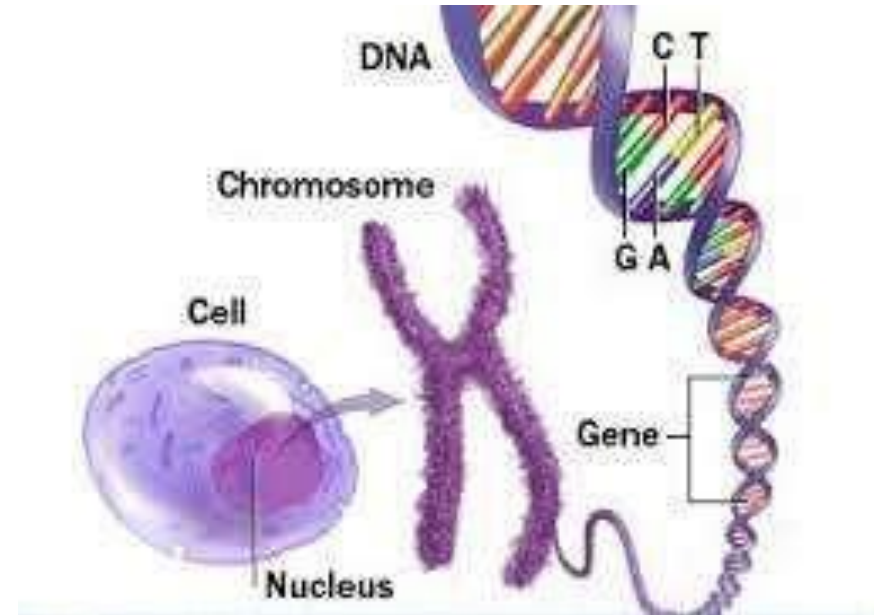
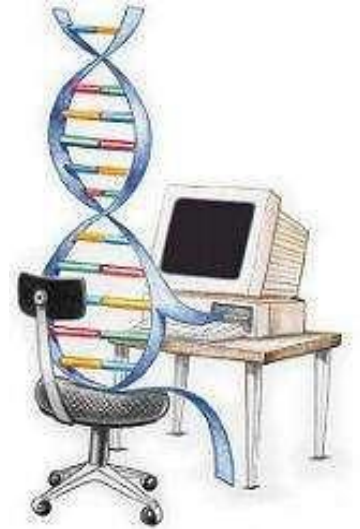
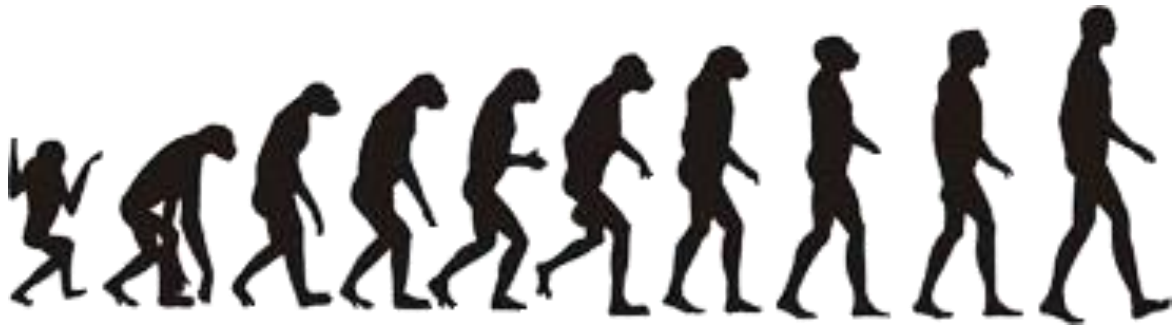


Genetic Algorithms

Trương Hải Bằng, VNU HCM - 2018

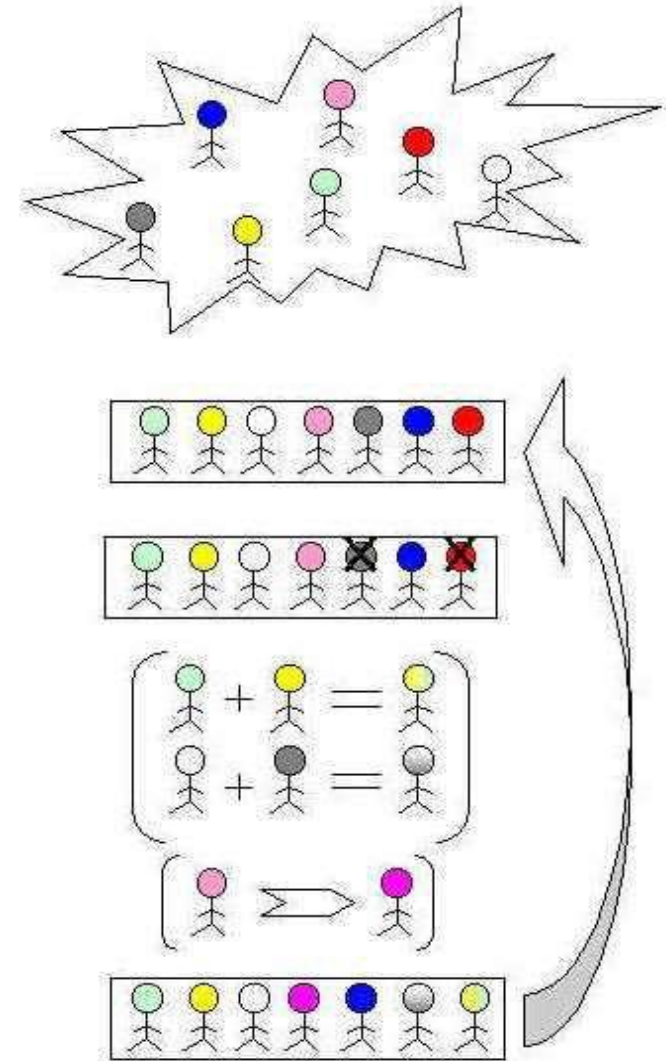


Nội dung

- 1 Tổng quan
- 2 Thuật toán di truyền
- 3 Ứng dụng bài toán TSP
- 4 Ứng dụng bài toán 8 - Queen

