



**UIT**

TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# INTRODUCTION TO EVOLUTIONARY ALGORITHMS

-

## GIỚI THIỆU CÁC THUẬT TOÁN TIẾN HÓA

# Tối ưu hoá (Optimization)

Tối thiểu hoá (minimization)

Cho hàm số  $f: S \rightarrow \mathbb{R}$ .

Tìm  $x^* \in S$  sao cho  $\forall x \in S, f(x^*) \leq f(x)$ .

# Tối ưu hoá (Optimization)

## Tối thiểu hoá (minimization)

Cho hàm số  $f: S \rightarrow \mathbb{R}$ .

Tìm  $x^* \in S$  sao cho  $\forall x \in S, f(x^*) \leq f(x)$ .

## Tối đa hoá (maximization)

Cho hàm số  $f: S \rightarrow \mathbb{R}$ .

Tìm  $x^* \in S$  sao cho  $\forall x \in S, f(x^*) \geq f(x)$ .

# Tối ưu hoá (Optimization)

Ví dụ 1:

Cho hàm số  $f(x) = (x - 0.5)^2$ .

Tìm giá trị nhỏ nhất của  $f$  trên  $\mathbb{R}$ .

Cách giải ?

# Tối ưu hoá (Optimization)

Ví dụ 1:

Cho hàm số  $f(x) = (x - 0.5)^2$ .

Tìm giá trị nhỏ nhất của  $f$  trên  $\mathbb{R}$ .

Cách giải: Tính đạo hàm (derivative) của  $f$ .

# Tối ưu hoá (Optimization)

Ví dụ 2:

Cho đồ thị  $G(V,E)$  biểu diễn độ dài đường đi giữa các thành phố. Tìm đường đi ngắn nhất giữa 2 thành phố.

Cách giải ?

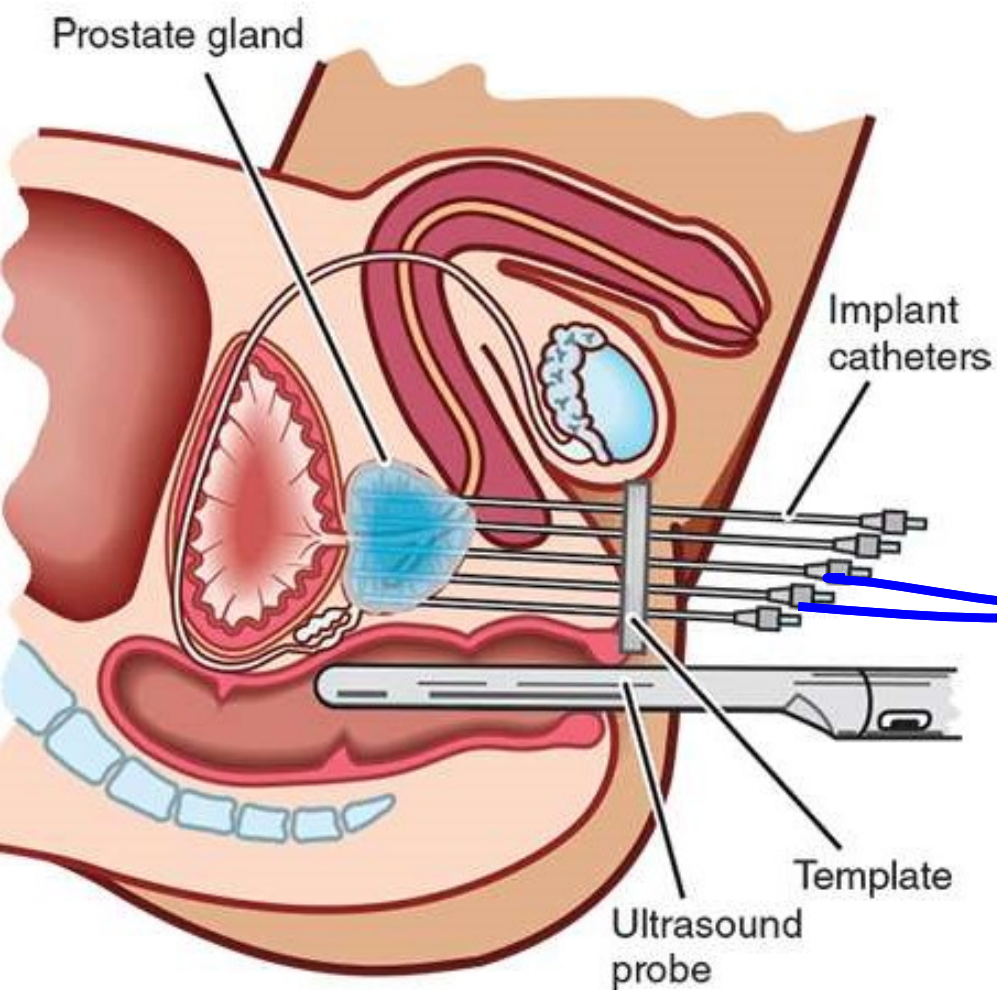
# Tối ưu hoá (Optimization)

Ví dụ 2:

Cho đồ thị  $G(V,E)$  biểu diễn độ dài đường đi giữa các thành phố. Tìm đường đi ngắn nhất giữa 2 thành phố.

Cách giải: Thuật toán Dijkstra.

# Tối ưu hoá (Optimization)



Ví dụ 3:

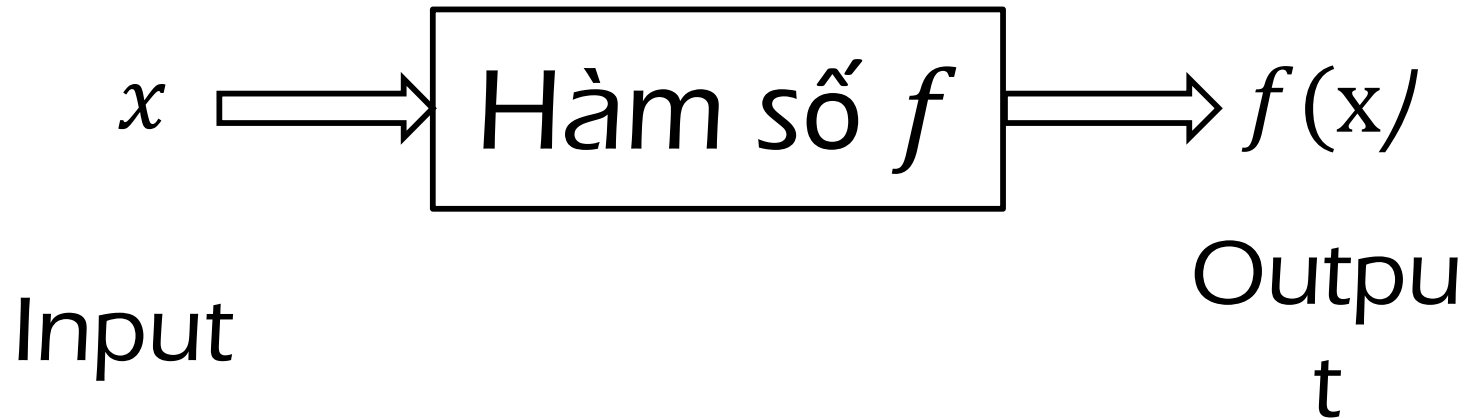
Cho ảnh y khoa thể hiện cấu trúc tuyến tiền liệt của một bệnh nhân. Tìm giải pháp xạ trị ung thư tối ưu.

Cách giải?





# Tối ưu hoá (Optimization)



→ Phân tích  $f$  để thiết kế và áp dụng thuật giải thích hợp.

# Tối ưu hoá (Optimization)



Làm sao để giải bài toán  
black-box optimization?

# Tối ưu hoá (Optimization)

Thách thức khi giải bài toán tối ưu hoá trong thực tế

- Thông tin vấn đề không có sẵn, không đầy đủ.
- Vấn đề có độ phức tạp cao, khó phân tích.
- Chưa có (hoặc không tồn tại) thuật giải hiệu quả.

# Tiến hoá (Evolution)

- Quá trình tiến hoá của các loài trong tự nhiên có nét tương đồng với một quá trình tối ưu hoá.
- Nhìn chung, sau mỗi thế hệ (generation), các cá thể (individuals) được sinh ra có độ thích nghi (fitness) tốt hơn với môi trường sống.
- Tại sao ta có thể lấy ý tưởng từ quá trình tiến hoá để thiết kế thuật giải cho các bài toán tối ưu hoá?

# Các thuật toán tiến hoá (Evolutionary Algorithms)

Cho hàm số  $f: S \rightarrow \mathbb{R}$ .

Tìm  $x^* \in S$  sao cho  $\forall x \in S, f(x^*) \geq f(x)$ .

Tối ưu hoá	Thuật toán tiến hoá
$f$ là hàm mục tiêu (objective function) / hàm tối ưu (optimization function).	$f^{EA}$ là hàm thích nghi (fitness function) / hàm lượng giá (evaluation function).
Mỗi $x$ là một lời giải / giải pháp (solution). $x^*$ là (một) giải pháp tối ưu.	Mỗi $x^{EA}$ là một cá thể, hay còn gọi là kiểu gen (genotype). $x$ được xem như là kiểu hình (phenotype).
$S$ là tập tất cả những giải pháp hợp lệ (set of feasible solutions) của bài toán.	$S^{EA}$ là không gian tìm kiếm hợp lệ (feasible search space) của thuật toán.



# Các thuật toán tiến hoá (Evolutionary Algorithms)

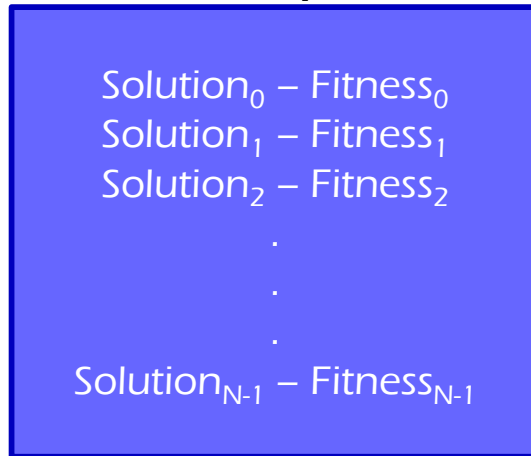
Nhìn chung các thuật toán tiến hoá (Evolutionary Algorithms – EAs) có 3 thành phần chính:

1. **Quần thể** (population) gồm nhiều cá thể (individuals).
2. **Phép biến đổi** (variation): để sinh ra cá thể mới (offspring).
3. **Phép chọn lọc** (selection): để lựa chọn những cá thể có độ thích nghi (fitness) tốt hơn cho thế hệ tiếp theo.

