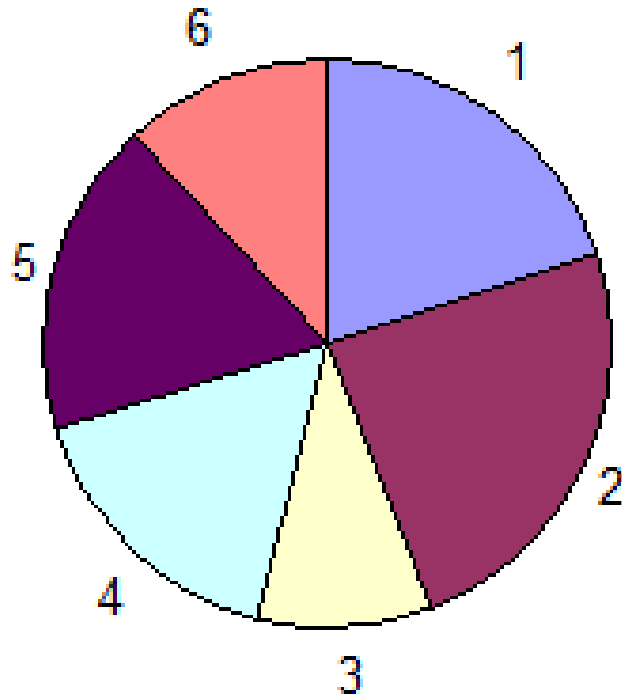
Roulette wheel Selection (RWS)

- Ví dụ: Giả sử ta có một dân số ban đầu với 6 cá thể (nhiễm sắc thể), tổng giá trị của hàm thích nghi là 50 như thể hiện trong bảng sau

| STT | Chuỗi | Giá trị thích nghi | Tỷ lệ % | Total |
|-----|-------|--------------------|---------|-------|
| 1 | 01110 | 10 | 20 | 10 |
| 2 | 11000 | 12 | 24 | 22 |
| 3 | 00100 | 5 | 10 | 27 |
| 4 | 10010 | 8 | 16 | 35 |
| 5 | 01100 | 9 | 18 | 44 |
| 6 | 00010 | 6 | 12 | 50 |

Roulette wheel Selection (RWS)

- Vòng tròn Roulette được biểu diễn như sau:



Roulette wheel Selection (RWS)

- Giả sử cần chọn ra 4 cá thể, khi đó ta cần tạo ra 4 số ngẫu nhiên trong khoảng từ (0, 50)

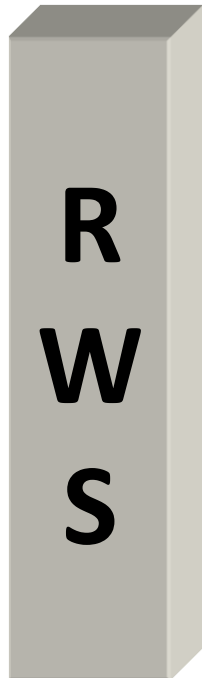
| | | | | |
|--------------------|----|----|----|----|
| Số ngẫu nhiên | 26 | 16 | 46 | 18 |
| Chuỗi NST (cá thể) | 3 | 2 | 6 | 2 |

Rõ ràng rằng, các chuỗi nào có giá trị thích nghi cao thì sẽ có nhiều khả năng được chọn



Hội tụ cục bộ !?!

Roulette wheel Selection (RWS)



Thực hiện M lần để chọn ra M cá thể

Xác suất lựa chọn tỷ lệ thuận với fitness

Duplication

Xảy ra khả năng hội tụ sớm

...

Phép chọn lọc (selection)

- Một số phép chọn lọc thường được sử dụng bao gồm:
 - Roulette wheel Selection - Chọn lọc ngẫu nhiên theo vòng tròn Roulette
 - **Stochastic universal sampling Selection - Chọn lọc theo mẫu ngẫu nhiên.**
 - Linear Ranking Selection- Chọn lọc theo thứ hạng tuyến tính
 - Tournament Selection- Chọn lọc theo cạnh tranh

Stochastic universal sampling Selection (SUS)

- Thực hiện giống như phương pháp vòng tròn Roulette, nhưng có một vài cải tiến. Các bước thực hiện như sau:
 - Giả sử quần thể ban đầu có N cá thể
 - Giả sử cần chọn ra M cá thể từ quần thể trên ($M < N$), khi đó khoảng cách giữa các con trỏ là $1/M$.
 - Vị trí của con trỏ đầu tiên được đưa tạo ra ngẫu nhiên và trong khoảng $[0, 1 / M]$.

