Simple GA

1. Start with a randomly generated population
2. Calculate the fitness of each chromosome in the population
3. Repeat the following steps until n offspring have been created
    a. Select a pair of parent chromosomes from the current population
    b. With probability pc crossover the pair at a randomly chosen point to from two offspring
    c. Mutate the two offspring at each locus with probability pm
4. Replace the current population with the new population
5. Go to step 2

Steps in each iteration

1. **Selection**
    a. Selects individual for reproduction
    b. Selection is done randomly
    c. Considers probability depending on the relative fitness of the individual
    d. Best ones are often chosen for reproduction rather than poor ones
2. **Reproduction**
    a. Offspring are bred by selected individuals
    b. To generate new chromosomes, algorithm can use both recombination and mutation
3. **Evaluation**
    a. Fitness of new chromosomes is evaluated
4. **Replacement**
    a. Individuals from the old populations are killed
    b. Replaced by new ones

➢ Algorithm is stopped when population converges towards optimal solution

Simple GA

```
BEGIN
  Generate initial population;
  Compute fitness of each individual;
  WHILE NOT finished DO LOOP
    BEGIN
      Select individuals from old generations
        For mating;
      Create offspring by applying recombination and/or mutation to the selected
      individuals;
      Compute fitness of new individuals;
       Kill old individuals to make room for new chromosomes and
         insert offspring in the new generalization;
       IF Population has converged
         THEN finishes: =TRUE;
      END
 END
```

Eduklash Just Another Way To Learn

General Genetic Algorithm

Step 1: Create a random initial state

- An initial population is created from a random selection of solutions
- In symbolic AI systems, initial state in a problem is already given

Step 2: Evaluate fitness

- A value for fitness is assigned to each solution
- It depends on how close it is to solving problem

Step 3: Reproduce

- Those chromosomes with a higher fitness value are more likely to reproduce offspring
- Offspring is a product of the father and mother

Step 4: Next generation

- If new generation contains a solution that produces desired output then problem is solved
- Else new generation will go through same process

Eduklash
Just Another Way To Learn

Operators in GA

Encoding

- It is a process of representing individual genes
- It can be performed using
  - Bits
  - Numbers
  - Trees
  - Arrays
  - Lists

Representation

Chromosomes could be:

- **Bit strings** (0101 ... 1100)
- **Real numbers** (43.2 -33.1 ... 0.0 89.2)
- **Permutations of element** (E11 E3 E7 ... E1 E15)
- **Lists of rules** (R1 R2 R3 ... R22 R23)
- **Program elements** (genetic programming)
- **... any data structure ...**

Binary Encoding

- Each chromosome encodes a binary string
- Each bit in the string can represent some characteristics of solution
- It is not natural for many problems
- Sometimes corrections must be made after generic operation is completed

Octal Encoding

- Uses string made up of octal numbers(0-7)

Hexadecimal Encoding

- It uses string made up of hexadecimal numbers(0-9, A-F)

Permutation Encoding

- Every chromosome is a string of numbers, represented in a sequence
- It is only useful for ordering problems
- Sometimes corrections have to be done after generic operation is complete

Value Encoding

- Every chromosome is a string of values
- Values can be anything connected to the problem
- It produces best results for some special problems
- It is often necessary to develop new genetic operators specific to the problem
- Direct value encoding can be used in problems, where some complicated values, such as real numbers are used

Tree Encoding

- It is mostly used for genetic programming
- Every chromosome is a tree of some objects such as functions and commands of a programming language