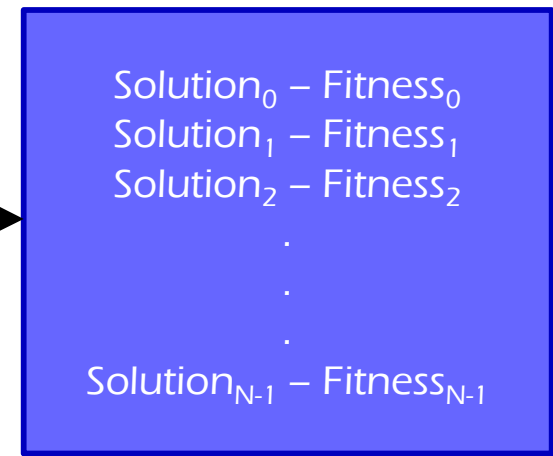


# Các thuật toán tiến hoá (Evolutionary Algorithms)

Thế hệ t

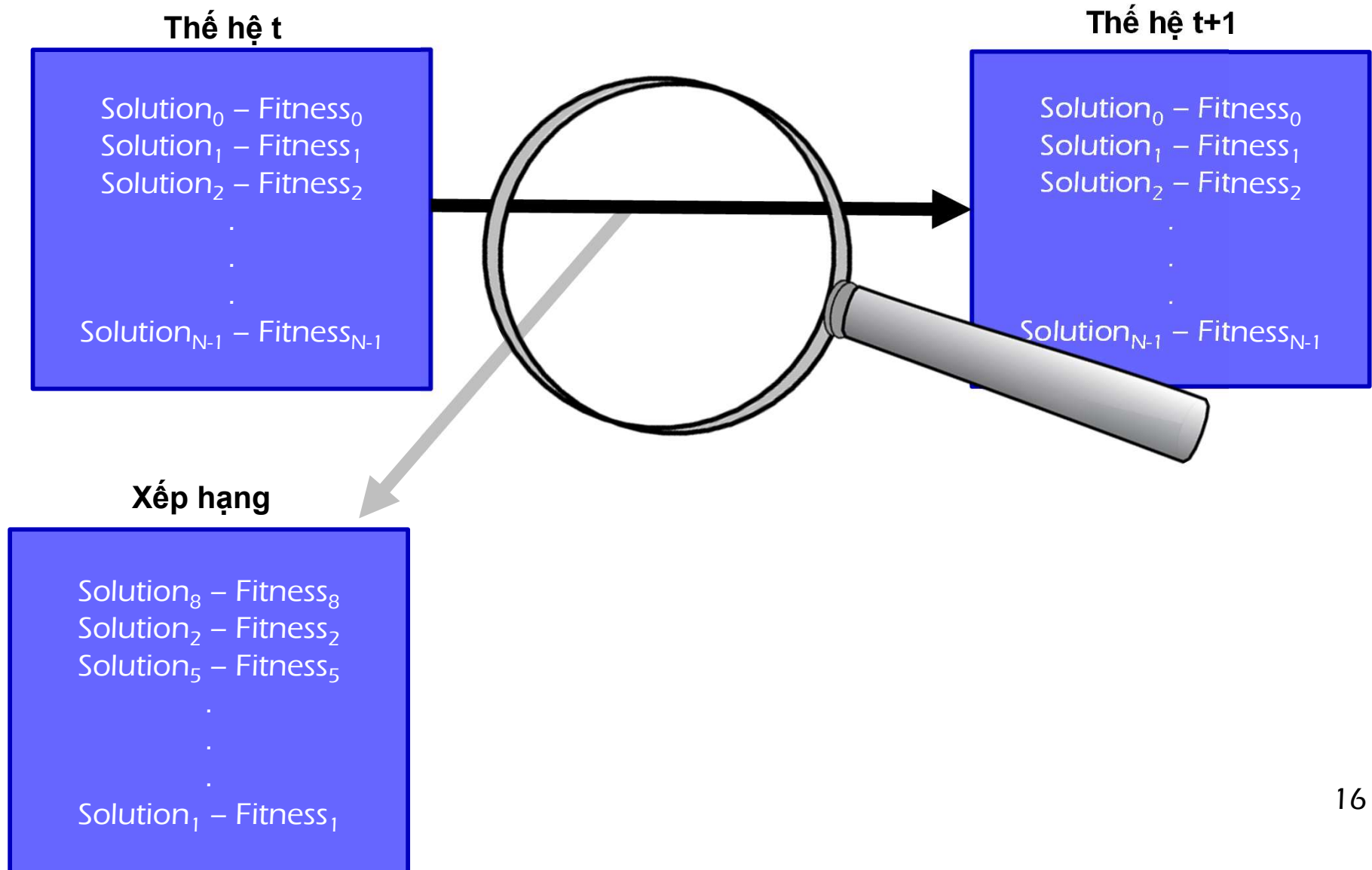


Thế hệ t+1





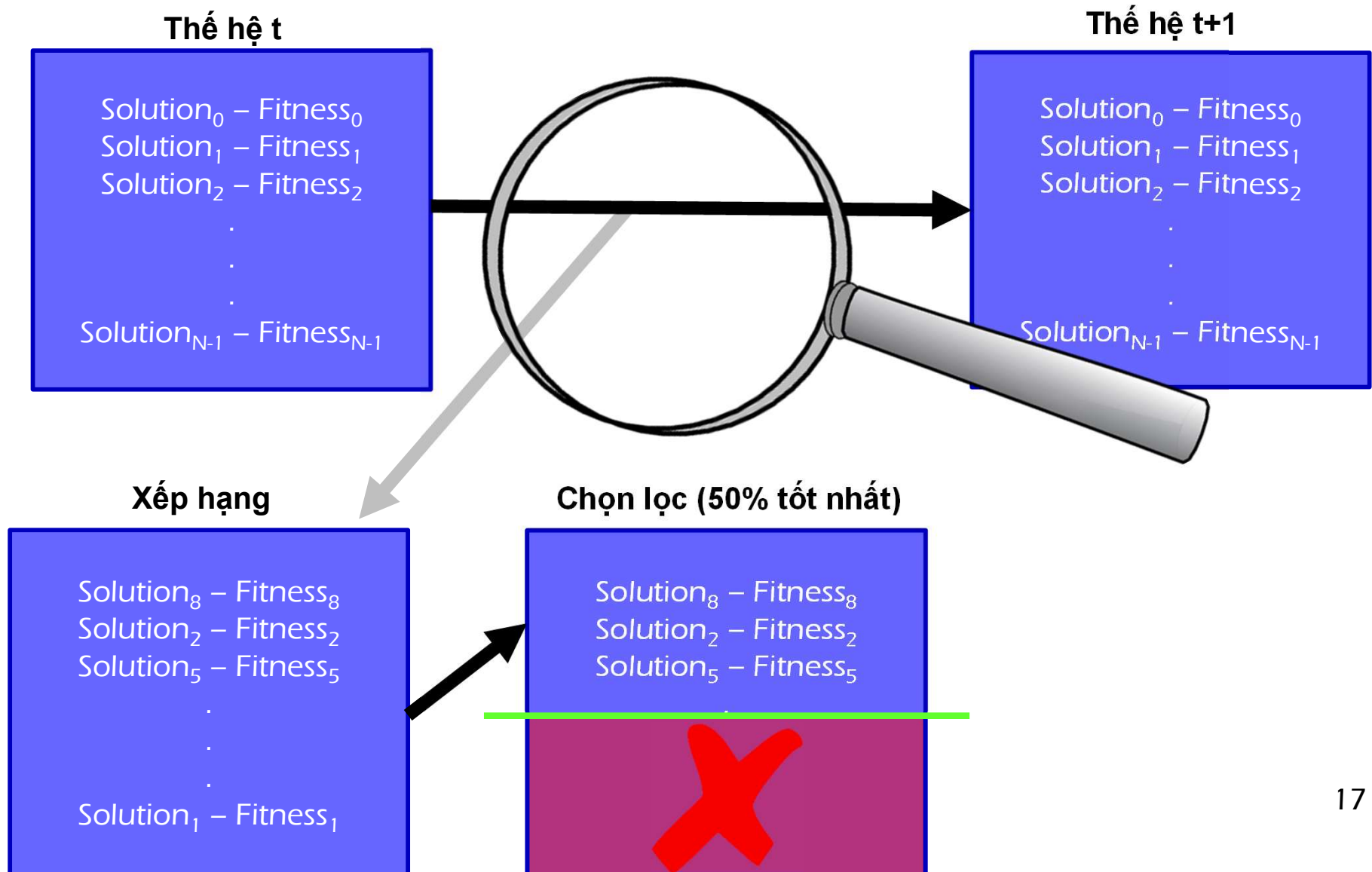
# Các thuật toán tiến hoá (Evolutionary Algorithms)