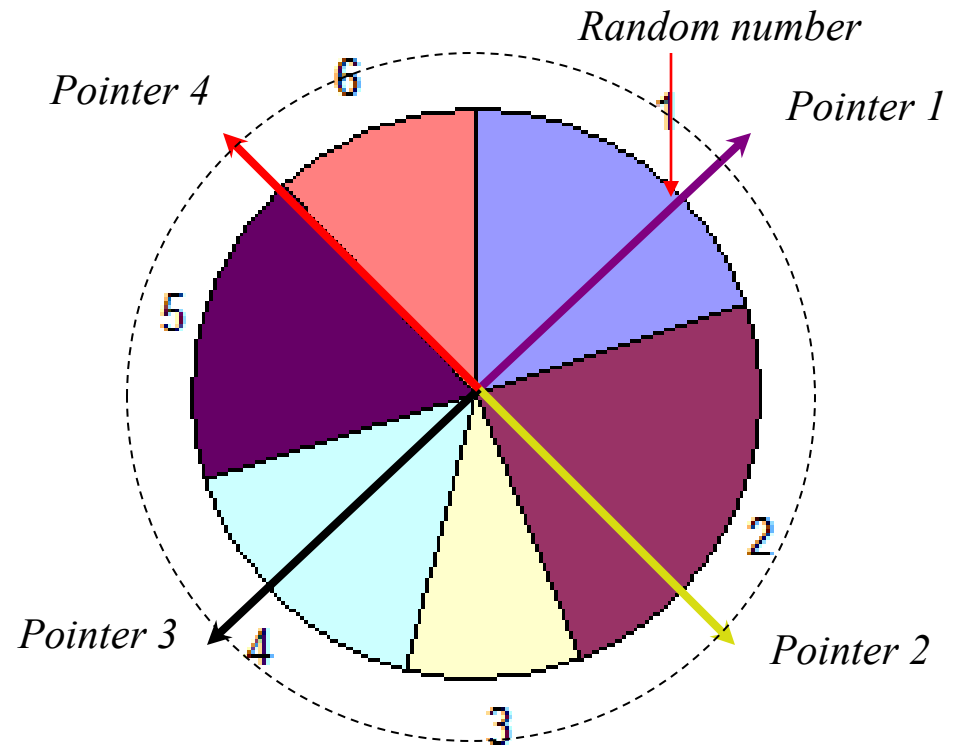
Stochastic universal sampling Selection (SUS)

- Cần chọn 4 cá thể nên khoảng cách giữa các con trỏ là:

$$1/4 = 0.25$$

- Sau khi áp dụng phương pháp này ta có các cá thể sau:

1, 2, 4 và 5



Stochastic universal sampling Selection (SUS)

S

U

S

Chỉ thực hiện 01 lần để chọn ra M cá thể

Giảm duplication

...

Phép chọn lọc (selection)

- Một số phép chọn lọc thường được sử dụng bao gồm:
 - Roulette wheel Selection - Chọn lọc ngẫu nhiên theo vòng tròn Roulette
 - Stochastic universal sampling Selection - Chọn lọc theo mẫu ngẫu nhiên.
 - **Linear Ranking Selection- Chọn lọc theo thứ hạng tuyến tính**
 - Tournament Selection- Chọn lọc theo cạnh tranh

Linear Ranking Selection

- Ý tưởng:
 - Sắp xếp các cá thể theo thứ tự giảm dần của độ thích nghi.
 - Từ số ngẫu nhiên r trong khoảng $[0,1]$, tìm thứ hạng \rightarrow tìm được cá thể cha mẹ. Lặp lại đến m lần (m số cá thể cha mẹ cần lựa chọn)

Linear Ranking Selection

- Ta có mối quan hệ giữa thứ hạng k và số ngẫu nhiên r ($r \in [0,1]$)

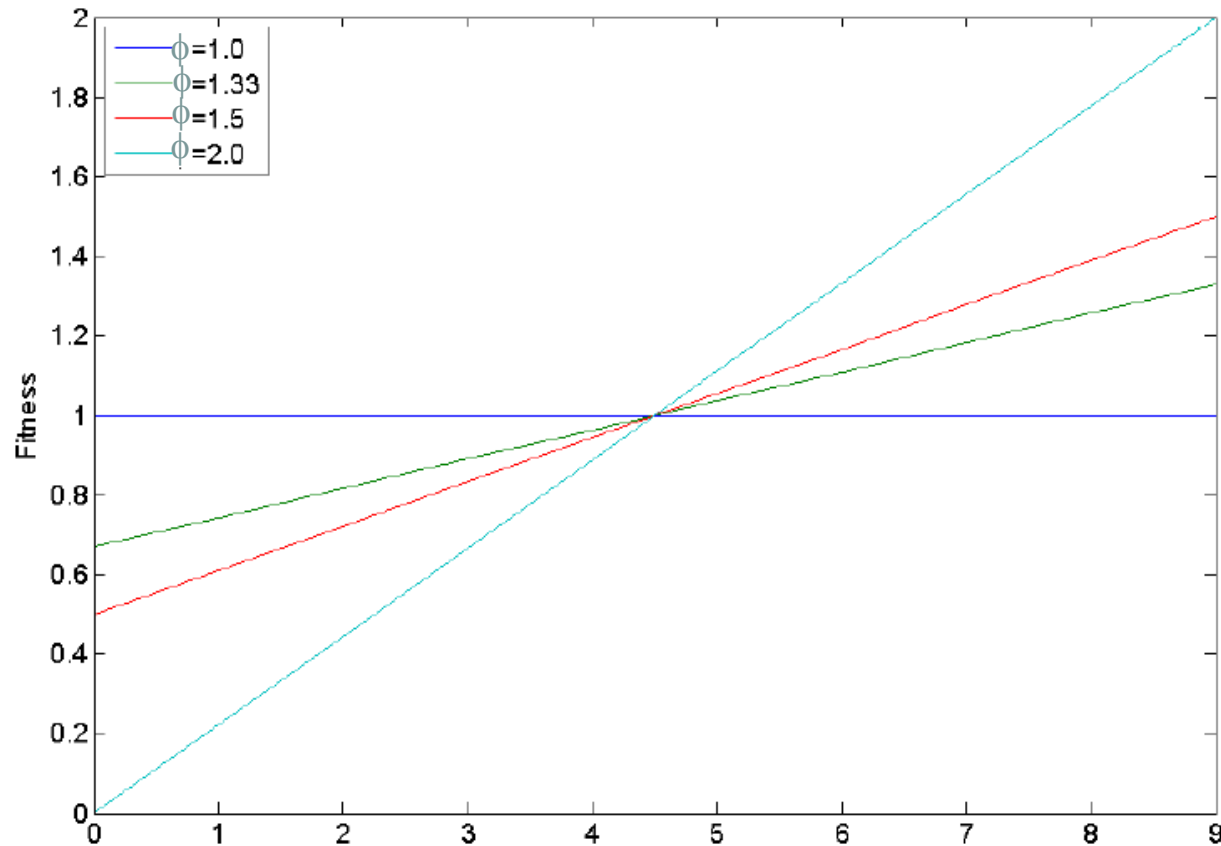
$$k = \frac{-(2\alpha + \beta) + \sqrt{(2\alpha + \beta)^2 + 8\beta r}}{2\beta}.$$