Just Another Way To Learn

Binary Encoding

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 1 0 0

Octal encoding

Chromosome 1	03467216
Chromosome 2	15723314

Hexadecimal Encoding

Chromosome 1	9CE7
Chromosome 2	3DBA

Educlash
Just Another Way To Learn

Permutation Encoding

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

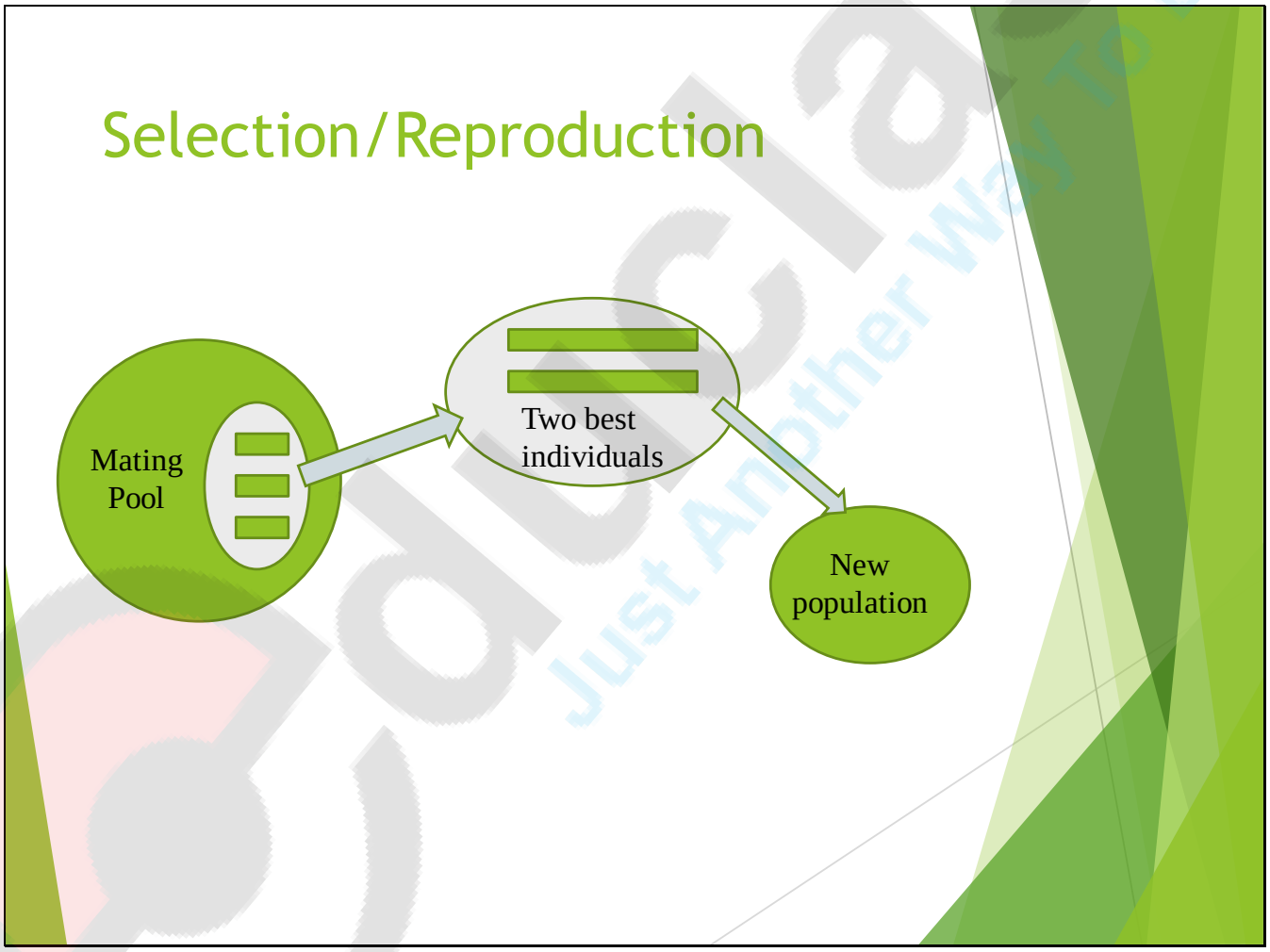
Value Encoding

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

Edulash
Just Another Way To Learn

Selection/Reproduction

- It is a process of choosing two parents from population for crossing
- Two types
  - **Proportionate-based selection**- selection based upon their fitness values relative to the fitness of other individuals in the population
  - **Ordinal-based selection**- selection based upon their rank within population
- The purpose of selection is to emphasize fitter individuals in the population in hopes that their offspring have higher fitness
- Chromosomes are selected from population to be parents for reproduction



Edclash
Just Another Way To Learn

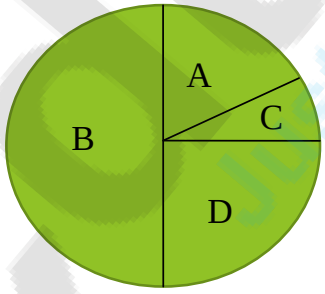
Roulette Wheel selection

- It is a linear search through a Roulette wheel
- Slots in the wheel weighted in the proportion to the individual's fitness values
- Target value is set
- Target value is a random proportion of sum of the fitnesses in the population
- Population is stepped through until target value is reached

Example

Fitness function= number of 1 bits in chromosome

Chromosome	Fitness
A: 00000110	2
B: 11101110	6
C: 00100000	1
D: 00110100	3



Random Selection

- It randomly selects a parent from the population
- It is little more disruptive, on average, than Roulette wheel selection

Rank Selection

- It ranks the population
- Every chromosome receives fitness from the ranking
- The worst has fitness 1
- Best has fitness N
- It preserves diversity and hence is a successful search
- It can be achieved in two ways as
  1.
     - Select a pair of individuals at random
     - Generate random number R between 0, 1
     - r is a parameter
     - If R<r first individual is a parent
     - If R>r second individual is a parent
     - Repeat to select second parent
  2.
     - Select to individual at random
     - One with highest evaluation becomes parent
     - Repeat to find second parent

Tournament Selection

- Provides selective pressure by holding a tournament competition among N individuals
- Winner is one with highest fitness
- Tournament and winner are inserted into mating pool
- Competition is repeated until mating pool for generating new offspring is filled
- The mating pool comprising the tournament winner has higher average population fitness
- The fitness difference provides the selection pressure
- This method is more efficient and leads to an optimal solution

Boltzmann Selection

- A continuously varying temperature controls rate of selection according to preset schedule
- The temperature starts out high, which means that the selection pressure is low
- Temperature is gradually lowered, which gradually increases the selection pressure
- Probability based
- Probability that best string is selected and introduced into mating pool is very high
- Few best chromosomes are copied to new population