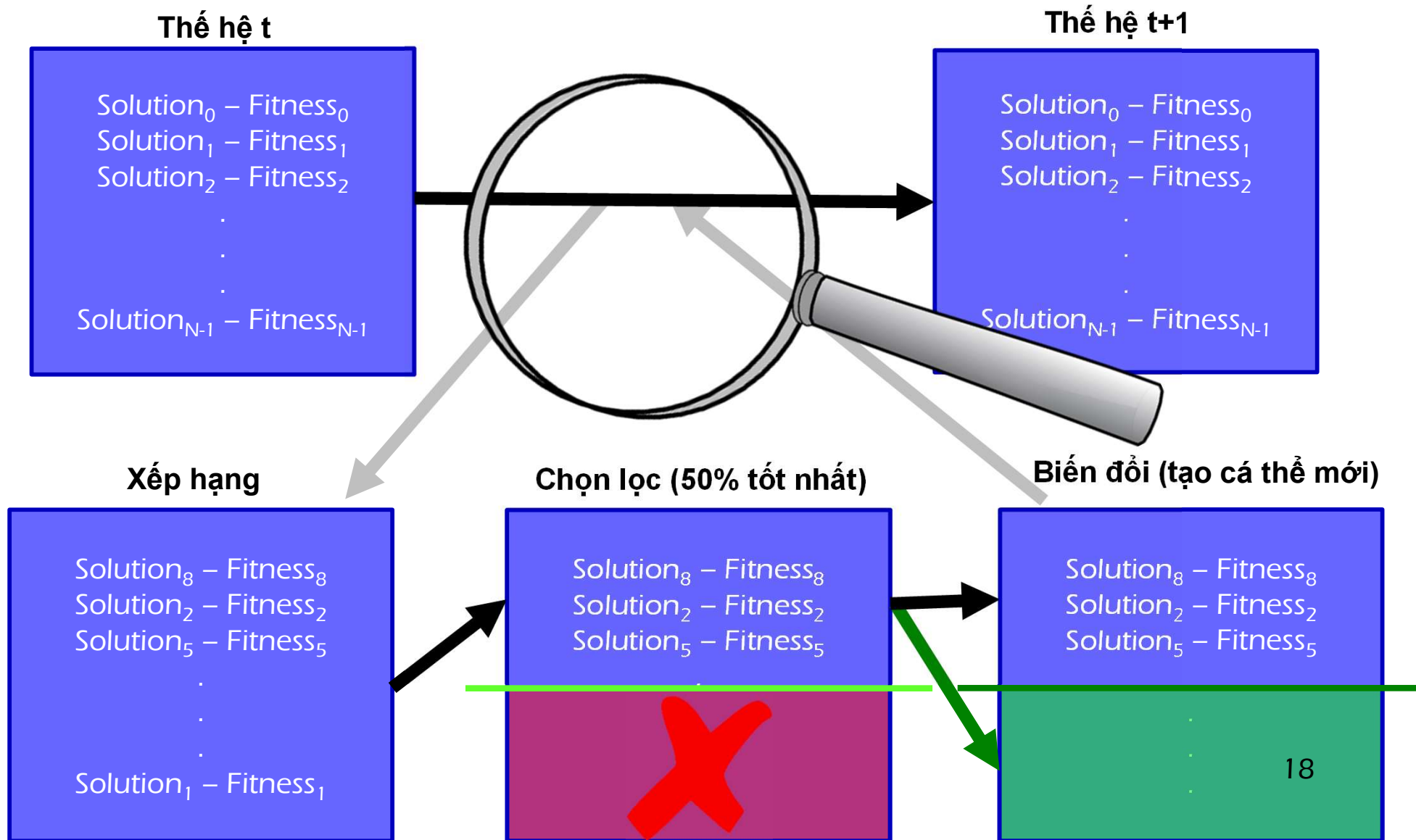


# Các thuật toán tiến hoá (Evolutionary Algorithms)





# Các thuật toán tiến hoá (Evolutionary Algorithms)



# Các thuật toán tiến hoá (cổ điển)

- Kiểu gen là các chuỗi nhị phân (binary strings):  
01000110, 1011100, 11100101,...
- Thuật giải di truyền (Genetic Algorithm) cổ điển.
- Kiểu gen là các vector số thực (real-valued vectors):  
(0.413, 1.313, 0.92), (1.521, 2.1843, 0.12),...
- Chiến lược tiến hoá (Evolution Strategy) cổ điển.
- Kiểu gen là các cấu trúc cây (tree structures), biểu diễn các chương trình / hàm số.
- Lập trình di truyền (Genetic Programming).



# Các thuật toán tiến hoá với mô hình (Model-Based EAs)

- Estimation-of-Distribution Algorithms
  - Extended Compact Genetic Algorithms (ECGA)
  - Bayesian Optimization Algorithm (BOA)
  - Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
  - Adapted Maximum-Likelihood Gaussian Model Iterated Density Estimation Evolutionary Algorithm (AMaLGaM)
  - Gene-pool Optimal Mixing Evolution Algorithms (GOMEA) / Linkage Tree Genetic Algorithm (LTGA).
  - ...



# Các thuật toán lấy cảm hứng từ sinh học (bio-inspired)

---

- Particle Swarm Optimization (PSO)
- Ant Colony Optimization (ACO)
- Differential Evolution (DE)
- Artificial Immune System (AIS)
- Artificial Bee Colony Algorithm (ABC)
- ...

# Lưu ý

- Hầu hết các thuật toán tiến hoá đều có 3 thành phần chính:
  1. Một quần thể.
  2. Một phép biến đổi.
  3. Một phép chọn lọc.
- Mục tiêu của các thuật toán tiến hoá là để giải bài toán cần giải quyết, KHÔNG phải để mô phỏng các hiện tượng tự nhiên.
- Các thuật ngữ sinh học được sử dụng để tiện cho việc trao đổi ý tưởng trong nghiên cứu và ứng dụng → tránh lạm dụng.