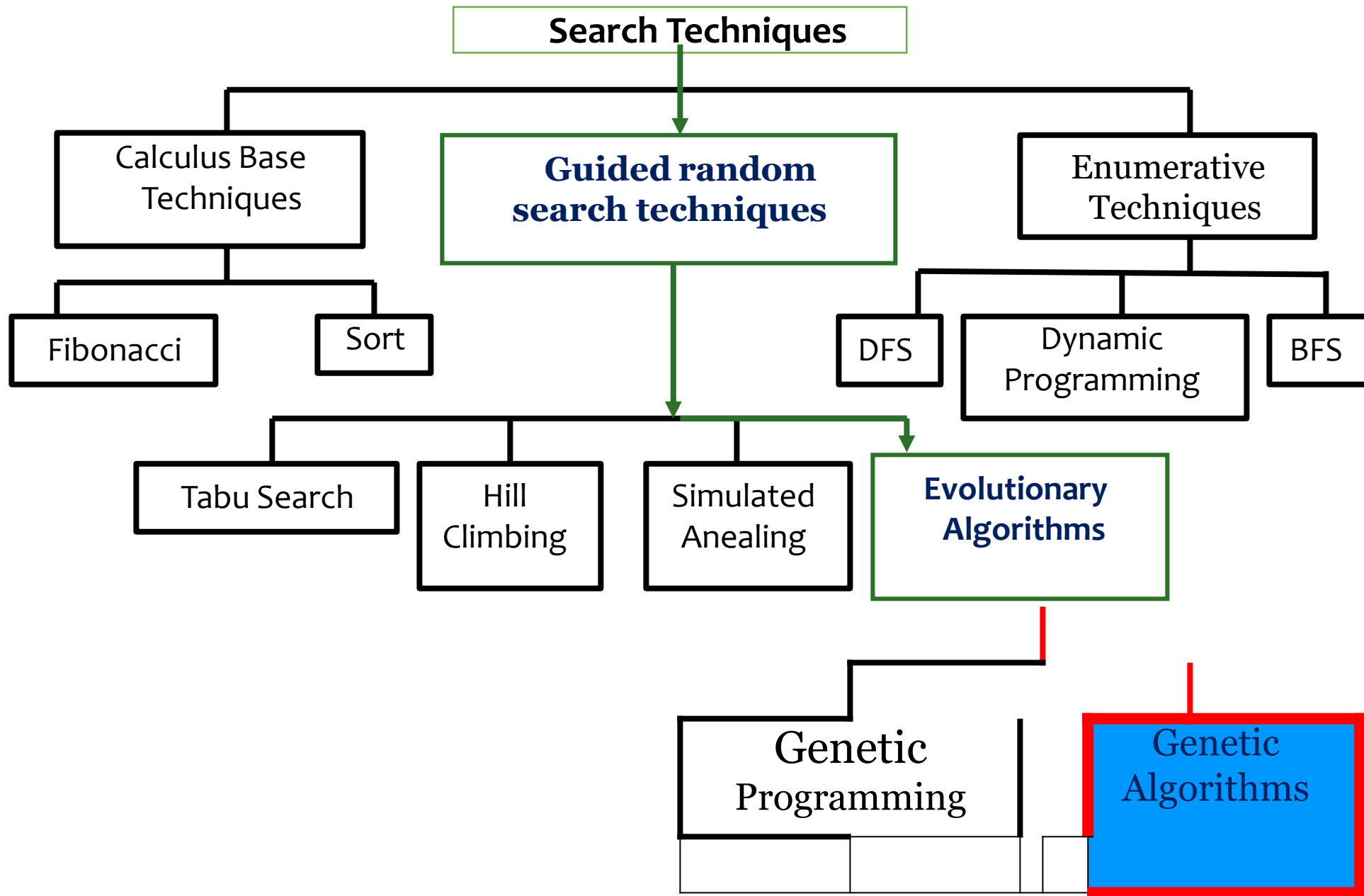
TỔNG QUAN

- ✓ Thuật giải di truyền (Genetic Algorithm - GA) là kỹ thuật mô phỏng theo sự tiến hóa dựa trên thuyết tiến hóa muôn loài của Darwin để giải quyết các bài toán không giải được bằng các phương pháp truyền thống (Hard Computing).
- ✓ GA không tìm ra lời giải tốt nhất, mà tìm kiếm một giải pháp tốt để giải các bài toán một cách hiệu quả.



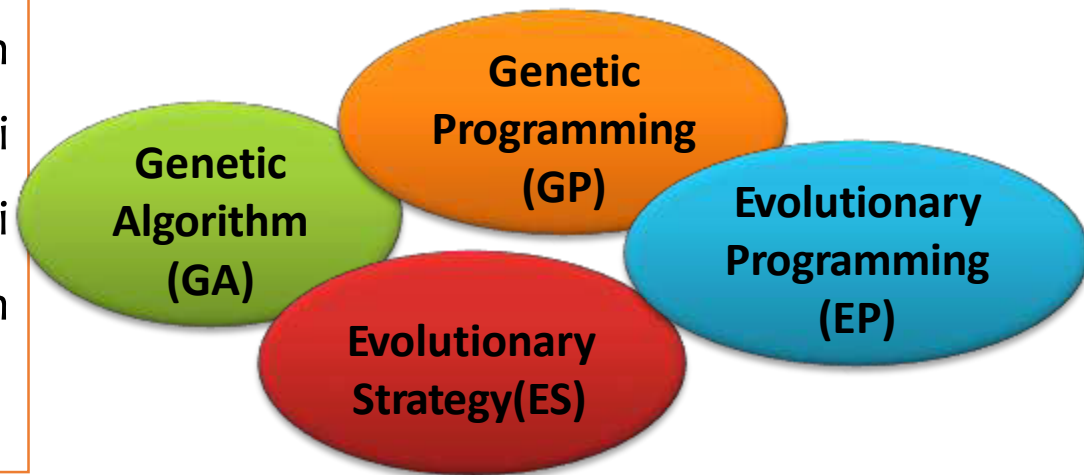
History Of Genetic Algorithms

- ❑ “Evolutionary Computing” was introduced in the 1960s by **I. Rechenberg**.
- ❑ John Holland wrote the first book on Genetic Algorithms ‘**Adaptation in Natural and Artificial Systems**’ in 1975.
- ❑ In 1992 **John Koza** used genetic algorithm to evolve programs to perform certain tasks. He called his method “**Genetic Programming**”.



GA - Các vấn đề liên quan

GA mô phỏng sự tiến hóa tự nhiên bằng cách tạo ra một quần thể cá nhân, đánh giá tính khả thi của chúng, tạo ra một quần thể mới thông qua các hoạt động di truyền và lặp lại quá trình này cho đến khi thỏa một điều kiện nào đó (hàm thích nghi).