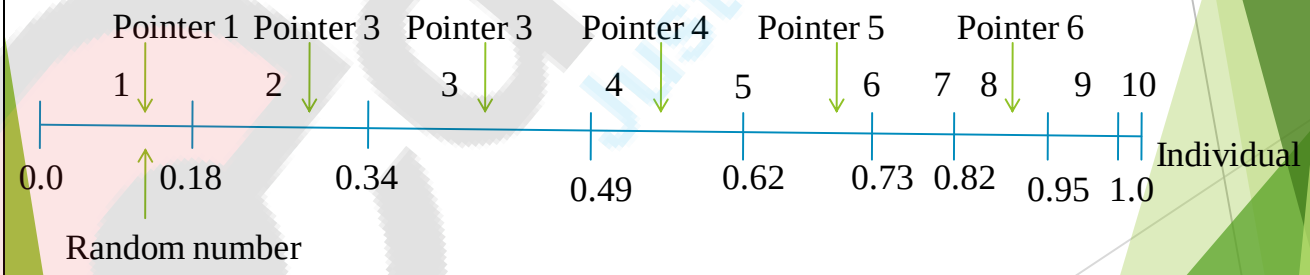
Educlash
Just Another Way To Learn

- Remaining individuals can be lost if they are not selected to reproduce or if crossover or mutation destroys them
- It significantly improves GA's performance

- **Stochastic Universal sampling**
  - It provides 0 bias
  - Individuals are mapped to contiguous segments of a line
  - Each individual's segment is equal in size to its fitness
  - Equally spaced pointers are placed over line
  - Number of pointers same as number of individuals to be selected(N)
  - Distance between two pointers are 1/N Pointer
  - Position of first pointer is given by random number in the range [0, 1/N Pointer]

Stochastic Universal sampling-Example

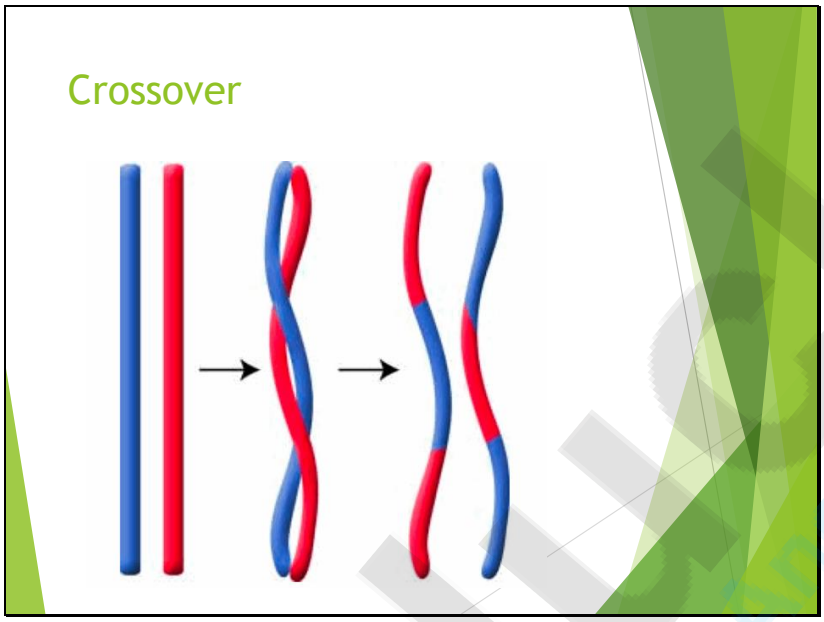
- For 6 individuals to be selected, the distance between the pointers is 1/6=0.167
- Select any Random number in the range [0, 0.167] as first position of pointer ⟶ 0.1



- After selection the mating population consists of the individuals : 1,2,3,4,6,8