# Thuật giải di truyền (Genetic Algorithm)

- Được đề xuất bởi John Holland (1975).
- Kiểu gen là các chuỗi nhị phân (binary strings) có chiều dài cố định  $l$ .

```
1.  $t \leftarrow 0$ 
2.  $P^t \leftarrow \text{initializeAndEvaluateInitialIndividuals}(n)$ 
3. while terminationCriteriaNotSatisfied(  $P^t$  ) do
    1.  $S^t \leftarrow \text{selectParents}(P^t, n)$ 
    2.  $O^t \leftarrow \text{createAndEvaluateOffspring}(S^t, n)$ 
    3.  $P^{t+1} \leftarrow O^t$ 
    4.  $t \leftarrow t + 1$ 
```



# Khởi tạo quần thể (Population Initialization)

- Quần thể đầu tiên thường được khởi tạo một cách ngẫu nhiên với phân bố đồng nhất (uniformly random).

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1
1	0	0	0	1	1	0	0	1	1
0	1	1	1	0	0	0	0	1	0
0	0	1	1	1	0	0	0	0	1
1	0	0	1	1	1	0	1	0	0





# Khởi tạo quần thể (Population Initialization)

- Quần thể đầu tiên thường được khởi tạo một cách ngẫu nhiên với phân bố đồng nhất (uniformly random).
  - Với mỗi bit của từng cá thể, phát sinh ngẫu nhiên một số  $r \in [0..1]$ .
  - $r < 0.5 \rightarrow$  bit nhận giá trị 0.
  - $r \geq 0.5 \rightarrow$  bit nhận giá trị 1.
  - Cách tốt hơn?
- Có thể sử dụng thông tin về bài toán đang xét (nếu có) để khởi tạo quần thể với phân bố phù hợp hơn.



# Chọn lọc tỷ lệ (Proportional Selection)

- Phép chọn lọc: Lựa chọn (ngẫu nhiên) những cá thể (có độ thích nghi cao) làm cá thể cha mẹ (parents) để sinh ra những cá thể mới (cá thể con – offspring).
- Chọn lọc theo tỷ lệ: Xác suất một cá thể được chọn làm cá thể cha mẹ tỷ lệ thuận với độ thích nghi của cá thể đó.
- Những giải pháp tốt hơn được ưu tiên lựa chọn nhiều hơn để tạo ra những giải pháp mới.



# Chọn lọc tỷ lệ (Proportional Selection)

- $P$  là quần thể có  $n$  cá thể.
- $P_i$  là cá thể thứ  $i$  của  $P$ .
- $\text{fitness}[P_i]$  là độ thích nghi của  $P_i$ .
- Tổng giá trị thích nghi của  $P$ :  $\sum_{i=0}^{n-1} \text{fitness}[P_i]$
- Xác suất cá thể  $P_i$  được chọn = tỷ lệ đóng góp của  $P_i$  cho tổng giá trị thích nghi của  $P$ :

$$p_i^S = \frac{\text{fitness}[P_i]}{\sum_{i=0}^{n-1} \text{fitness}[P_i]}$$



# Chọn lọc tỷ lệ (Proportional Selection)

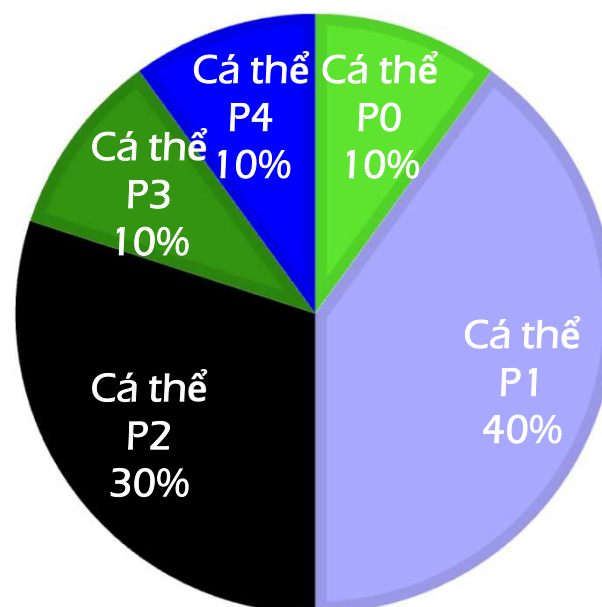
Tạo ra tập lựa chọn  $S$  (selection set) có  $n$  cá thể cha mẹ từ quần thể  $P$  gồm  $n$  cá thể.

- Bước 1: Tính xác suất được lựa chọn  $p_i^S$  của từng cá thể.
- Bước 2: Tính xác suất tích lũy (cumulative probability)  $p_i^{S,C}$  của từng cá thể:  $p_i^{S,C} = \sum_{j=0}^i p_j^S$
- Bước 3: Phát sinh ngẫu nhiên 1 số thực  $r \in [0..1)$ .
- Bước 4: Chọn cá thể  $P_i$  sao cho  $p_{i-1}^{S,C} \leq r < p_i^{S,C}$  với  $i \in \{0, 1, \dots, n-1\}$  và  $p_{-1}^{S,C} = 0$ .  
Lặp lại bước 3 – 4 tới khi chọn đủ  $n$  cá thể.



# Chọn lọc theo tỷ lệ (Proportional Selection)

- Một quần thể có 5 cá thể với độ thích nghi là  $(1, 4, 3, 1, 1)$ .
- Tỷ lệ đóng góp của từng cá thể vào tổng giá trị thích nghi là  $(0.1, 0.4, 0.3, 0.1, 0.1) \rightarrow$  xác suất được chọn  $p_i^S$  của từng cá thể.
- Xác suất tích lũy  $p_i^{S,C}$  của từng cá thể là  $(0.1, 0.5, 0.8, 0.9, 1.0)$ .