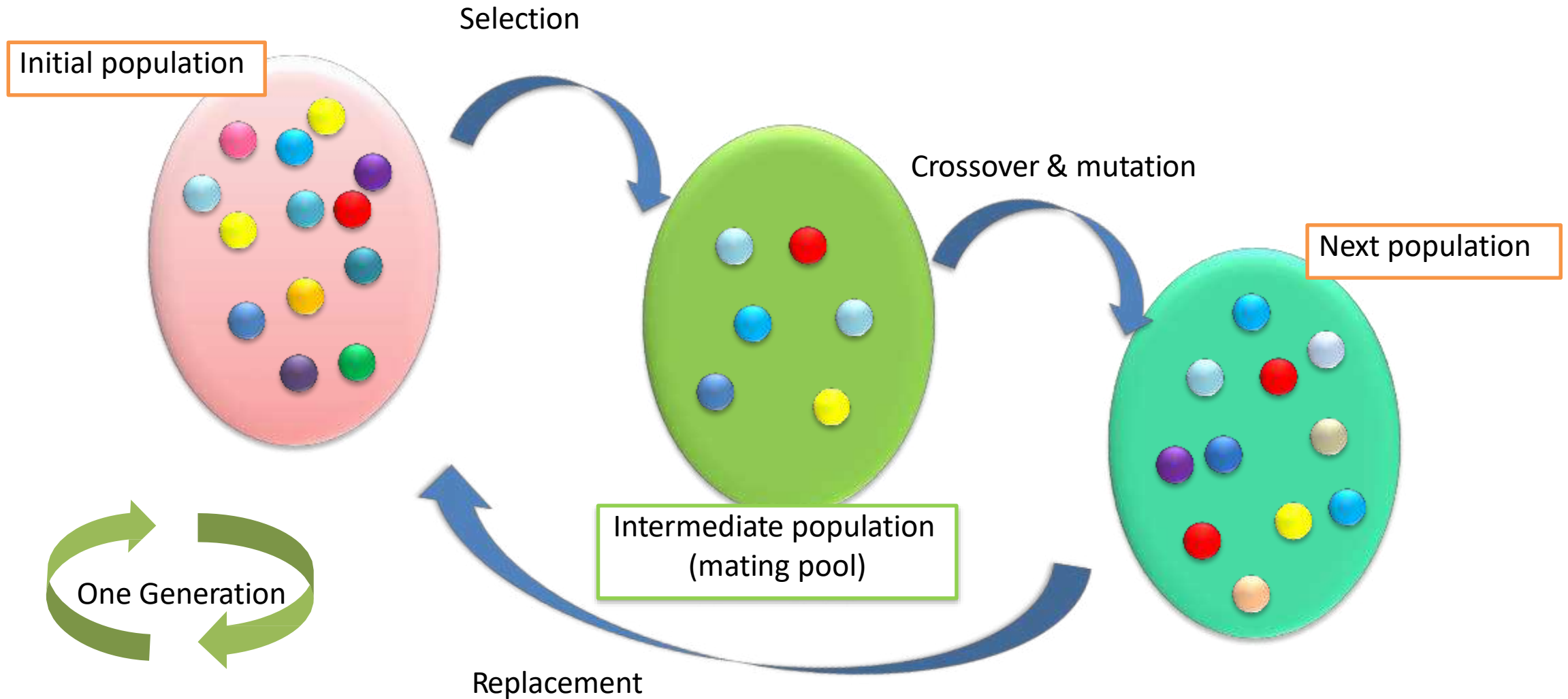


Applications of GAs

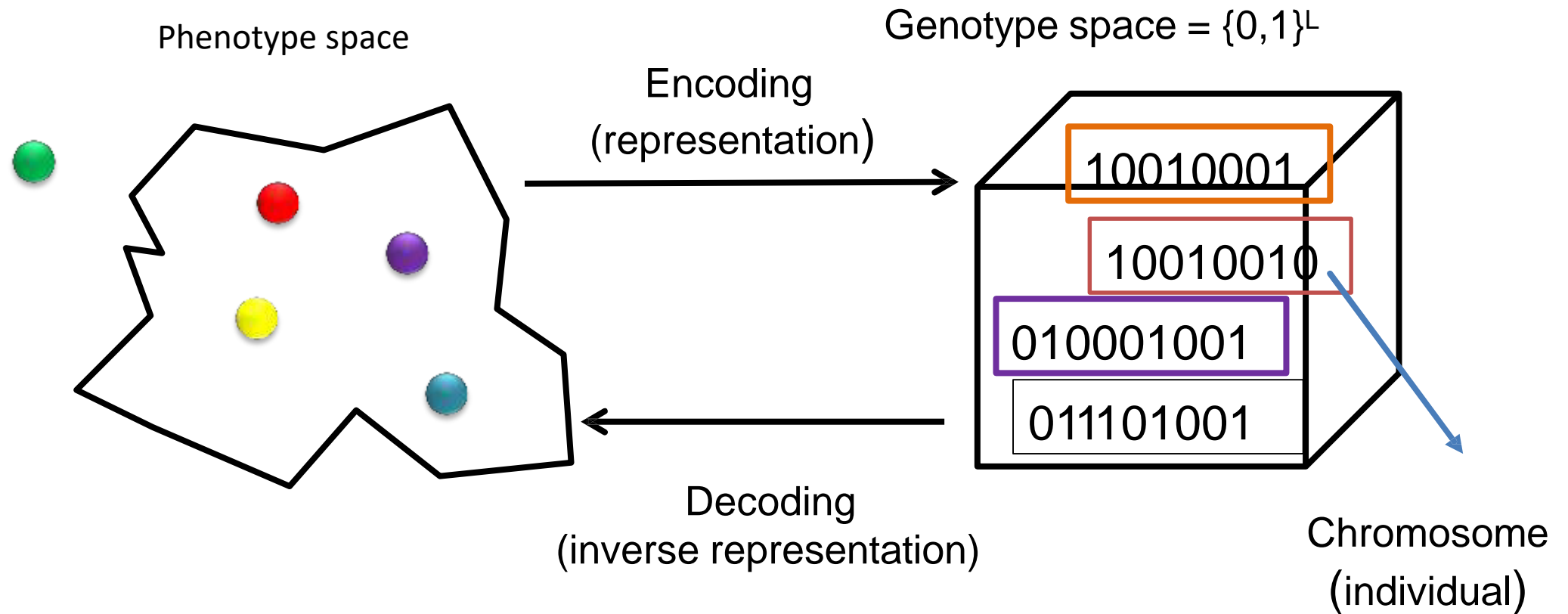
- Tối ưu hóa số và Tổ hợp: Lập lịch, TSP
- Lập trình Tự động: Lập trình gen
- Học Máy: Phân lớp, dự báo
- Kinh tế: Chiến lược đấu thầu, xu hướng cổ phiếu
- Sinh thái học: Mối liên kết giữa ký chủ và ký sinh trùng, dòng chảy tài nguyên, các cuộc đua cánh tay sinh học
- Di truyền dân số: Tính khả thi của sự lan truyền gen
- Hệ thống xã hội: Sự tiến hoá của hành vi xã hội trong các quần thể côn trùng
- ...

Genetic Algorithms Implementation

GA overview

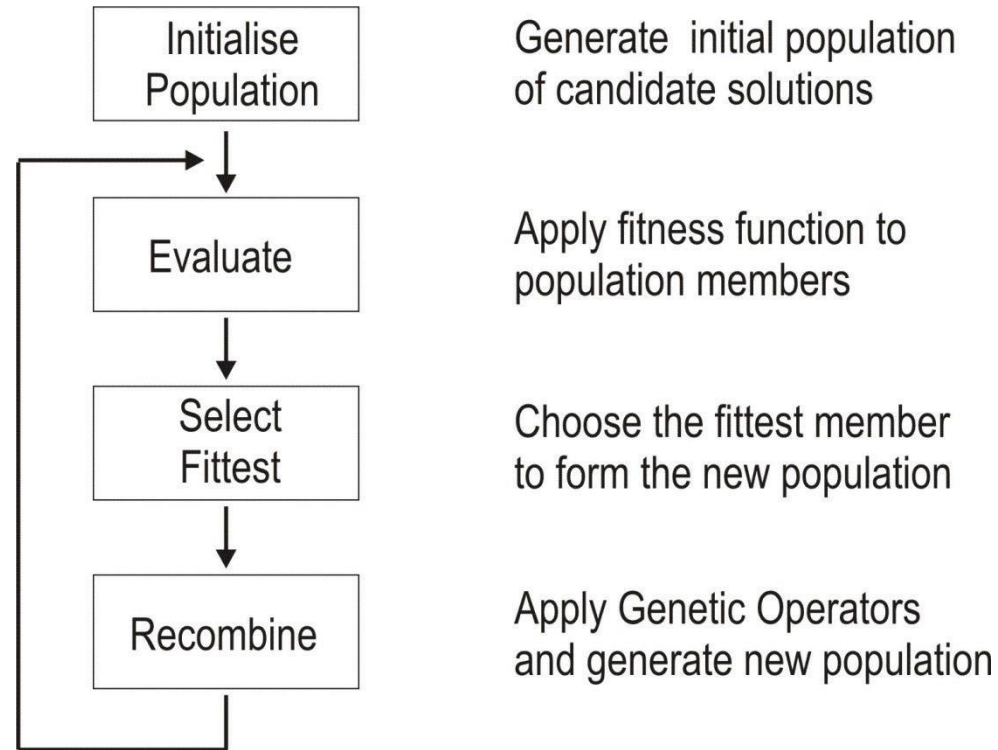


Các Thuật ngữ

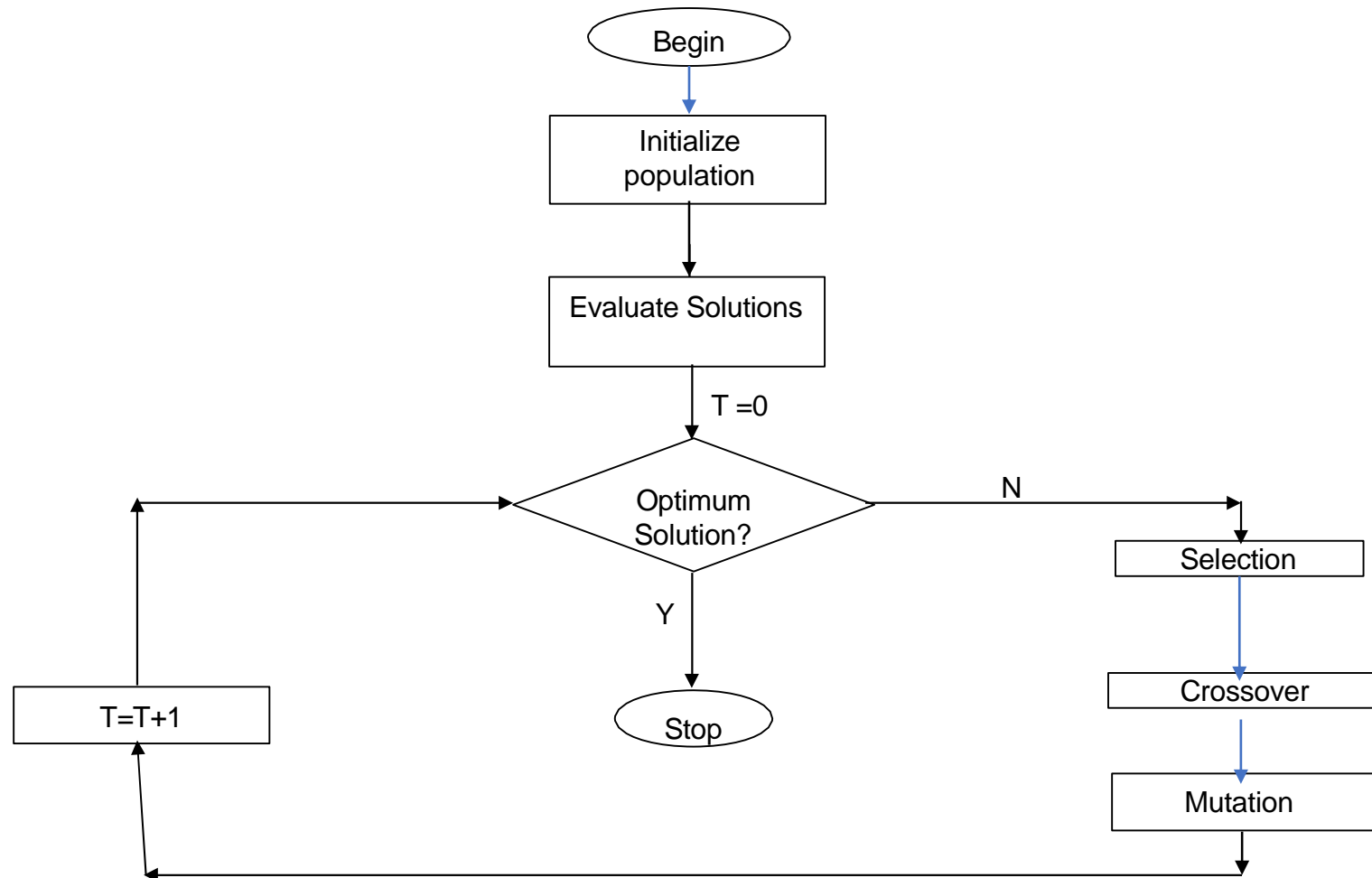


Computational Model

Main GA algorithm



Working Mechanism Of GAs



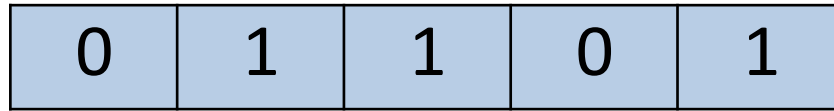
Simple Genetic Algorithm

```
function GENETIC-ALGORITHM(population, FITNESS-FN, crossover-rate, mutation-rate)  
  returns an individual  
  inputs: population, a set of individuals  
           FITNESS-FN (the fitness function)  
  repeat  
    new_population  $\leftarrow$  empty set  
    calculate the fitness value of each individual  
    loop for i from 1 to SIZE(population) do  
      x  $\leftarrow$  RANDOM-SELECTION(population, FITNESS-FN)  
      add x to new_population  
    loop for i from 1 to SIZE(population) * crossover-rate do  
      x  $\leftarrow$  RANDOM-SELECTION(new_population)  
      y  $\leftarrow$  RANDOM-SELECTION(new_population)  
      x, y  $\leftarrow$  REPRODUCE(x, y)  
    loop for i from 1 to SIZE(population) * mutation-rate do  
      x  $\leftarrow$  RANDOM-SELECTION(new_population)  
      x  $\leftarrow$  MUTATE(x)  
    population  $\leftarrow$  new_population  
  until the average fitness values are stable, or enough time has elapsed  
  return the best individual found in any population
```