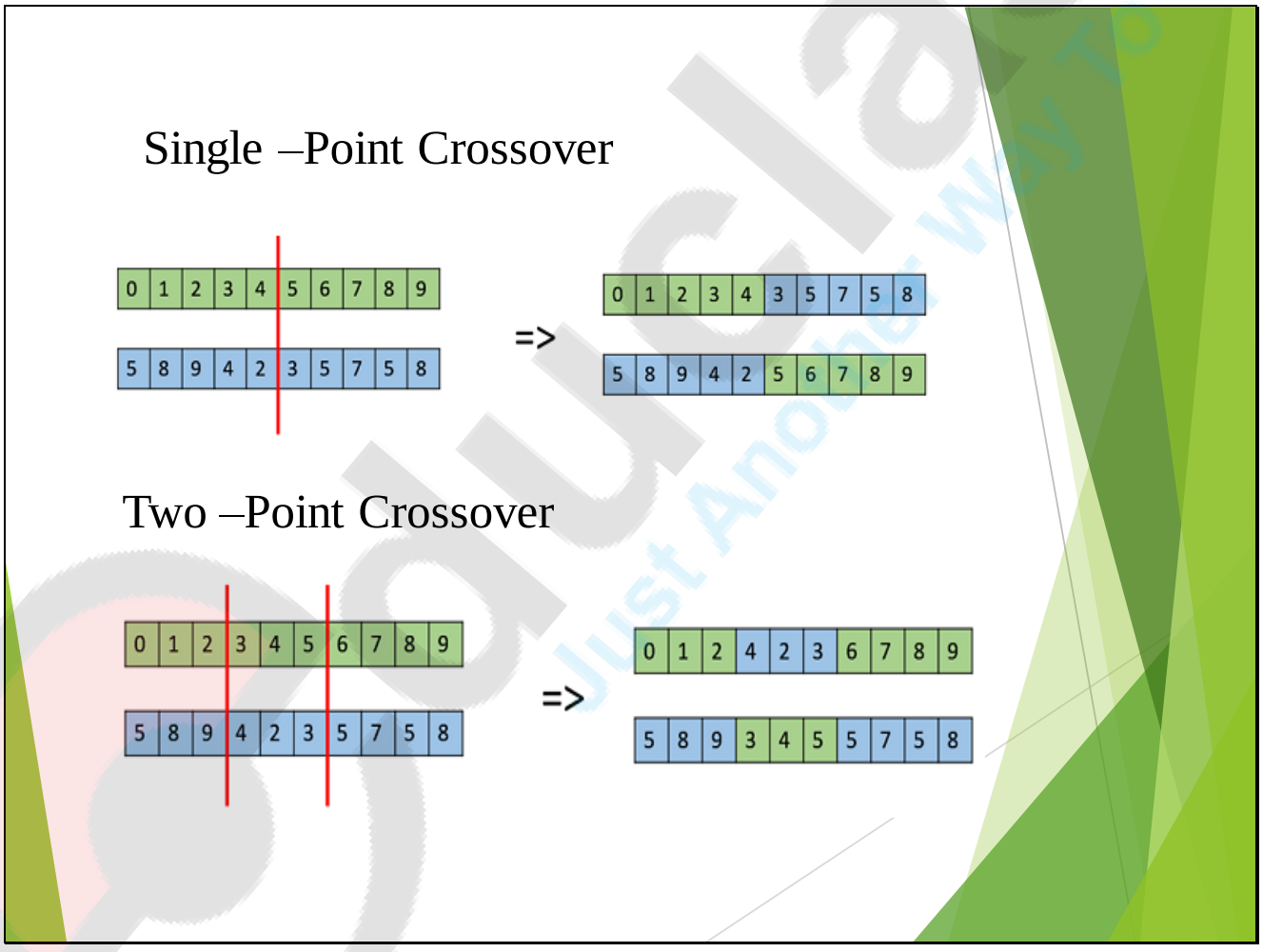
Crossover

- It is a process of taking 2 parents and producing child from them
- Steps
  - Reproduction operator selects at random a pair of 2 parents
  - A cross site is selected
  - Position values are swapped between two strings

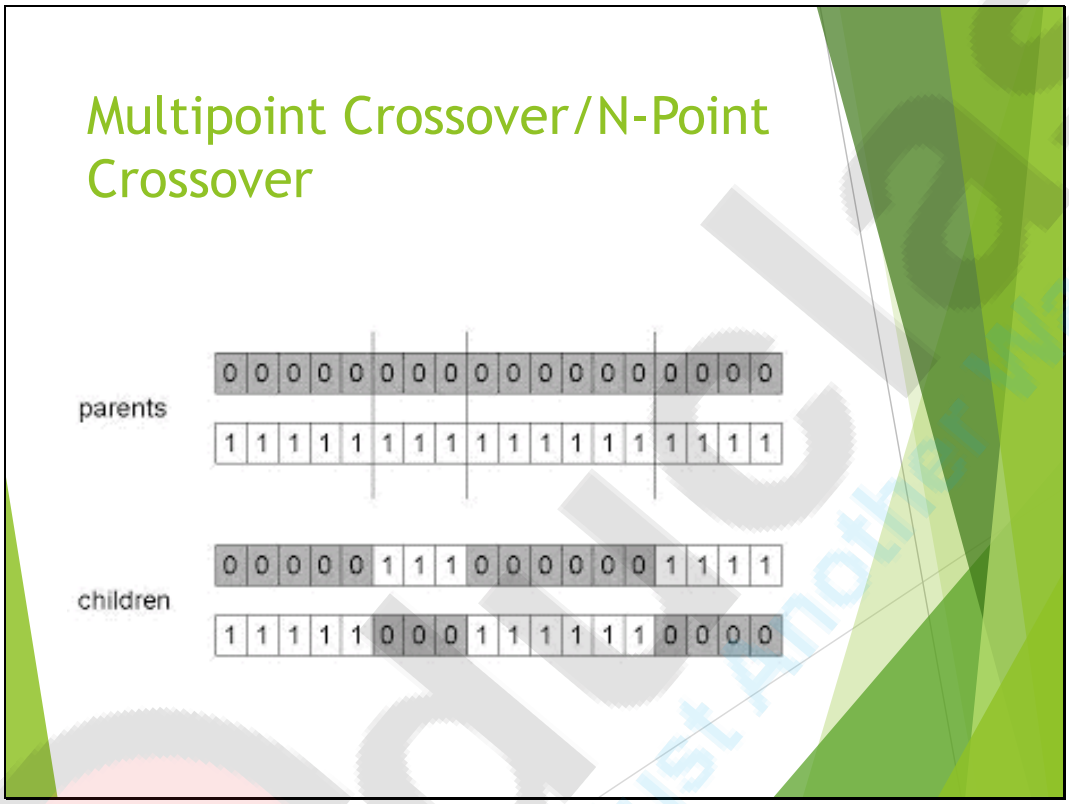


eduCash
Just Another Way To Learn

- **Single -Point Crossover**
  - Traditional GA uses single-point crossover
  - 2 mating chromosomes are cut once at corresponding points
  - Sections after cuts are exchanged
  - Crossover point can be chosen randomly
- **Two -Point Crossover**
  - It chooses 2 crossover points
  - Contents between these 2 points are exchanged