- Trong đó:

$$\beta = \frac{2(\phi - 1)}{N(N - 1)} \quad \text{và} \quad \alpha = \frac{2N - \phi(N + 1)}{N(N - 1)}$$

($\phi \in [1,2]$, giá trị của selection pressure)

Linear Ranking Selection



Fitness assignment for linear ranking with different selection pressures ϕ

Phép chọn lọc (selection)

- Một số phép chọn lọc thường được sử dụng bao gồm:
 - Roulette wheel Selection - Chọn lọc ngẫu nhiên theo vòng tròn Roulette
 - Stochastic universal sampling Selection - Chọn lọc theo mẫu ngẫu nhiên.
 - Linear Ranking Selection- Chọn lọc theo thứ hạng tuyến tính
 - **Tournament Selection- Chọn lọc theo cạnh tranh**

Tournament Selection

- Ý tưởng:
 - Lấy ngẫu nhiên k cá thể (k được gọi là tour size) và so sánh độ thích nghi giữa chúng, sau đó chọn ra cá thể có độ thích nghi cao nhất. Quá trình này được lặp đi lặp lại đúng m lần (m số cá thể muốn lựa chọn).

Tournament Selection

- Giải thuật:

Input: The population $\vec{J} = (J_1, \dots, J_N)$,
the fitness values of the population $\vec{\phi} = (\phi_1, \dots, \phi_N)$.
the tournament size $t \in \{1, 2, \dots, N\}$

Output: The population after selection $\vec{J}' = (J'_1, \dots, J'_N)$

tournament($t, \vec{J}, \vec{\phi}$):

 for $i \leftarrow 1$ to N do

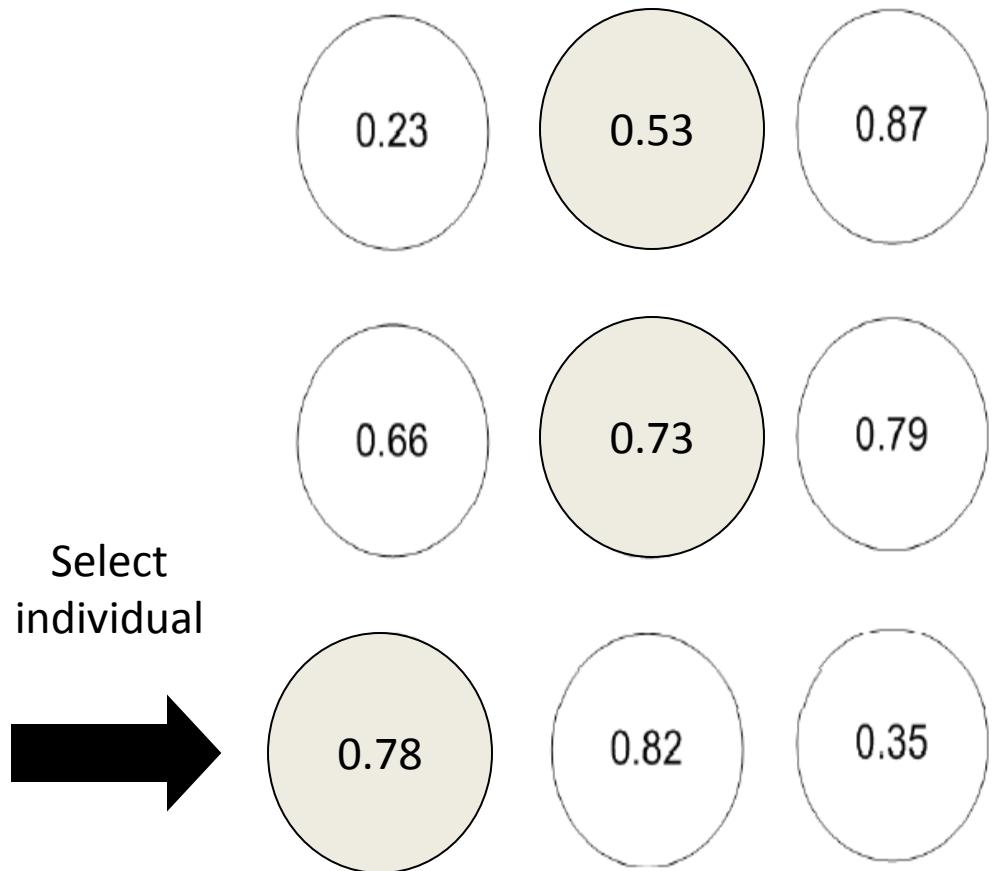
$J'_i \leftarrow$ best fit individual out of t randomly picked
 individuals from \vec{J}

 od

 return \vec{J}'