# Chọn lọc tỷ lệ (Proportional Selection)

## XÁC SUẤT LỰA CHỌN

Cá thể P4  
10%

Cá thể P0  
10%

Cá thể P3  
10%

Cá thể P2  
30%

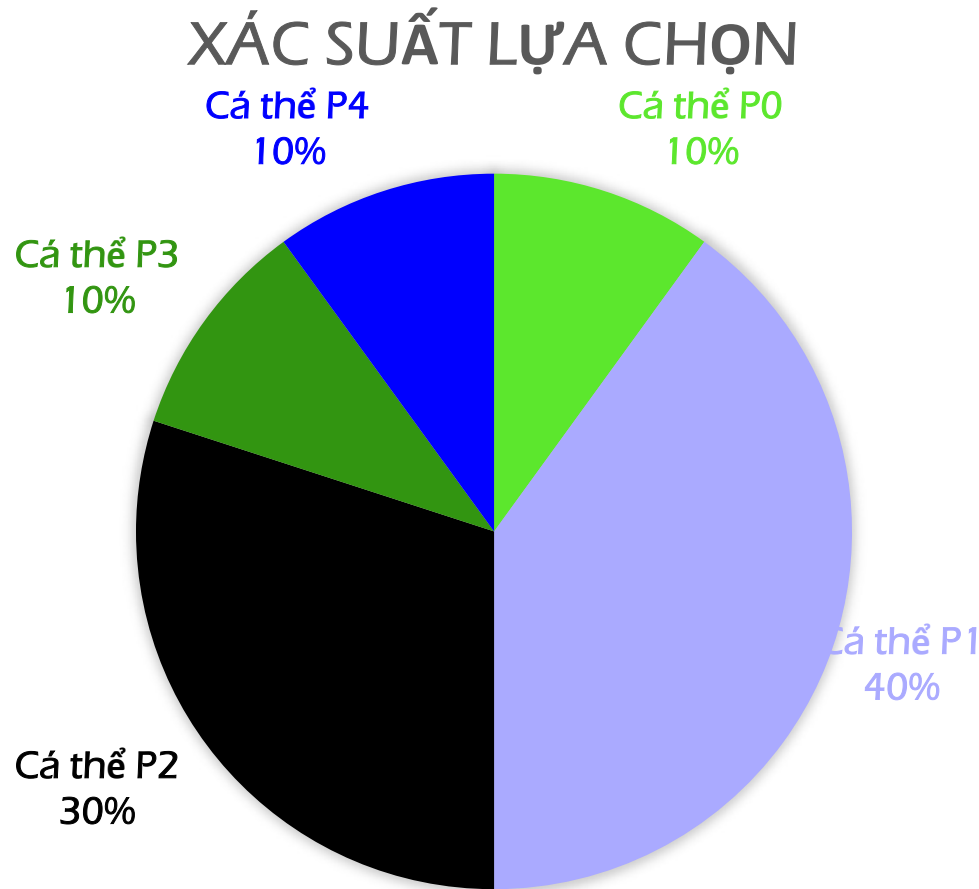
Cá thể P1  
40%

Ví dụ:

- $r = 0.37 \rightarrow$  lựa chọn cá thể  $P_1$
- $r = 0.07 \rightarrow$  lựa chọn cá thể  $P_0$
- $r = 0.39 \rightarrow$  lựa chọn cá thể  $P_1$
- $r = 0.72 \rightarrow$  lựa chọn cá thể  $P_2$
- $r = 0.34 \rightarrow$  lựa chọn cá thể  $P_1$



# Chọn lọc tỷ lệ (Proportional Selection)



Ví dụ:

- $r = \dots \rightarrow$  lựa chọn ...
  - $r = \dots \rightarrow$  lựa chọn ...
  - $r = \dots \rightarrow$  lựa chọn ...
  - $r = \dots \rightarrow$  lựa chọn ...
  - $r = \dots \rightarrow$  lựa chọn ...
- Proportional selection còn được gọi là Roulette wheel selection.

# Các phép biến đổi (Variation)

- Phép biến đổi: được thực hiện trên tập lựa chọn  $S$  (selection set) nhằm tạo ra các cá thể mới (offspring) từ các cá thể cha mẹ (parents).
- Thuật giải di truyền GA có 2 phép biến đổi chính:
  - Lai ghép (crossover), còn được gọi là Tái tổ hợp (recombination).
  - Đột biến (mutation)



# Phép biến đổi – Lai ghép

Phép lai ghép: Tạo ra cá thể mới bằng cách kết hợp kiểu gen của những cá thể (có độ thích nghi cao) trong quần thể.

Cá thể cha mẹ (Parents)	Phép lai (Crossover)	Cá thể con (Offspring)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Lai một điểm (One-point crossover – 1X)	0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Lai hai điểm (Two-point crossover – 2X)	0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Lai đồng nhất (Uniform crossover – UX)	1 0 1 1 0 0 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 1