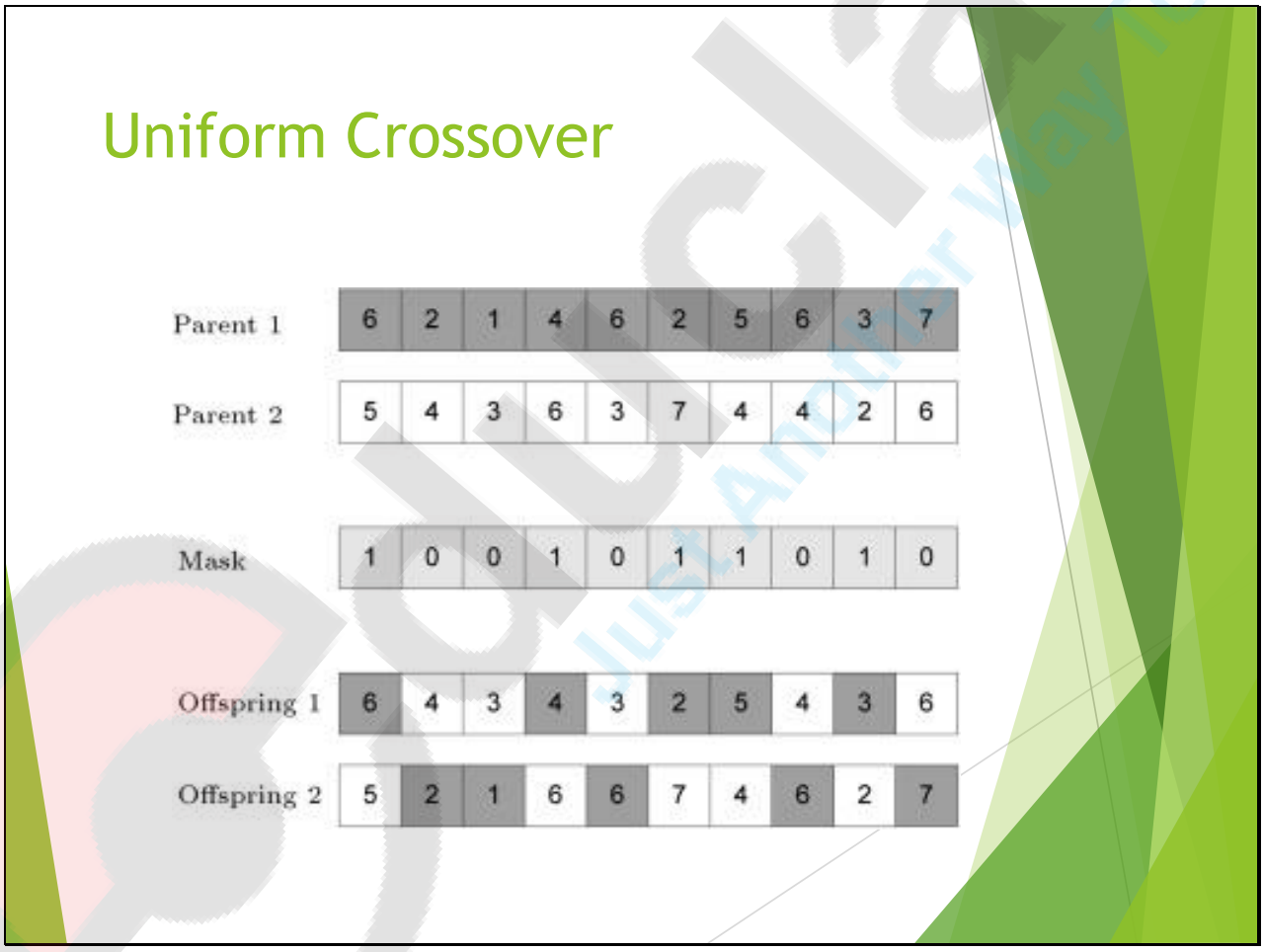


- Multipoint Crossover/N-Point Crossover
  - Choose n random crossover points
  - Split along those points
  - Swap alternating segments to get new off-springs
  - Generalization of 1 point (still some positional bias)