GA Operators and Parameters

Mã hóa

$x=13$,
 $f(x)=169$



Chromosome

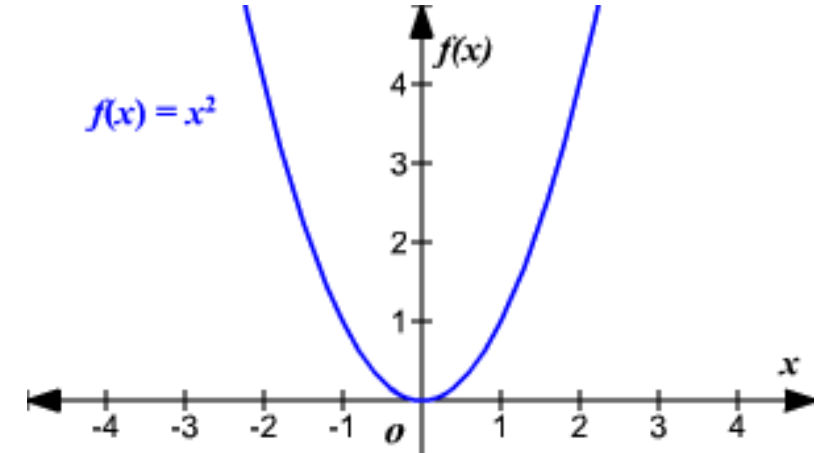
Gene

1. Mã hóa nhị phân

Chromosome A	101101100011
Chromosome B	010011001100

2. Mã hóa Hoán vị (TSP)

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9



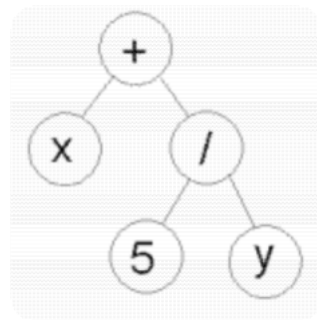
Hàm thích nghi:
 $\text{Max } f(x)=x^2, 0 < x < 31$

3. Mã hóa giá trị: (Tìm trọng số neural network)

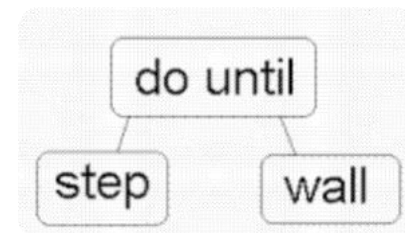
Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

4. Mã hóa cây:

Cho giá trị input và output, tìm một hàm tương ứng tốt nhất



(+ x (/ 5 y))



(do_until step wall)

Initialization Population

Khởi tạo (Initialization): tạo ra quần thể ban đầu, thường là chọn ngẫu nhiên, kích cỡ tùy ý, các cá thể của quần thể thường được thể hiện dưới dạng mảng bit.

$$\text{Max } f(x) = x^2$$

$$0 < x < 31, \text{ Pop Size} = 4$$

A series of numbers	Starting population (random)	Value of the variable x	$f(x) = x^2$ f_i
1	0 1 <u>1</u> 0 1	13	169
2	1 1 <u>0</u> <u>0</u> <u>0</u>	24	576
3	0 1 <u>0</u> <u>0</u> <u>0</u>	8	64
4	1 <u>0</u> <u>0</u> <u>1</u> <u>1</u>	19	361

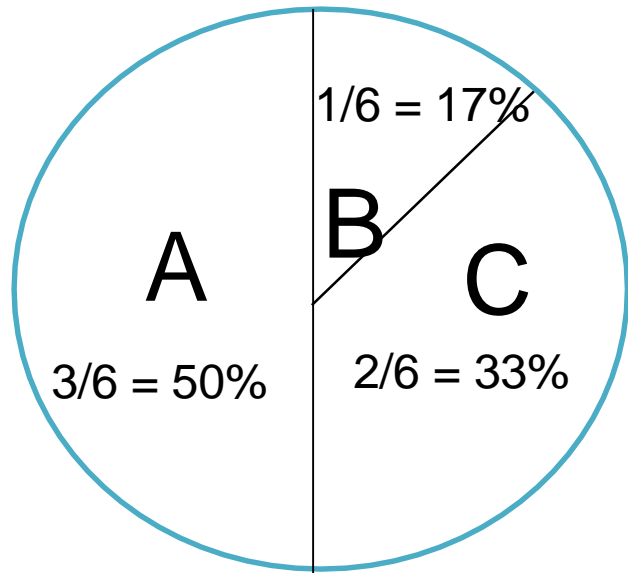
Selection

Chọn lọc (Selection): dựa vào hàm thích nghi (fitness), loại bỏ những cá thể xấu, chỉ để lại những cá thể tốt trong quần thể dành cho thế hệ tiếp theo.

Selection strategy- roulette wheel

– Implementation: roulette wheel technique

- Gán cho mỗi cá nhân là một phần của bánh xe roulette
- Quay bánh xe n lần để chọn n cá thể



fitness(A) = 3

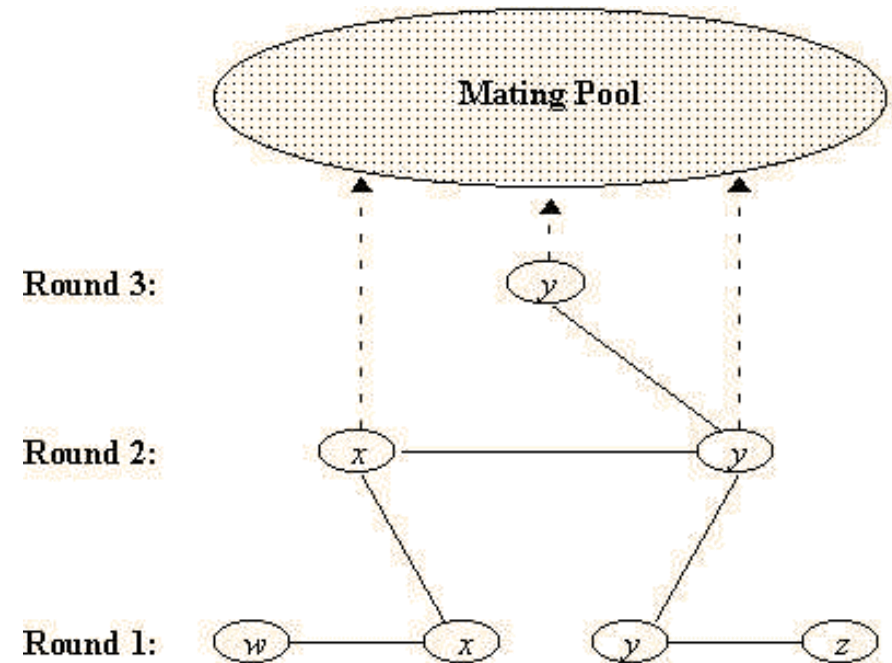
fitness(B) = 1

fitness(C) = 2

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

Selection strategy-Tournament

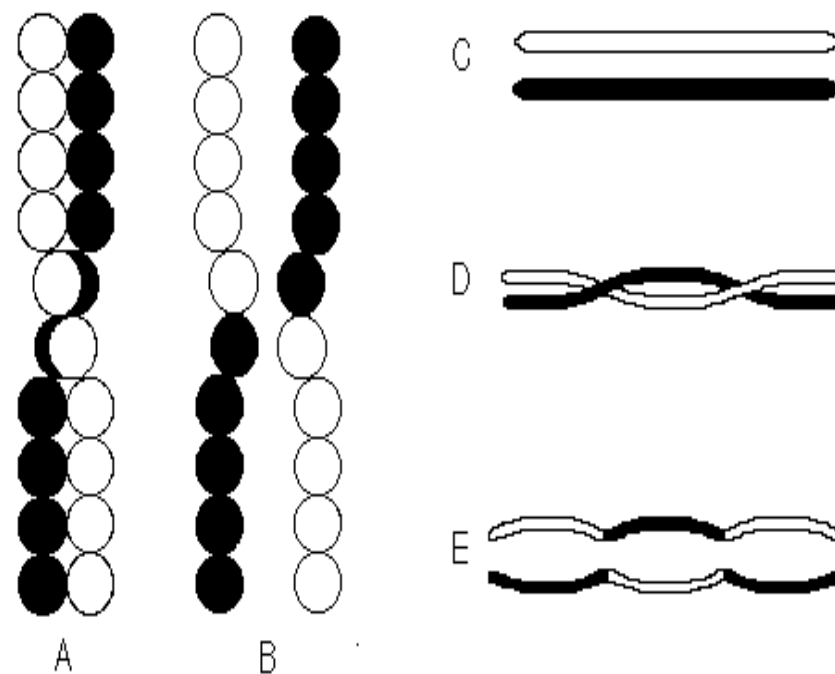
- Chiến lược lựa chọn giải đấu mang lại áp lực chọn lọc bằng sự cạnh tranh thi đấu giữa các cá thể N.
- Cá thể tốt nhất từ giải đấu là người có sức khỏe cao nhất - người chiến thắng
- Các cuộc thi đấu và người chiến thắng sẽ được giữ lại trong thế hệ tiếp theo.