Tournament Selection

- Ví dụ: Cho quần thể có 9 cá thể có độ thích nghi như hình bên, tour size=3

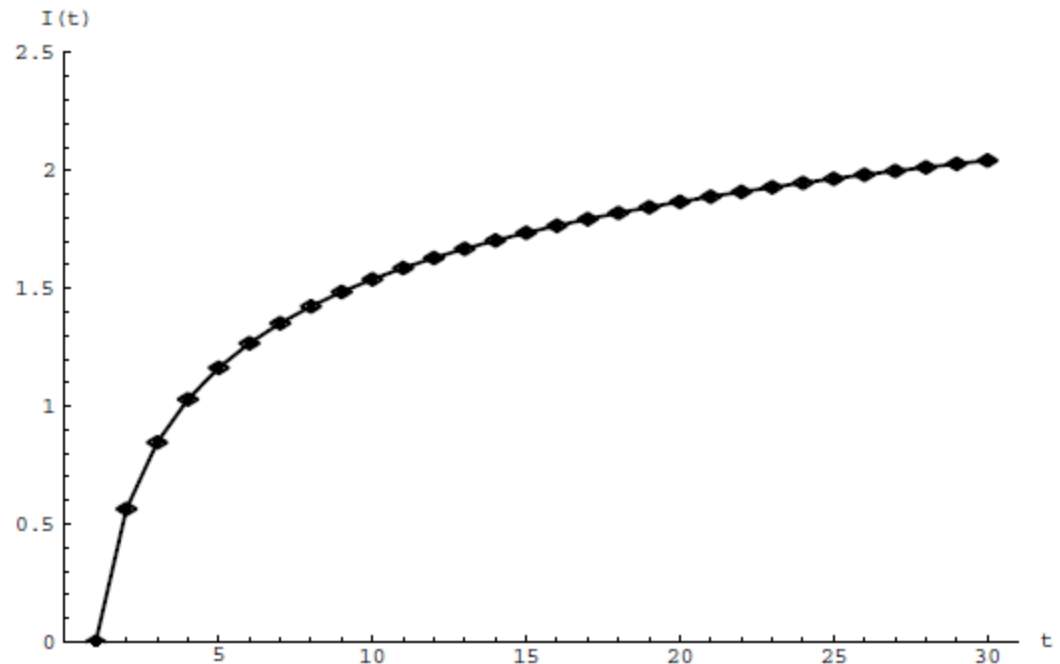


Tournament Selection

Chú ý:

- Tour size tỉ lệ thuận với số cá thể bị loại và selection pressure

- Selection intensity $I(t)$ và tour size t có mối quan hệ với nhau theo đồ thị sau:

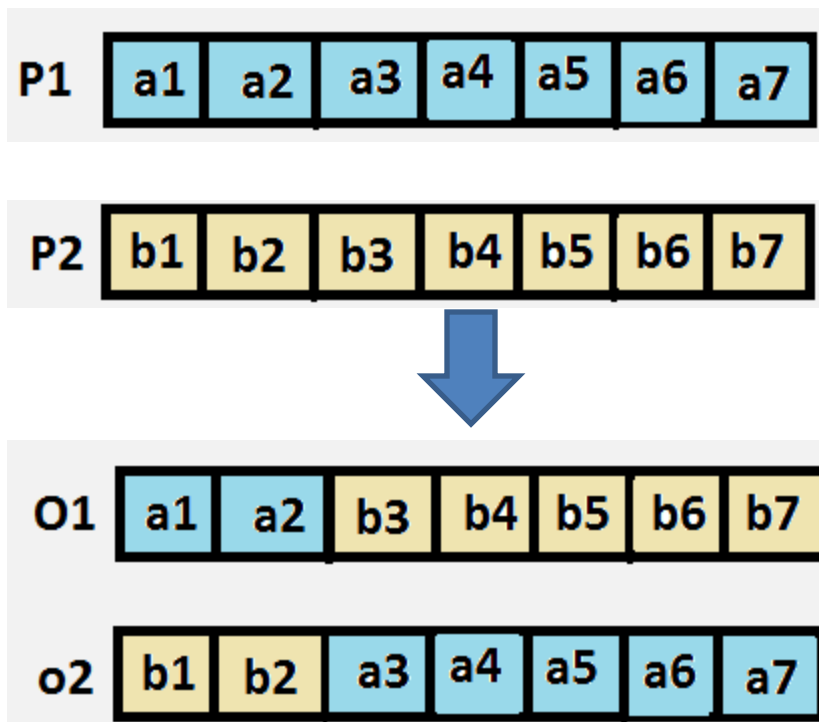


Phép lai (crossover)

- Quá trình kết hợp 2 cá thể cha mẹ đã được lựa chọn để tạo ra các cá thể con.
- Các phương pháp lai ghép:
 - Lai 1 điểm, lai m điểm
 - Lai phi tuyến (non- linear crossover)
 - Phép lai n điểm tổng quát (Generalized n-point crossover)

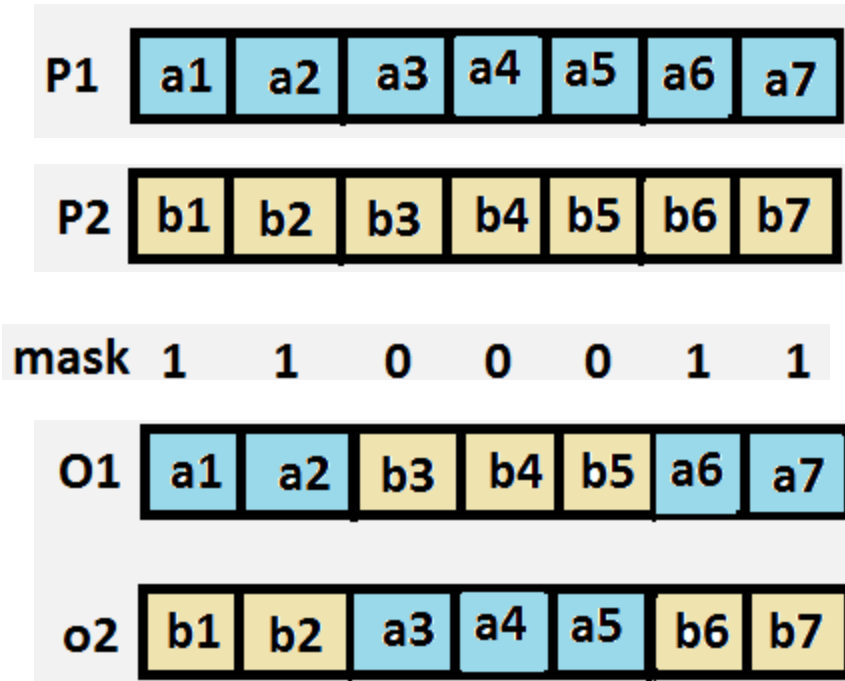
Lai 1 điểm

- Chọn ngẫu nhiên 1 vị trí trên NST và lai ghép các gen cha mẹ với nhau.