# Lai một điểm

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1



# Lai một điểm

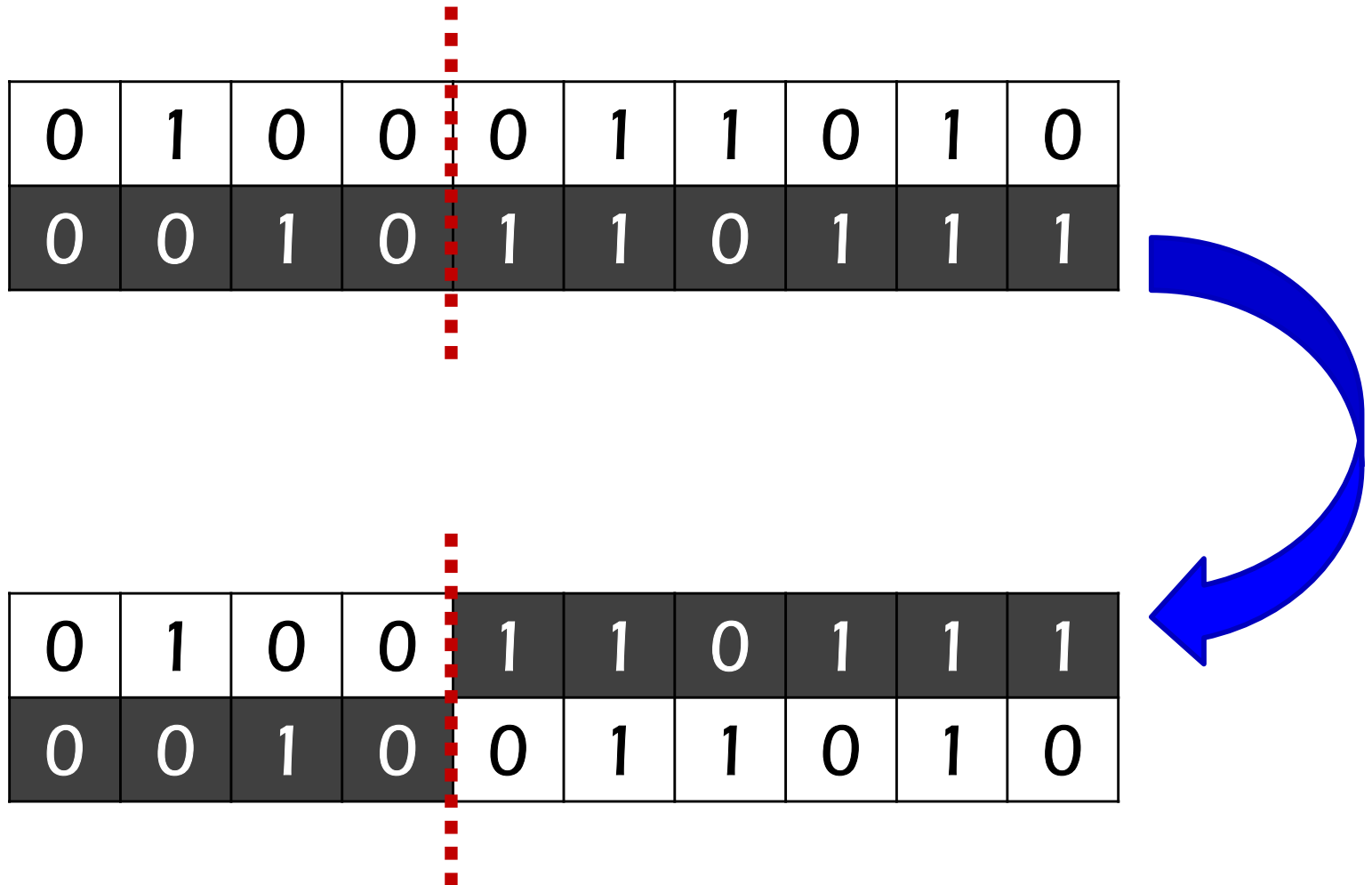
0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1



**UIT**

TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# Lai một điểm





# Lai hai điểm

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1



# Lai hai điểm

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1



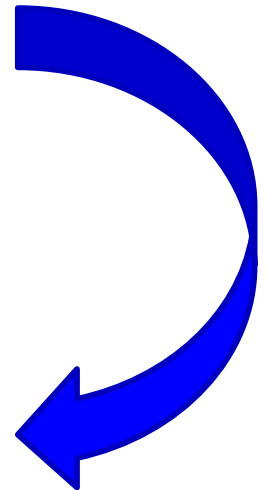
**UIT**

TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# Lai hai điểm

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1

0	1	0	0	1	1	0	1	1	0
0	0	1	0	0	1	1	0	1	1





# Lai đồng nhất

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1



# Lai đồng nhất

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1

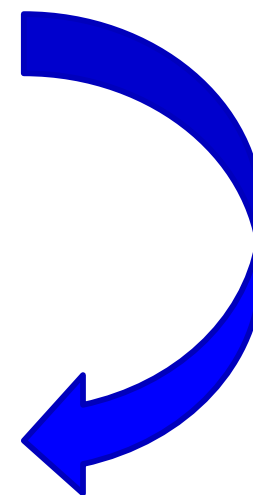
0.14	0.35	0.67	0.49	0.72	0.53	0.03	0.91	0.6	0.25
------	------	------	------	------	------	------	------	-----	------

# Lai đồng nhất

0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1

0.14	0.35	0.67	0.49	0.72	0.53	0.03	0.91	0.6	0.25
------	------	------	------	------	------	------	------	-----	------

0	0	0	0	0	1	0	0	1	1
0	1	1	0	1	1	1	1	1	0



# Phép biến đổi – Lai ghép

- Lựa chọn xác suất lai  $p_c$ .
- Ghép cặp các cá thể trong tập lựa chọn  $S$ .
- Với mỗi cặp đôi cá thể cha mẹ, phát sinh ngẫu nhiên 1 số thực  $r \in [0..1)$ .
- Nếu  $r < p_c$ , tiến hành phép lai tạo ra 2 cá thể con mới.
- Ngược lại, 2 cá thể con là sao chép (clone) của 2 cá thể cha mẹ.

# Phép biến đổi – Đột biến

- Phép đột biến: Tạo ra cá thể mới bằng cách phát sinh những biến đổi ngẫu nhiên trên kiểu gen của các cá thể trong quần thể.
- Lựa chọn xác suất đột biến  $p_m$  (thường nhỏ).
- Với mỗi bit của từng cá thể, phát sinh ngẫu nhiên 1 số thực  $r \in [0..1)$ .
  - Nếu  $r < p_m$ , đảo ngược giá trị của bit ( $0 \rightarrow 1$  hoặc  $1 \rightarrow 0$ ).
  - Ngược lại, giữ nguyên giá trị của bit.