Lai ghép - Crossover

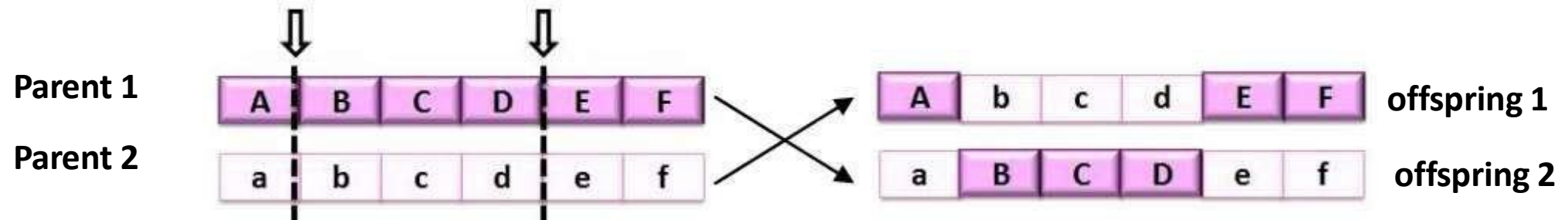
Lai ghép (Crossover): mỗi cặp cá thể được chọn sẽ đan chéo các thành phần với nhau để tạo ra cá thể mới, có nhiều phương thức lai ghép khác nhau.

Chromosome Crossing-over

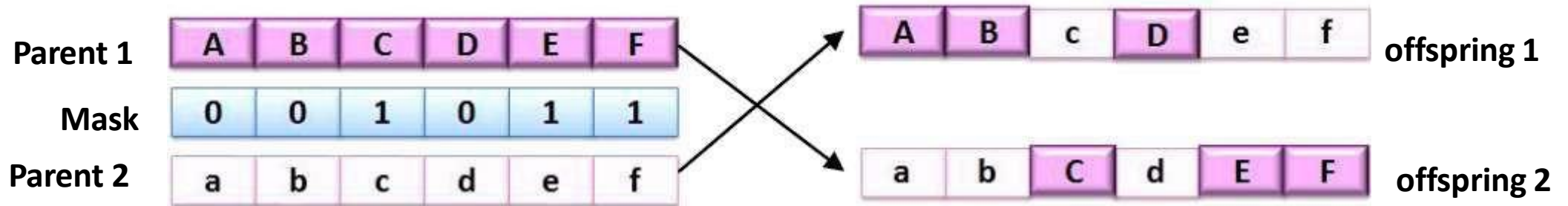


Toán tử Lai ghép

- Two point crossover



- Uniform crossover



$0.6 < \text{Probability of crossover (Pc)} < 0.9$

Crossover effects

First generation offspring	Randomly chosen cross-over partner	Cross-over point (random)	New population	Value of the variable x	$f(x) = x^2$ f_i/f_{avg} (rounded up)
0 1 1 0 1	2	4	0 1 1 0 0	12	144 (0.32, 0)
1 1 0 0 0	1	4	1 1 0 0 1	25	625 (1.42, 1)
1 1 0 0 0	4	2	1 1 0 1 1	27	729 (1.66, 2)
1 0 0 1 1	3	2	1 0 0 0 0	16	256 (0.58, 1)
Sum					1754
Average: f_{avg}					439
Maximum					729

Exploration



Invalid offspring!!!

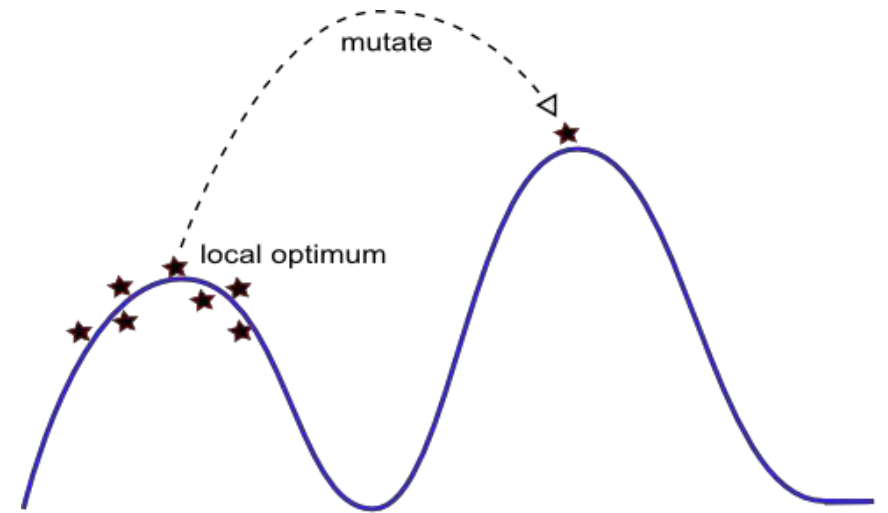
Remove

Repair

Penalize

Đột biến - Mutation

Đột biến (Mutation): tạo sự đa dạng di truyền, đột biến sẽ thay đổi nhỏ một cách ngẫu nhiên vào cá thể con, có nhiều phương thức đột biến khác nhau.



Toán tử Đột biến

Swap mutation

A B C D E F

Before mutation

A F C D B F

After mutation

Insertion mutation

A B C D E F

Before mutation

E F A D B C

After mutation

$0.01 < \text{Xác suất đột biến (Pm)} < 0.001$

Điều kiện kết thúc

- Điều kiện dừng là tiêu chí của thuật toán di truyền quyết định xem tiếp tục tìm kiếm hoặc ngừng tìm kiếm.
- Số thế hệ - Phương pháp chấm dứt dừng tiến trình phát triển khi số lượng tiến trình tối đa được người dùng xác định.
- Thời gian tiến hóa
- Ngưỡng thích nghi (Fitness Threshold)
- Sự hội tụ của quần thể (population)

Kết thúc vòng lặp, kỳ vọng để tìm một hoặc nhiều nhiễm sắc thể phù hợp.

GA TỔNG QUÁT

Bắt đầu

$t = 0;$

Khởi tạo $P(t)$

Tính độ thích nghi cho các cá thể thuộc $P(t)$;

Khi (điều kiện dừng chưa thỏa) lặp

$t = t + 1;$

Chọn lọc $P(t)$;