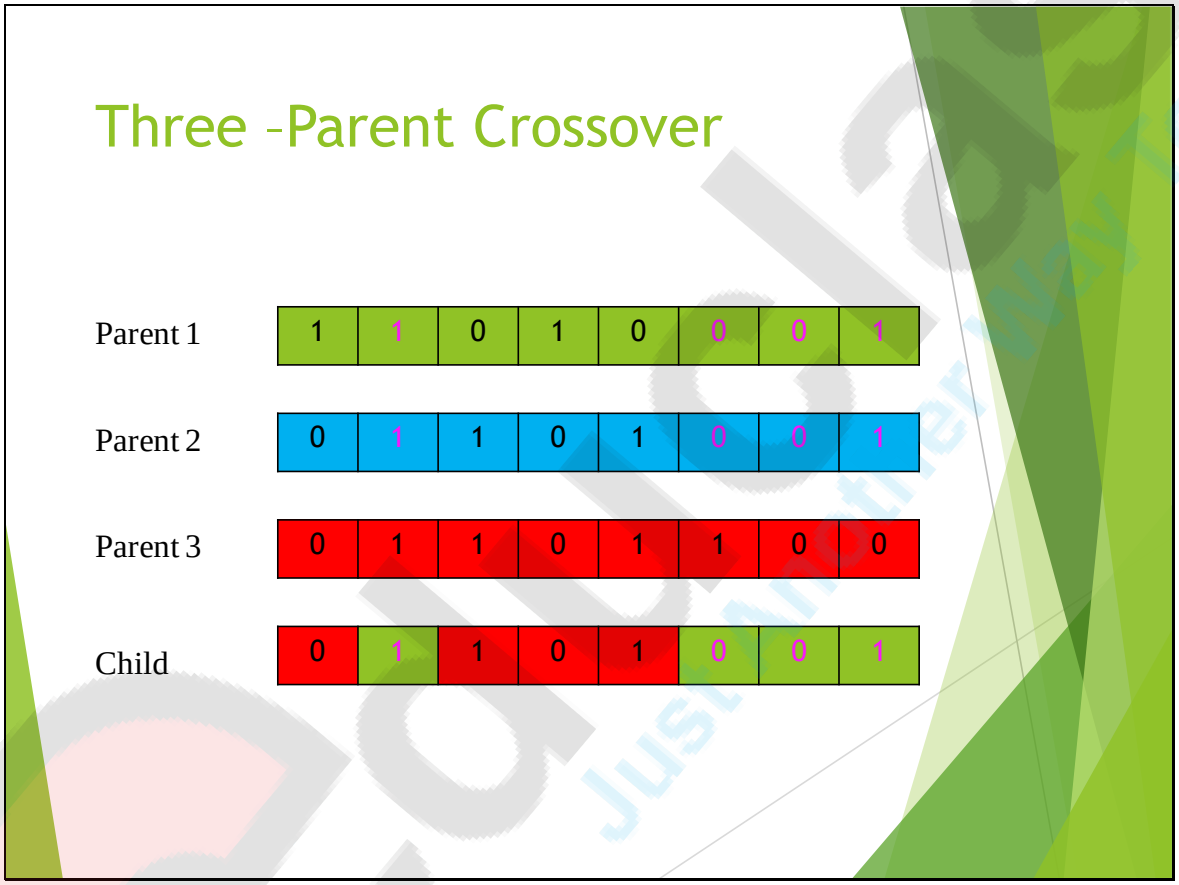


Uniform Crossover

- For each pair of parents, Binary Crossover mask is generated randomly
- Length of mask is same as parent chromosomes
- For every 1 in the crossover mask, gene is copied from first parent
- For every 0 in the crossover mask, gene is copied from the second parent
- Make an inverse copy of the gene for the second child



- **Three -Parent Crossover**
  - Three parents are randomly chosen
  - Each bit of first parent is compared with bit of second parent
  - If both are same, bit is taken for offspring
  - Else bit from third parent is taken for the offspring