Lai m- điểm

- Lai ghép giữa cá thể cha mẹ với nhau thông qua việc sử dụng 1 mask.



Non- linear crossover

- Xét ví dụ: Trong bài toán TSP
- Sử dụng phép lai 1 điểm:

| | | | |
|----|---------|----|---------|
| P1 | 1634752 | O1 | 1612657 |
| | X | | |
| P2 | 4312657 | O2 | 4334752 |

Thấy rằng: Trong O1, thành phố 1 và 6 đi đến 2 lần trong khi đó không xuất hiện 3 và 4, O2 cũng tương tự thế, O1 và O2 là 2 nghiệm sai.

- Phép lai tuyến tính áp dụng trong bài toán này sẽ tạo những nghiệm sai, không mong muốn.

Non- linear crossover

- Partially matched crossover (PMX)
- Chọn ngẫu nhiên 2 vị trí từ 1 đến l-1, hoán đổi đoạn gen giữa 2 vị trí này với nhau.
- Ví dụ:

| | | | | | | | |
|----|---|---|---|---|---|---|---|
| P1 | 1 | 6 | 3 | 4 | 7 | 5 | 2 |
| | | | | | | | |
| P2 | 4 | 3 | 1 | 2 | 6 | 5 | 7 |

| | | | | | | | |
|----|---|---|---|---|---|---|---|
| O1 | 3 | 7 | 3 | 4 | 7 | 5 | 4 |
| | | | | | | | |
| O2 | 2 | 1 | 1 | 2 | 6 | 5 | 6 |

$3 \leftrightarrow 1; \quad 4 \leftrightarrow 2; \quad 7 \leftrightarrow 6$

Non- linear crossover

- Uniform crossover (UX)
- Sử dụng mask, tương tự như trong lai tuyến tính.
- Giữ nguyên giá trị tại vị trí mà mask có giá trị là 1, các vị trí còn lại được lấy theo thứ tự của P còn lại.

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

- | | | | | | | |
|----|---------------|---|---------------|---|----|---------------|
| P1 | 1 6 3 4 7 5 2 | ⇒ | 1 - 3 - - 5 - | ⇒ | O1 | 1 4 3 2 6 5 7 |
| P2 | 4 3 1 2 6 5 7 | ⇒ | 4 - 1 - - 5 - | ⇒ | O2 | 4 6 1 3 7 5 2 |

Generalized n-point crossover

Choose a random integer $n \in [1, \ell - 1]$;

Choose n crosspoints;

Generate a random permutation σ of $(1, \dots, n + 1)$ for segment order;

Designate one parent for copying;

$k \leftarrow 1$;

repeat

 copy all compatible alleles of segment σ_k from designated parent;

 swap parent designations

$k \leftarrow k + 1$;

until $k = n + 1$;

if child incomplete **then** insert legal allele(s) at required position, using random tie-breaking if necessary.

Generalized n-point crossover

- Ví dụ

| | | | | | | | |
|----|---|---|---|---|---|---|---|
| P1 | 1 | 6 | 3 | 4 | 7 | 5 | 2 |
| | | | | | | | |
| P2 | 4 | 3 | 1 | 2 | 6 | 5 | 7 |

- $n=2$, vị trí chọn là 2 và 5, ta có các đoạn được sắp thứ tự là $\sigma = (1, 3, 2)$.
- $k=1$, ta lấy đoạn gen 1 trong P1:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 6 | — | — | — | — |
|---|---|---|---|---|---|

 ;
- $k=2$, ta lấy đoạn gen 3 trong P2:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 6 | — | — | 5 | 7 |
|---|---|---|---|---|---|

 ;
- $k=3$, ta lấy đoạn gen 2 trong P1:

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 6 | 3 | 4 | — | 5 | 7 |
|---|---|---|---|---|---|---|
- Cuối cùng bổ sung allele 2 còn thiếu vào

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 6 | 3 | 4 | 2 | 5 | 7 | . |
|---|---|---|---|---|---|---|---|

Nhận xét

- Lai 1 điểm, lai m- điểm là phép lai tuyến tính, chỉ sử dụng cho các bài toán biểu diễn NST bằng chuỗi nhị phân.
- Lai 1 điểm hạn chế việc trao đổi thông tin giữa 2 cá thể cha mẹ hơn phép lai m-điểm.
- Lai phi tuyến sử dụng cho các bài toán với không gian mẫu được biểu diễn bằng các hoán vị (ví dụ bài toán TSP).

Các yếu tố trong thuật giải di truyền

- Quần thể ban đầu
- Điều kiện dừng
- Hàm thích nghi (fitness)
- Các phép toán di truyền
 - Phép chọn lọc (selection)
 - Phép lai (crossover)
 - Phép đột biến (mutation)
- Ví dụ minh họa

Phép đột biến (mutation)

- Đột biến là thay đổi các bit trên chuỗi nhiễm sắc thể một cách ngẫu nhiên để tạo tính đa dạng.
- Toán tử đột biến được xây dựng để tránh việc nhận được giá trị tối ưu cục bộ.
- Mỗi gen trên NST có tỉ lệ đột biến như nhau

$$\mu = 1/l.$$

Với l là chiều dài NST.