# Đột biến

Ví dụ: Xác suất đột biến 0.05

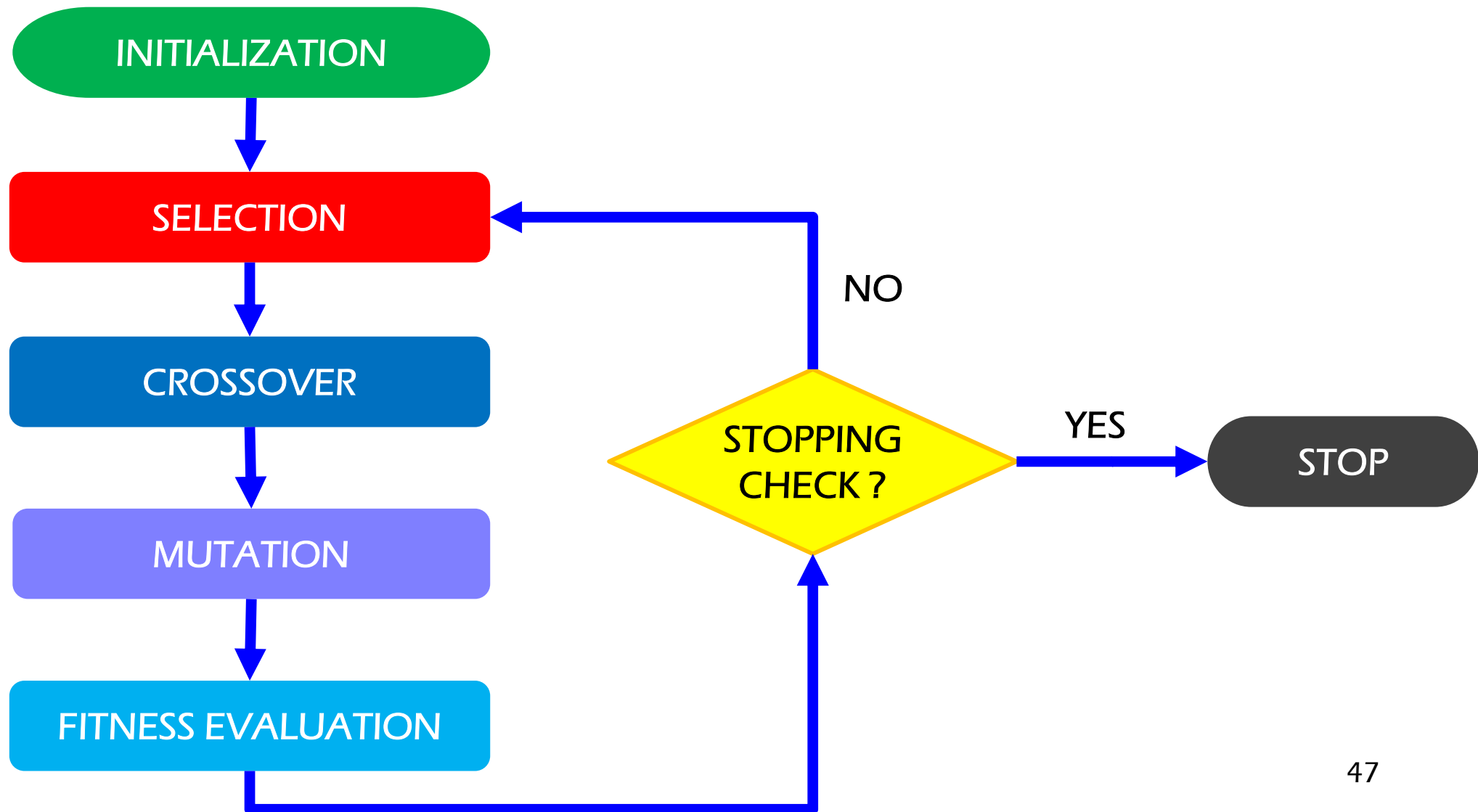
0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1	1	1
1	0	0	0	1	1	0	0	1	1
0	1	1	1	0	0	0	0	1	0
0	0	1	1	1	0	0	0	0	1
1	0	0	1	1	1	1	0	0	0



0	1	0	0	0	1	1	0	1	0
0	0	1	1	1	1	0	1	1	1
1	0	0	0	1	1	0	0	1	1
0	1	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	1
1	0	0	1	1	1	1	0	0	0

# Điều kiện dừng (Termination)

- Sử dụng hết tài nguyên tính toán.
  - Thời gian.
  - Số lần gọi hàm lượng giá (evaluation function calls).
- Quần thể không còn độ đa dạng (diversity).
  - Các cá thể trong quần thể gần giống nhau về kiểu gen.
  - Các cá thể trong quần thể có độ thích nghi gần giống nhau.
- Đạt được độ thích nghi mong muốn.
  - Tìm được một giải pháp chấp nhận được cho bài toán.



## Ví dụ

Độ thích nghi trung bình của quần thể là 177.0

P0	Fitness
10010	324
01100	144
01001	81
10100	400
01000	64
00111	49



# Ví dụ

Độ thích nghi trung bình của quần thể P1 là 318.8

P0	Fitness	Selection	Crossover	Mutation	Fitness
10010	324	10010	10   010	10100	400
01100	144	10100	10   100	10010	324
01001	81	01000	010   00	01010	100
10100	400	10010	100   10	1000 <b>1</b>	289
01000	64	01100	0   1100	00100	16
00111	49	10100	1   0100	11100	784

# Ví dụ

Độ thích nghi trung bình của quần thể P2 là 426.3

P1	Fitness	Selection	Crossover	Mutation	Fitness
10100	400	11100	111   00	11110	900
10010	324	10010	100   10	10000	256
01010	100	10001	10   001	00100	16
10001	289	10100	10   100	10001	289
00100	16	10001	100   01	10000	256
11100	784	11100	111   00	11101	841

# Ví dụ

Độ thích nghi trung bình của quần thể P3 là 654.5

P2	Fitness	Selection	Crossover	Mutation	Fitness
11110	900	11110	111   10	11101	841
10000	256	10001	100   01	10010	324
00100	16	11101	11   101	11000	576
10001	289	10000	10   000	10101	441
10000	256	11110	1111   0	11111	961
11101	841	11101	1110   1	11100	784

# Ví dụ

Độ thích nghi trung bình của quần thể P4 là 862.0

P3	Fitness	Selection	Crossover	Mutation	Fitness
11101	841	11101	11   101	11111	961
10010	324	11111	11   111	11101	841
11000	576	11100	1   1100	11101	841
10101	441	11101	1   1101	11100	784
11111	961	11111	111   11	11100	784
11100	784	11000	110   00	11111	961

# Tiến hoá (Evolution)

Tiến hoá diễn ra trên một quần thể (population) gồm nhiều cá thể (individual). Mỗi thế hệ nhiều cá thể con được sinh ra.

- Tiến hoá đã tạo ra nhiều sinh vật có độ phức tạp cao (Tính hữu hiệu – effectiveness).
- “Xử lý song song”, nhiều lựa chọn có thể được thử nghiệm cùng lúc (Tính hiệu quả – efficiency).
- Nếu một vài cá thể thất bại, quá trình tiến hoá vẫn diễn ra (Tính bền vững – robustness).
- Ý tưởng tổng quát đơn giản, dễ tùy biến (Tính đơn giản – simplicity).

# Tham khảo

Prof. dr. P.A.N. Bosman's speech on  
Evolutionary Intelligence.

[https://www.youtube.com/watch?v=sOyoYkkbDn0&ab\\_channel=CentrumWiskunde%26Informatica](https://www.youtube.com/watch?v=sOyoYkkbDn0&ab_channel=CentrumWiskunde%26Informatica)