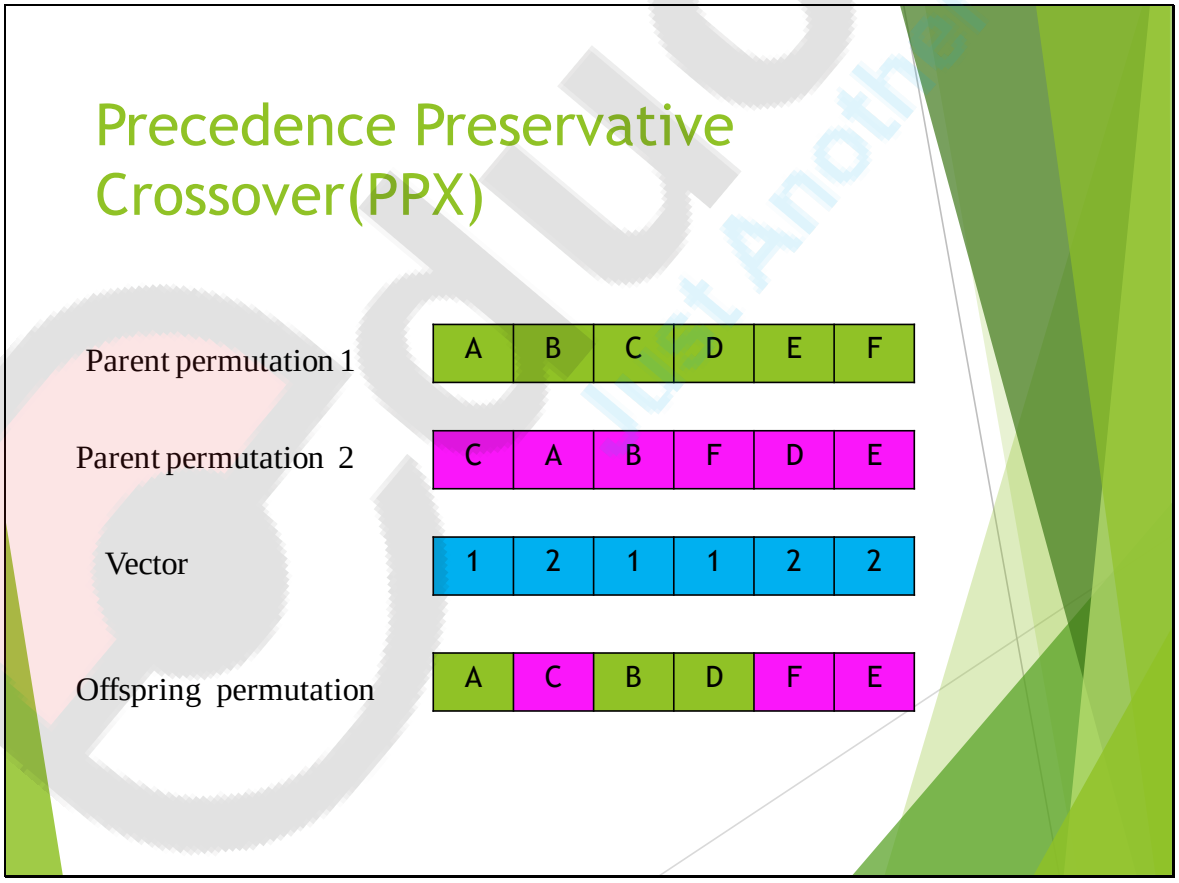


- **Crossover with Reduced Surrogate**
  - It restricts the location of crossover points
  - Crossover points only occur where gene values differ
- **Shuffle Crossover**
  - It is related to uniform crossover
  - A single crossover position is selected
  - Variables are randomly shuffled in both the parents before they are swapped

- Variables in offspring are unshuffled
- This removes positional bias

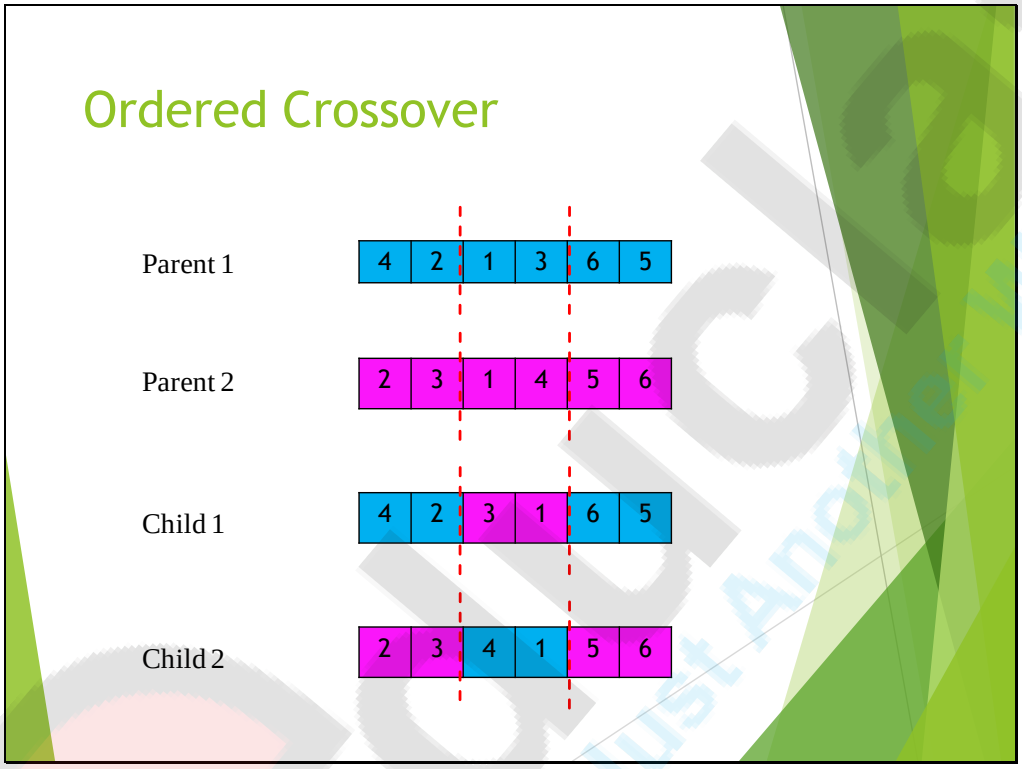
Precedence Preservative Crossover(PPX)

- It was developed for vehicle routing problems by Blanton and Wainwright(1993)
- Precedence relations of operations are given in 2 parental permutations
- Operator passes on theses relations to one offspring at the same rate
- No new precedence relations are introduced
- The operator works as follows:
  1. A vector representing the number of operations involved in problem is randomly filled with elements of the set {1,2}
  2. This vector defines the order in which operations are successively drawn from parent 1 and parent 2
  3. Parent and offspring permutations can be considered as lists
  4. Initialize an empty offspring
  5. Leftmost operation in one of 2 parents is selected in accordance with the order of parents given in vector
  6. After operation is selected, it is deleted in both parents
  7. Selected operation is appended to the offspring
  8. Step 7 is repeated until both parents are empty and offspring is filled