Phép đột biến (mutation)

- Ví dụ: Ta có chuỗi

1 1 0 1 1 0 1

Đột biến trên gen thứ 2 và 6, sau khi đột biến được:

1 0 0 1 1 1 1

Phép đột biến (mutation)

- Trường hợp, các cá thể được biểu diễn bằng cách khác (không phải nhị phân), khi đó mỗi gen sẽ có nhiều giá trị, do đó khi thực hiện đột biến ta sẽ phải chọn giá trị nào thay vào?
- *Biểu diễn bằng hoán vị*: Chọn hai vị trí bất kỳ rồi hoán đổi giá trị của chúng cho nhau: (1 2 **3** 4 5 6 7 **8** 9) => (1 2 **8** 4 5 6 7 **3** 9)
- *Biểu diễn bằng giá trị*: Chọn một vài giá trị rồi thêm hoặc bớt (cộng hay trừ) một giá trị nhỏ:
(3.49; 7.63; **3.55**; **7.24**; 4.83) => (3.49; 7.63; **3.61**; **7.18**; 4.83)

Các yếu tố trong thuật giải di truyền

- Quần thể ban đầu
- Điều kiện dừng
- Hàm thích nghi (fitness)
- Các phép toán di truyền
 - Phép chọn lọc (selection)
 - Phép lai (crossover)
 - Phép đột biến (mutation)
- Ví dụ minh họa

Ví dụ minh họa

- Tìm giá trị cực tiểu của hàm

$$g(x) = (x_1 - 6)^2 + (x_2 - 4)^2$$

Với $0 \leq x_1, x_2 \leq 5$

Ví dụ minh họa

- Với kích thước quần thể là $N=30$, độ chính xác là 0.1
- Biểu diễn nhị phân
 - Sử dụng chuỗi NST có $l=12$ để biểu diễn cặp (x,y) , với x được biểu diễn bằng 6 bit đầu, y là 6 bit cuối.
 - Ví dụ:
 - 000000 000000 $\rightarrow (0,0)$
 - 000100 001000 $\rightarrow (0.4, 0.8)$

Ví dụ minh họa

- Khởi tạo quần thể ban đầu: tạo ngẫu nhiên $n=30$ chuỗi nhị phân có chiều dài 12
- Hàm thích nghi (fitness): $f(x) = C - g(x)$
với C là tham số đầu vào đủ lớn để $f(x) > 0$
- Sử dụng phép chọn lọc Roulette dựa trên hàm thích nghi để chọn các cá thể vào thế hệ sau, phép lai 1 điểm và phép đột biến.
- Sau 40 lần lặp kết quả (4.9, 3.9)

ĐẶT VẤN ĐỀ

- Thuật giải di truyền truyền thống không phải là một thuật toán tối ưu nhất.
- Trong thuật giải di truyền truyền thống toán tử chọn lọc thông thường là dùng chọn lọc tự nhiên. Chọn lọc này có những vấn đề tiềm ẩn như sau:
 - Cá thể tốt của quần thể có thể không được chọn lọc để sinh ra trong lần lai tạo kế tiếp.
 - Hai cá thể tương tự nhau thì khi lai tạo có thể không cho ra cá thể tốt.
- Nhưng nếu áp dụng chọn lọc nghiêm khắc thì quần thể không có tính đa dạng.

Breeder GA

- BGA lần đầu tiên được giới thiệu bởi Muhleiihein và Schlierkamp-Voosen vào năm 1993.
- BGA biểu diễn những giải pháp dưới dạng vector các giá trị thực, cho phép biểu diễn gần với thực tế hơn những GA thông thường (sử dụng vector các giá trị nguyên hoặc nhị phân)
- BGA là một thuật giải di truyền dựa trên việc ***chọn lọc nhân tạo*** tương tự như những gì con người thực hiện việc nhân giống thực vật và động vật.

- BGA sử dụng mô hình **chọn lọc xén** (truncation rate): $T\%$ (được gọi là tỉ lệ xén) những cá thể tốt nhất được chọn lựa và được “*gây giống*” một cách *ngẫu nhiên* cho đến khi số lượng con cháu đạt đến kích thước của quần thể. Thế hệ con cháu sẽ thay thế thế hệ bố mẹ.