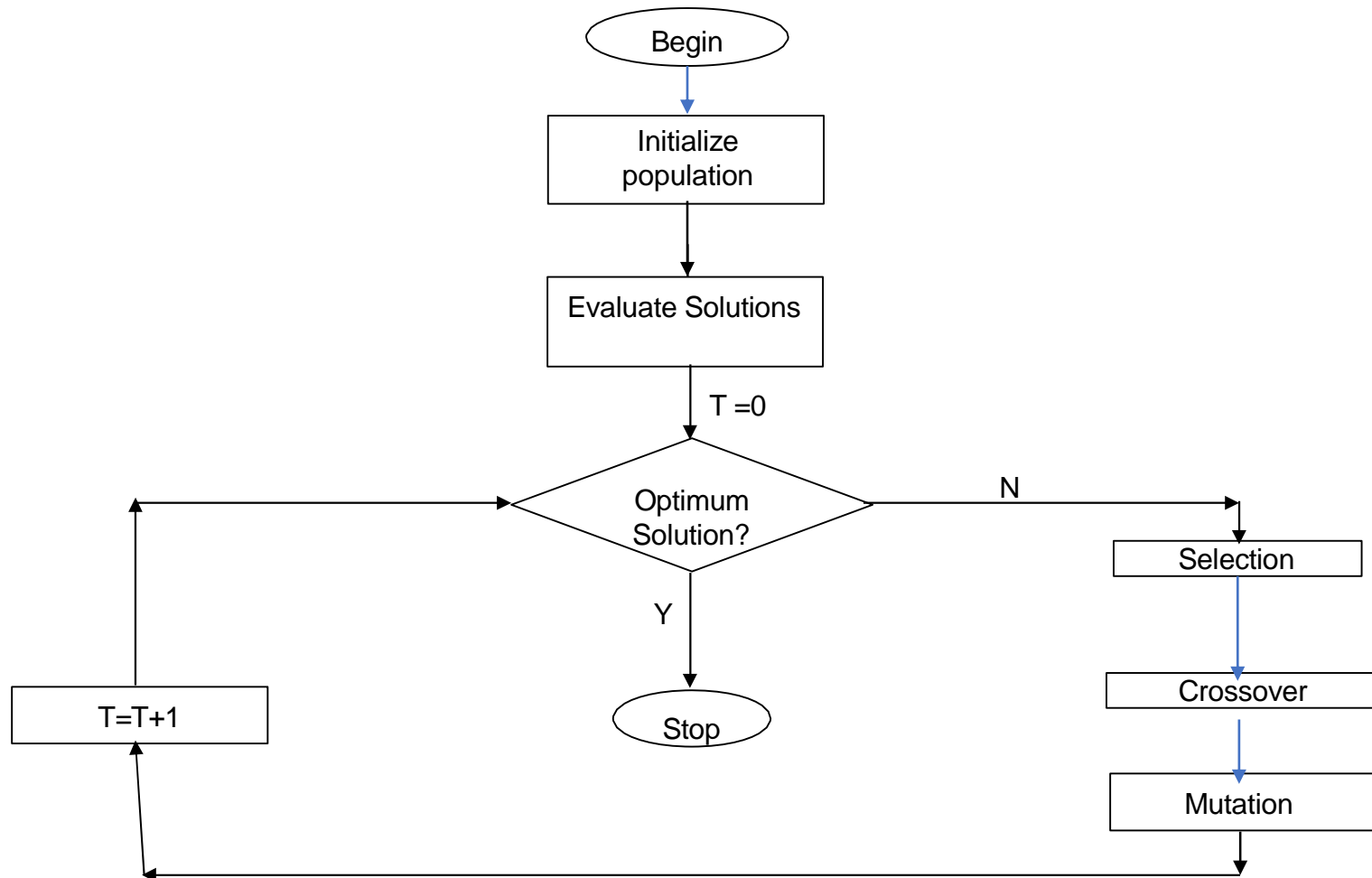
Lai $P(t)$;

Đột biến $P(t)$;

Hết lặp

Kết thúc

Lưu đồ GAs





BÀI TOÁN NGƯỜI BÁN HÀNG



Các bước thực hiện bài toán



BÀI TOÁN NGƯỜI BÁN HÀNG



3.3.1. Mã hóa bài toán

► Mã hóa đồ thị

Số hiệu đỉnh	Hoành độ	Tung độ
1	0	105
2	69	96
...

Tính đường thẳng:

$$d_{M,N} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

► Mã hóa chu trình (cá thể - gen)

C1	1	3	2	4	6	9	8	7	5	10
C2	10	7	9	6	4	2	8	3	1	5

Mỗi chu trình cần phải có thêm thông số về chi phí của toàn bộ chu trình đó. Chi phí này được tính bằng tổng độ dài tất cả các cạnh tạo nên chu trình đó



3.3.2. Khởi tạo quần thể

- ▶ Quần thể ban đầu được khởi tạo bằng cách sinh ngẫu nhiên các chu trình, số lượng chu trình khởi tạo là một nửa số kích thước cá thể tối đa. Việc sinh ngẫu nhiên sử dụng hàm đột biến. Số kích thước cá thể tối đa có thể tùy biến theo số đỉnh của đồ thị cần giải

3.3.3. Đánh giá độ thích nghi

- ▶ Việc đánh giá độ thích nghi nhằm mục đích chọn lựa ra những chu trình (cá thể) tốt nhất.
- ▶ Chu trình được xem tốt nhất khi tổng chi phí của chu trình là bé nhất trong một quần thể.
- ▶ Tổng chi phí của chu trình chính là tổng chiều dài của các cặp cạnh có trong chu trình.
- ▶ Độ dài của cạnh được tính bằng công thức:

$$d_{M,N} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

3 BÀI TOÁN NGƯỜI BÁN HÀNG

3.3.3. Lai ghép

C1	1	3	2	4	6	9	8	7	5	10
C2	10	7	9	6	4	2	8	3	1	5

Thực hiện lai ghép 1 điểm cắt với vị trí cắt là ngẫu nhiên :

- Cắt từ điểm p đến hết chu trình của C2 đưa vào chu trình mới, lấy một ví dụ $p = 5$:

CON	2	8	3	1	5					
-----	---	---	---	---	---	--	--	--	--	--

- Xét từ đầu đến cuối chu trình 1, nạp dần các điểm chưa có trong con lai theo thứ tự duyệt ta được chu trình mới :

CON	2	8	3	1	5	4	6	9	7	10
-----	---	---	---	---	---	---	---	---	---	----

- Tính lại chi phí cho chu trình mới sinh.

3.3.3. Lai ghép

► Cài đặt code

```
private Circle hybridize(Circle c1, Circle c2) {
    Circle child = new Circle(c1.getLength());
    Random rand = new Random();
    int p = rand.nextInt(child.length - 1);
    int i=0, j=0, k=0;

    for (i=p; i<child.length; i++)
    {
        child.vertex[j] = c2.vertex[i];
        j++;
    }

    for (i=0; i<child.length; i++)
    {
        for (j=0; j<child.length - p; j++)
        {
            if (c1.vertex[i] == child.vertex[j])
                break;
            else
            {
                if (j == child.length - p - 1)
                {
                    k++;
                    child.vertex[j+k] = c1.vertex[i];
                }
            }
        }
    }
    return child;
}
```

3 BÀI TOÁN NGƯỜI BÁN HÀNG



3.3.4. Đột biến

Phương thức đột biến được thực hiện dựa trên 1 cá thể đầu vào

C1	1	3	2	4	6	9	8	7	5	10
----	---	---	---	---	---	---	---	---	---	----

Thực hiện đột biến bằng trao đổi các điểm trên gen cho nhau. Số lần trao đổi được sinh ngẫu nhiên từ trong khoảng 5% chiều dài chu trình (tức là có tối đa 10% vị trí trên 1 gen có thể bị đột biến), vị trí điểm trao đổi cũng được sinh ngẫu nhiên trong quá trình chạy.

C3	1	9	2	10	6	3	8	7	5	4
----	---	---	---	----	---	---	---	---	---	---

3 BÀI TOÁN

3.3.4. Đột biến

► Cài đặt code

```
private Circle mutate(Circle c) {  
    int n = c.getLength();  
    Circle nextgen = new Circle(n);  
    nextgen.setCircle(n, c.getCircle());  
    Random rand = new Random();  
    int count = rand.nextInt(mutateCoefficient) + 1;  
    int p1, p2, temp;  
  
    while (count > 0)  
    {  
        p1 = rand.nextInt(n);  
        p2 = rand.nextInt(n);  
        temp = nextgen.vertex[p1];  
        nextgen.vertex[p1] = nextgen.vertex[p2];  
        nextgen.vertex[p2] = temp;  
        count--;  
    }  
    return nextgen;  
}
```

3 BÀI TOÁN NGƯỜI BÁN HÀNG



3.3.5. Chọn lọc tự nhiên

- ▶ Kích thước quần thể là cố định qua các thế hệ. Ở mỗi thế hệ ta lại có các cá thể mới sinh bằng lai ghép và đột biến do đó cần phải có sự chọn lọc để đảm bảo tính cân bằng của quần thể
- ▶ $[Số\ cá\ thể\ 1\ thế\ hệ] = [kích\ thước\ mặc\ định] + [số\ cá\ thể\ mới\ sinh]$
- ▶ Cách thức chọn lọc cá thể được đánh giá dựa trên chi phí của mỗi chu trình. Cá thể được chọn làm lời giải cuối cùng là cá thể có chi phí nhỏ nhất trong quần thể sau một số thế hệ tiến hóa.

3.3.5. Chọn lọc tự nhiên

► Giải pháp:

- Sắp xếp quần thể theo chi phí tăng dần.
- Lựa chọn ngẫu nhiên 1 chỉ số : a ($0 < a < 1$)
- Loại cá thể thứ a [*kích thước mặc định*], kém thích nghi nhất từ [*kích thước mặc định*] cá thể đứng đầu danh sách quần thể.
- Loại đến khi số cá thể còn lại bằng kích thước mặc định.