Thuật giải

- Input

- n : String length

- P_0 : Initial frequency of desirable allele

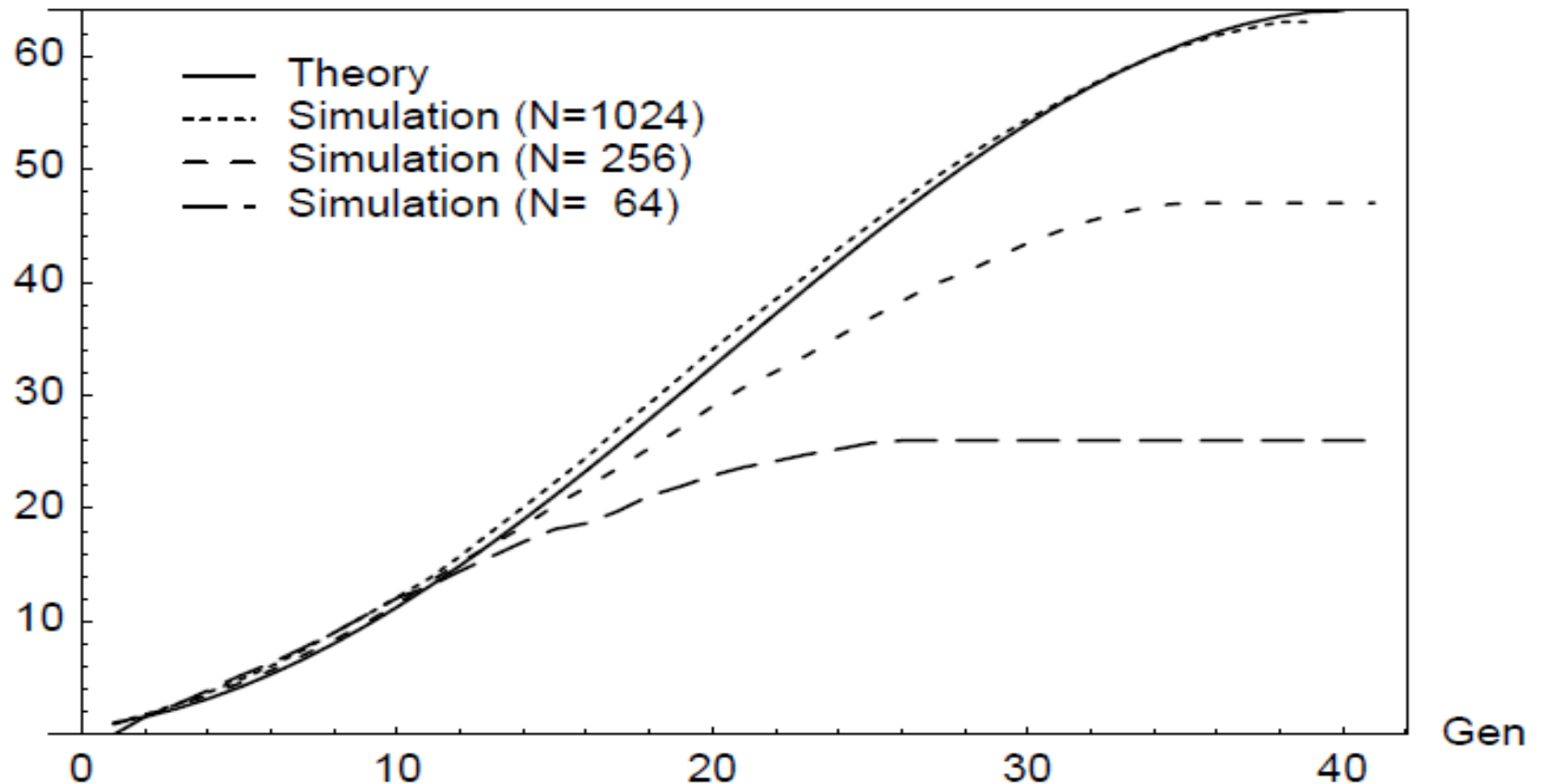
- N : Population size

- M : Mutation rate

- I : Selection intensity

Lai tạo làm việc hiệu quả trên những quần thể đủ lớn

MeanFit



Thuật giải

Procedure *Breeder Genetic Algorithm*

begin

 randomly initialise a population of N individuals;

while termination criterion not fulfilled **do**

evaluate goodness of each individual;

save the best individual in the new population;

select the best $T\%$ individuals;

for $i = 1$ **to** $N-1$ **do**

randomly select two elements among the $T\%$;

recombine them so as to obtain one offspring;

perform mutation on the offspring

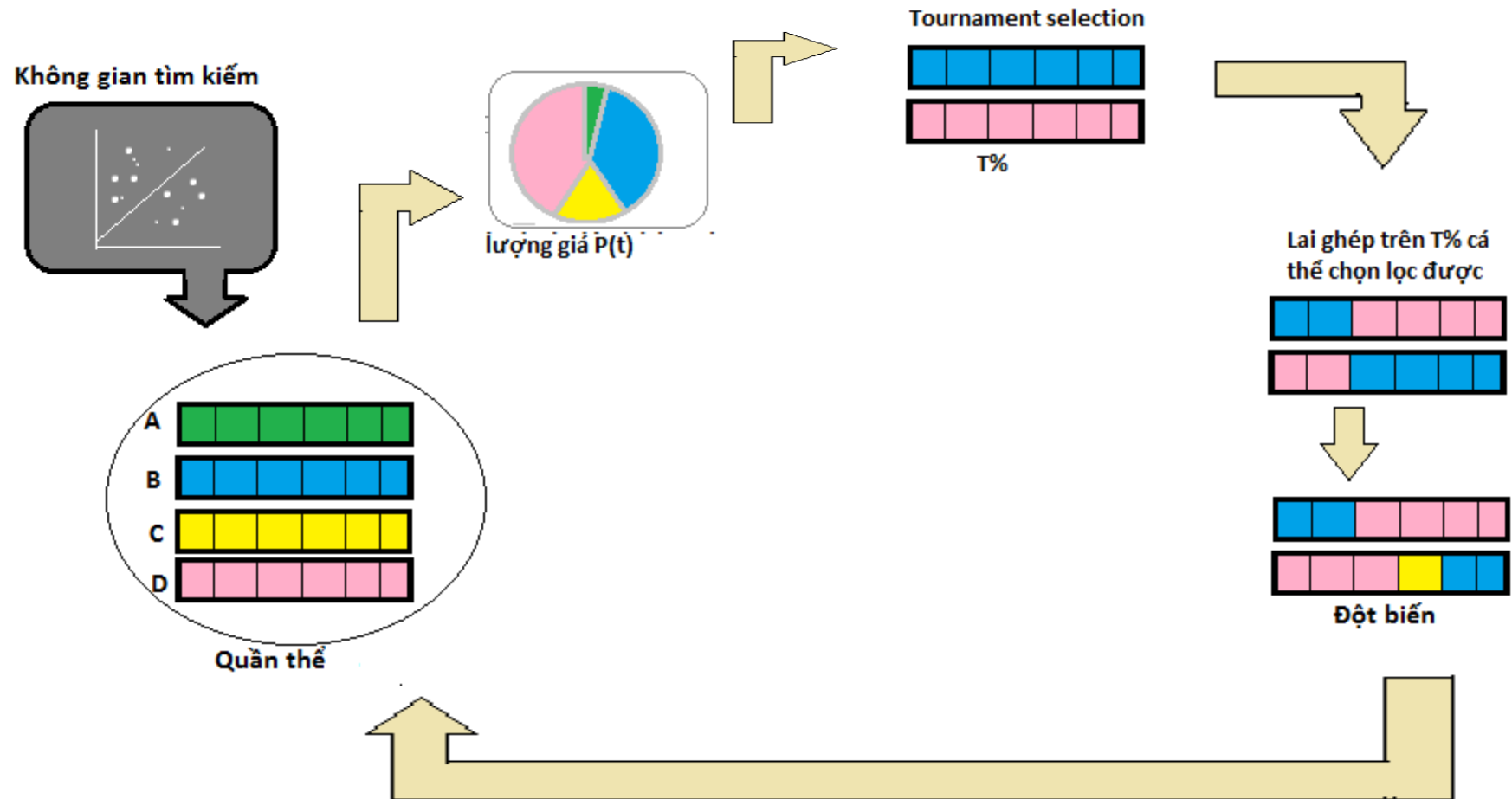
Loop

update variables for termination;

Loop

End

Sơ đồ thực hiện BGA



Cataclysmic Hux Combination: CHC

- Lai tạo được nhấn mạnh nhiều
- Sử dụng chọn lọc Elitism
- Hầu như đột biến không sử dụng.
- Duy trì tính đa dạng quần thể: Sử dụng phiên bản gây rối UX (hay HUX): gọi là sự ngăn cản dứt khoát gọi là incest, nghĩa là những cá thể tương tự nhau không được phép kết hợp với nhau.

Thuật giải

- Input
 - n : String length
 - N : Population size
 - r : Divergence rate
 - d : Difference Threshold

Thuật giải

- 1 $t = 0$
2 **initialize** $P(t)$
3 **evaluate** structures in $P(t)$
4 while not end do
5 $t = t + 1$
6 **select**: $C(t) = P(t-1)$
7 **recombine**: $C'(t) = \text{'incest prevention'} + \text{HUX}(C'(t))$
8 **evaluate** $C''(t)$ from structures in $C'(t)$
9 **replace** $P(t)$ from $C''(t)$ and $P(t-1)$
10 if convergence($P(t)$)
11 **diverge** $P(t)$
12 end while

GAVaPS

- Kích thước quần thể là một trong những lựa chọn quan trọng mà tất cả người sử dụng di truyền phải đương đầu và có thể tùy thuộc ứng dụng. Có hai vấn đề về kích thước
 - Kích thước quá nhỏ -> hội tụ nhanh
 - Kích thước quá lớn thì có tính đa dạng quần thể nhưng hao phí tài nguyên máy tính.

GAVaPS

- Trong thuật giải di truyền hai vấn đề quan trọng là:
 - Duy trì tính đa dạng quần thể (maintenance of diversity)
 - Áp lực chọn lọc (Selection intensity)
- Hai tính chất này đều bị ảnh hưởng bởi kích thước quần thể.

GAVaPS

- Trong giải thuật **GAVaPS** không dùng một cơ chế chọn lọc nào mà dùng khái niệm tuổi của một nhiễm sắc thể, *tương đương với một số thế hệ mà NST còn sống*.
- Vì vậy, khái niệm “tuổi” thay thế khái niệm chọn lọc. Do phụ thuộc vào độ thích nghi của cá thể, nên ảnh hưởng đến kích thước quần thể ở mỗi giai đoạn tiến trình.
- Thời gian sống: là thời gian tồn tại của NST đó.

Thuật toán

- Input

N: Initial_size

p: Reproduction ratio

M: Mutation ratio

C: Crossover ratio

n: Length of chromosome

MaxLT: Max LifeTime

MinLT: Min LifeTime

Procedure GAVaPS

begin

t = 0

initialize P(t)

evaluate P(t)

while (not termination-condition) do

begin

t = t+1

increase the *age* to each individual by 1

recombine P(t) $AuxPopSize(t) = \lfloor PopSize(t) * p \rfloor$

evaluate P(t)

remove from P(t) all individuals with *age* greater than their *lifetime*

$$PopSize(t+1) = PopSize(t) + AuxPopSize(t) - Dead(t)$$

end

end

Giá trị thời gian sống

- Phân phối tỉ lệ

$$\min(\text{MinLT} + \eta \frac{\text{fitness}[i]}{\text{AvgFit}}, \text{MaxLT})$$

$$\text{where } \eta = \frac{1}{2} (\text{MaxLT} - \text{MinLT})$$

Trong đó: fitness là giá trị thích nghi

MinLT: giá trị thời gian sống cực tiểu

MaxLT: giá trị thời gian sống cực đại

avgFit: Giá trị thích nghi trung bình tại thời điểm hiện tại

Giá trị thời gian sống

- Phân phối tuyến tính

$$MinLT + 2\eta \frac{fitness[i] - AbsFitMin}{AbsFitMax - AbsFitMin}$$

Trong đó: fitness là giá trị thích nghi

MinLT: giá trị thời gian sống cực đại

absFitMax: giá trị thích nghi lớn nhất từ trước đến giờ.

absFitMin: giá trị thích nghi nhỏ nhất từ trước đến giờ.

Giá trị thời gian sống

- Phân phối song tuyến tính

$$MinLT + \eta \frac{fitness[i] - MinFit}{AvgFit - MinFit} \quad \text{if } AvgFit \geq fitness[i]$$

$$\frac{1}{2}(MinLT + MaxLT) + \eta \frac{fitness[i] - AvgFig}{MaxFit - AvgFit} \quad \text{if } AvgFit < fitness[i]$$