3 BÀI TOÁN NGƯỜI BÁN HÀNG

3.3.5

```
private void sortList() {
    int i=0, j=0, min;
    Circle temp;
    for (i=0; i<population.size() - 1; i++)
    {
        min = i;
        for (j=i+1; j<population.size(); j++)
        {
            if (population.get(j).cost < population.get(min).cost)
                min = j;
        }
        temp = population.get(i);
        population.add(i, population.get(min));
        population.remove(i+1);
        population.remove(min);
        population.add(min,temp);
    }
}
```

3 BÀI TOÁN NGƯỜI BÁN HÀNG



3.3.6. Tiến hóa

- ▶ Với quần thể khởi tạo ban đầu, chúng được tiến hóa một cách ngẫu nhiên và thích nghi với điều kiện chọn lọc, các cá thể kém thích nghi sẽ bị loại thải và cá thể tốt nhất được chọn làm lời giải.
- ▶ Việc tiến hóa diễn ra qua một số thế hệ, ở mỗi thế hệ ta thực hiện lai ghép và đột biến ngẫu nhiên trên toàn quần thể.
 - ▶ Lai ghép : ngẫu nhiên trong 50% số cá thể đứng đầu tiên của quần thể (Lựa chọn cha mẹ ngẫu nhiên).
 - ▶ Đột biến cho toàn bộ quần thể 100% (tuy điều này có vẻ trái tự nhiên nhưng việc tìm ra nguồn gen mới cần ưu tiên hơn hết).

3 BÀI TOÁN NGƯỜI BÁN HÀNG



3.4. Ví dụ minh họa: dữ liệu gồm 4 thành phố

Thành phố (đỉnh)	Hoành độ	Tung độ
1	10	10
2	5	15
3	30	10
4	20	15
5	10	15
6	30	15
7	20	5
8	5	5

3 BÀI TOÁN NGƯỜI BÁN HÀNG



Khởi tạo quần thể: 4 chu trình (1 nửa kích thước của số lượng thành phố (NST))

1

Thành phố (đỉnh)	Hoành độ	Tung độ
1	10	10
2	5	15
3	30	10
4	20	15
5	10	15
6	30	15
7	20	5
8	5	5

2

Thành phố (đỉnh)	Hoành độ	Tung độ
4	20	15
1	10	10
3	30	10
2	5	15
5	10	15
6	30	15
7	20	5
8	5	5

3

Thành phố (đỉnh)	Hoành độ	Tung độ
1	10	10
2	5	15
3	30	10
4	20	15
6	30	15
7	20	5
5	10	15
8	5	5

4

Thành phố (đỉnh)	Hoành độ	Tung độ
3	30	10
4	20	15
2	5	15
1	10	10
5	10	15
6	30	15
7	20	5
8	5	5

3 BÀI TOÁN NGƯỜI BÁN HÀNG



Đánh giá độ thích nghi của từng chu trình

- ▶ Chu trình 1 (1-2-3-4-5-6-7-8) = $5\sqrt{2} + 5\sqrt{26} + 5\sqrt{5} + 10 + 20 + 10\sqrt{2} + 15 = 102,8886$
- ▶ Chu trình 2 (4-1-3-2-5-6-7-8) = $5\sqrt{5} + 20 + 5\sqrt{26} + 5 + 20 + 10\sqrt{2} + 15 = 110,8176$
- ▶ Chu trình 3 (1-2-3-4-6-7-5-8) = $5\sqrt{2} + 5\sqrt{26} + 5\sqrt{5} + 10 + 10\sqrt{2} + 10\sqrt{2} + 5\sqrt{5} = 93,2111$
- ▶ Chu trình 4 (3-4-2-1-5-6-7-8) = $5\sqrt{5} + 15 + 5\sqrt{2} + 5 + 20 + 10\sqrt{2} + 15 = 87,3935$

Nhận xét: 2 chu trình có tổng giá trị nhỏ nhất là chu trình 3,4.

=> Ta tiếp tục bước lại ghép tiếp theo thông qua 2 chu trình này.

3 BÀI TOÁN NGƯỜI BÁN HÀNG



Tiến hành lại

3

Thành phố (đỉnh)	Hoành độ	Tung độ
1	10	10
2	5	15
3	30	10
4	20	15
6	30	15
7	20	5
5	10	15
8	5	5

4

Thành phố (đỉnh)	Hoành độ	Tung độ
3	30	10
4	20	15
2	5	15
1	10	10
5	10	15
6	30	15
7	20	5
8	5	5

CON

Thành phố (đỉnh)	Hoành độ	Tung độ
5	10	15
6	30	15
7	20	5
8	5	5
1	10	15
2	5	15
3	30	10
4	20	15

3 BÀI TOÁN NGƯỜI BÁN HÀNG



Đột biến con lai

CON LAI

Thành phố (đỉnh)	Hoành độ	Tung độ
5	10	15
6	30	15
7	20	5
8	5	5
1	10	15
2	5	15
3	30	10
4	20	15

CON LAI ĐỘT BIẾN

Thành phố (đỉnh)	Hoành độ	Tung độ
5	10	15
6	30	15
2	5	15
8	5	5
1	10	15
7	20	5
3	30	10
4	20	15

$$20 + 25 + 10 + 5\sqrt{5} + 10\sqrt{2} + 5\sqrt{5} + 5\sqrt{5} = 102,6831$$

Con lai đột biến sẽ bị loại bỏ khỏi quần thể bởi độ thích nghi của nó kém hơn với cha mẹ

4 KẾT LUẬN



► Ưu điểm :

Giải thuật GA lập luận mang tính chất ngẫu nhiên để tìm giải pháp tối ưu cho những vấn đề phức tạp, thay vì xác định như toán học giải tích. Chính hàm số thích nghi là giúp GA tìm giải pháp tối ưu trong rất nhiều giải pháp có thể có

Đối với các bài toán nhỏ số đỉnh dưới 100 thì thuật toán luôn đưa ra những kết quả tối ưu và với các bài toán vừa với số đỉnh dưới 1000 thì thuật toán đưa ra kết quả tiệm cận

► Nhược điểm

Vấn đề lớn nhất là đối với những bài toán với số đỉnh lớn hơn 1000 giải thuật vẫn chưa tìm ra được lời giải như mong đợi

Ngoài ra việc giải bài toán lớn còn mất rất nhiều thời gian, công sức mà chưa chắc đã đưa ra kết quả đúng

Giải thuật di truyền đã đề xuất phần lớn vẫn phụ thuộc vào sự “may mắn” để tìm ra

Tuy nhiên trong tương lai không xa khi mà các cải tiến về giải thuật GA như quần thể động, đa quần thể tương tác được hoàn thiện thì sẽ giúp ta tìm ra được lời giải chính xác hơn với những bài toán lớn với số đỉnh trên 1000

8-queens Problem

8-queens: (a) Initialization

Khởi tạo quần thể với 4 cá thể:

$$v_1 = (24748552)$$

$$v_2 = (32752411)$$

$$v_3 = (24415124)$$

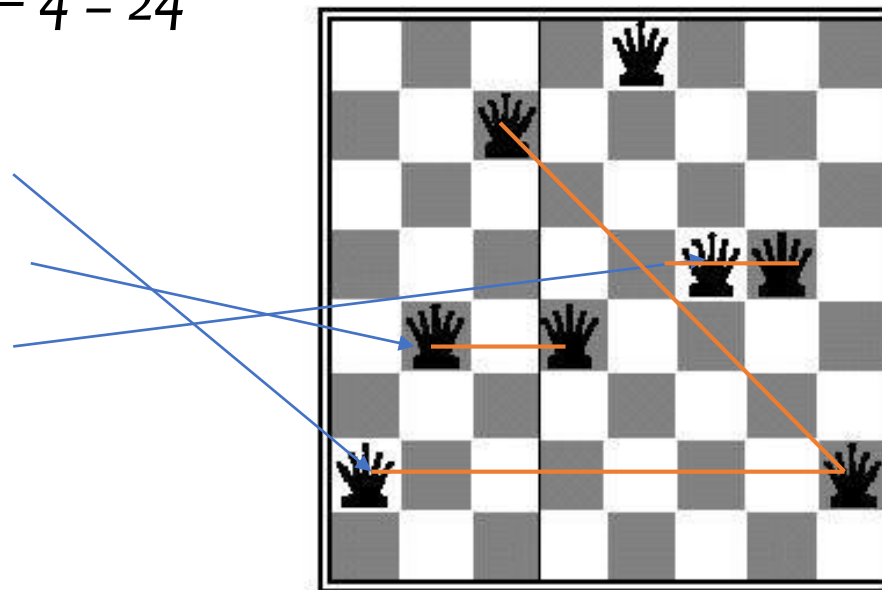
$$v_4 = (32543213)$$

8-queens: (b) Tính toán thích nghi (Fitness function)

- Fitness function: càng ít xung đột (khả năng hậu ăn nhau) càng tốt
- Chúng ta có thể sử dụng số lượng các cặp hậu không ăn nhau. Giá trị cao nhất có thể của Fitness function là $c_2^8 8 = 28$. Mỗi lời giải pháp có một giá trị =28.

8-queens: (b) Tính toán thích nghi (Fitness function)

- Độ thích nghi (*fitness*) của v_1 (24748552) là $28 - 4 = 24$
- Bởi vì có 4 trường hợp hậu ăn nhau:
 - The queens on 1st and 8th column
 - The queens on 2nd and 4th column
 - The queens on 6th and 7th column
 - The queens on 3rd and 8th column



8-queens: (b) Tính toán thích nghi (Fitness function)

- Các giá trị fitness được tính như sau:

$$\text{eval}(v_1) = 24$$

$$\text{eval}(v_2) = 23$$

$$\text{eval}(v_3) = 20$$

$$\text{eval}(v_4) = 11$$

$$v1 = (24748552)$$

$$v2 = (32752411)$$

$$v3 = (24415124)$$

$$v4 = (32543213)$$

- Không có nhiễm sắc thể nào là lời giải. Nếu một lời giải pháp được tìm thấy, thuật toán kết thúc

8-queens: (c) Selection – Chọn lọc

- Tổng fitness values = $24 + 23 + 20 + 11 = 78 \rightarrow$ xác suất của mỗi nhiễm sắc thể được lựa chọn cho thế hệ kế tiếp là :

$$prob(v_1) = 24/78 = 31\%$$

$$prob(v_2) = 23/78 = 29\%$$

$$prob(v_3) = 20/78 = 26\%$$

$$prob(v_4) = 11/78 = 14\%$$

8-queens: (c) Selection – Chọn lọc

- Tiếp theo, sắp xếp các xác suất khác nhau từ 0 đến 1 cho việc xử lý quay bánh xe roulette :

v_1 : 0.00 to 0.31

v_2 : 0.31 to 0.60

v_3 : 0.60 to 0.84

v_4 : 0.84 to 1.00

8-queens: (c) Selection – Chọn lọc

- Bốn số ngẫu nhiên được chọn cho thế hệ kế tiếp, các số ngẫu nhiên sau:

0.4012

0.1486

0.5973

0.8129

$v_1 = (24748552)$

$v_2 = (32752411)$

$v_3 = (24415124)$

$v_4 = (32543213)$

- Các cá thể được chọn:

$0.4012 \rightarrow v_2 (32752411) \rightarrow v_1'$

$0.1486 \rightarrow v_1 (24748552) \rightarrow v_2'$

$0.5973 \rightarrow v_2 (32752411) \rightarrow v_3'$

$0.8129 \rightarrow v_3 (24415124) \rightarrow v_4'$

8-queens: (d) Lai ghép - Crossover

- ❑ Bước tiếp theo, các nhiễm sắc thể sẽ thực hiện Lai ghép (crossover). Giả sử xác suất crossover là 0.80. Tất cả 4 nhiễm sắc thể được chọn để crossover (được làm tròn thành một số chẵn).
- ❑ Các nhiễm sắc thể đã chọn được ghép nối ngẫu nhiên.
- ❑ Điểm crossover được lựa chọn ngẫu nhiên cho mỗi lai ghép.
 - Giả sử chữ số thứ ba trong cặp đầu tiên được chọn làm điểm lai ghép.

$$v_1' = (327 \mid 52411)$$

$$v_2' = (247 \mid 48552)$$

- Sau khi lai ghép, chúng ta có:

$$v_1'' = (327 \mid 48522)$$

$$v_2'' = (247 \mid 52411)$$

8-queens: (d) Lai ghép - Crossover

- Giả sử chữ số thứ ba trong cặp đầu tiên được chọn làm điểm lai ghép.

$$v_1' = (327 \mid 52411)$$

$$v_2' = (247 \mid 48552)$$

- Sau khi lai ghép, chúng ta có:

$$v_1'' = (327 \mid 48522)$$

$$v_2'' = (247 \mid 52411)$$

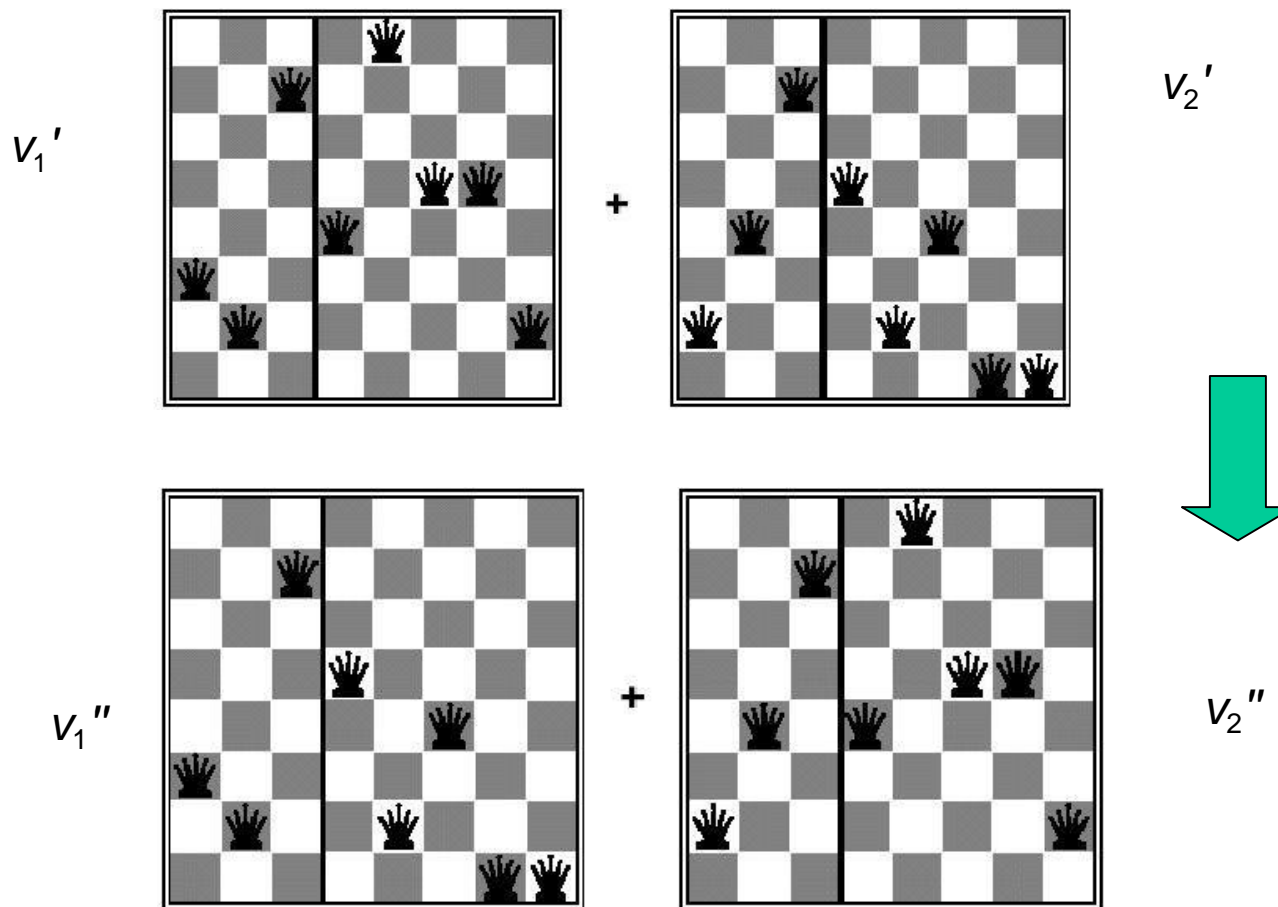
8-queens: (d) Lai ghép - Crossover

$$v_1' = (327 \mid 52411)$$

$$v_2' = (247 \mid 48552)$$

$$v_1'' = (327 \mid 48522)$$

$$v_2'' = (247 \mid 52411)$$



8-queens: (e) Mutation – Đột biến

- ✓ Các gen (digit) sẽ bị biến đổi.
- ✓ Chọn một hậu ngẫu nhiên và di chuyển nó đến một vị trí ngẫu nhiên cột của bàn cờ.
- ✓ Giả sử xác suất đột biến là 0,05
- ✓ Giả sử số ngẫu nhiên thứ 6, số 19 và số 32 nhỏ hơn 0,05.
- ✓ Ba gen tương ứng sẽ bị đột biến là:
 - 6th digit in v_1''
 - 3rd digit in v_3''
 - 8th digit in v_4''

8-queens: (e) Mutation – Đột biến

- Đối với mỗi digit bị đột biến, một số ngẫu nhiên sẽ được tạo ra để xác định vị trí của hậu di chuyển đến, ví dụ: $v_1'' = (32748\textcolor{red}{5}22)$
- Nếu một số ngẫu nhiên xác định gen được đột biến đổi thành 1, nhiễm sắc thể mới sẽ là: $v_1''' = (32748\textcolor{red}{1}22)$

8-queens: (e) Mutation – Đột biến

- Quá trình tương tự được áp dụng cho việc đột biến gen
- Các gen cuối cùng cho thể hệ mới như sau:
 $v_1''' = (32748122)$
 $v_2''' = (24752411)$
 $v_3''' = (32252124)$
 $v_4''' = (24415417)$
- Quá trình này được lặp lại từ bước (b) cho đến khi tìm thấy lời giải.

8-queens: A Summary