Ordered Crossover

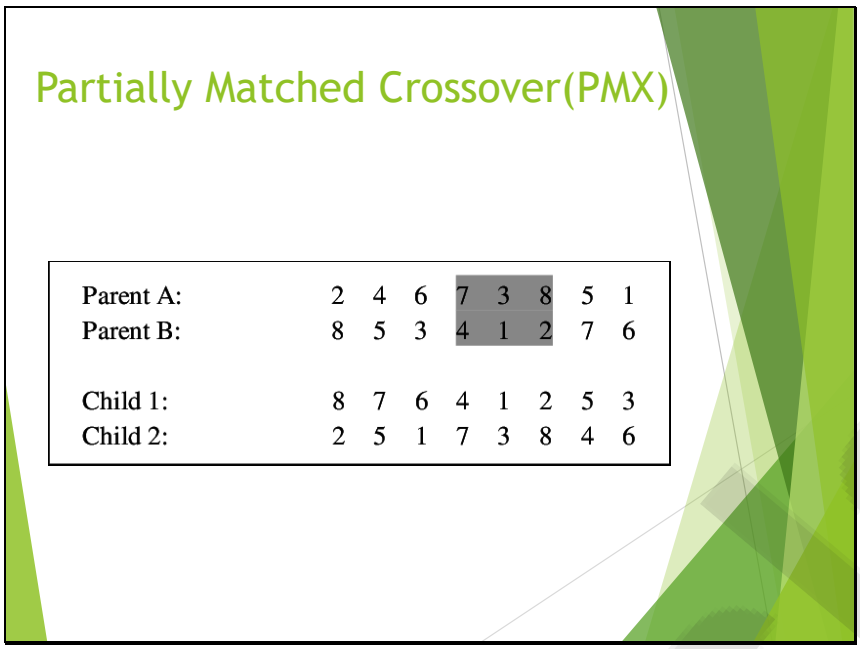
- It is used when problem is order based
- 2 random crossover points are selected
- These points partition parent chromosomes into left, middle and right portions
- Child inherits its left and right section from parent 1
- Its middle section is determined by genes in middle section of parent 1 in the order in which values appear in parent 2



Partially Matched Crossover(PMX)

- It is applied in TSP
- TSP chromosomes are simply sequences of integers
- Each integer represents different city
- Order represents time at which a city visited
- PMX proceeds as follows
- 2 chromosomes are aligned
- 2 crossing sites are selected uniformly at random
- Matching section is used to effect a cross through position by position exchange operation

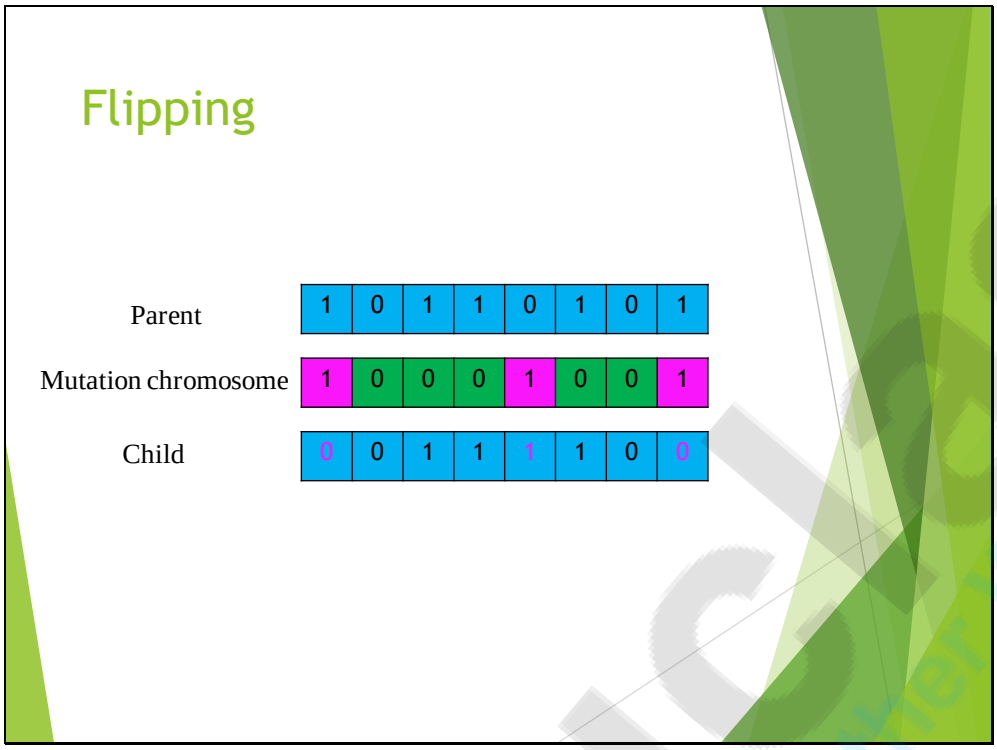
- Alleles are moved to their new positions in offspring



- **Crossover Probability**
  - It is a parameter that describes how often crossover will be performed
  - If there is a crossover, offspring are made from parts of both parents' chromosome
  - If pc is 100%, all offspring are made by crossover
  - If pc is 0%, all offspring are made from exact copies of chromosomes from old population

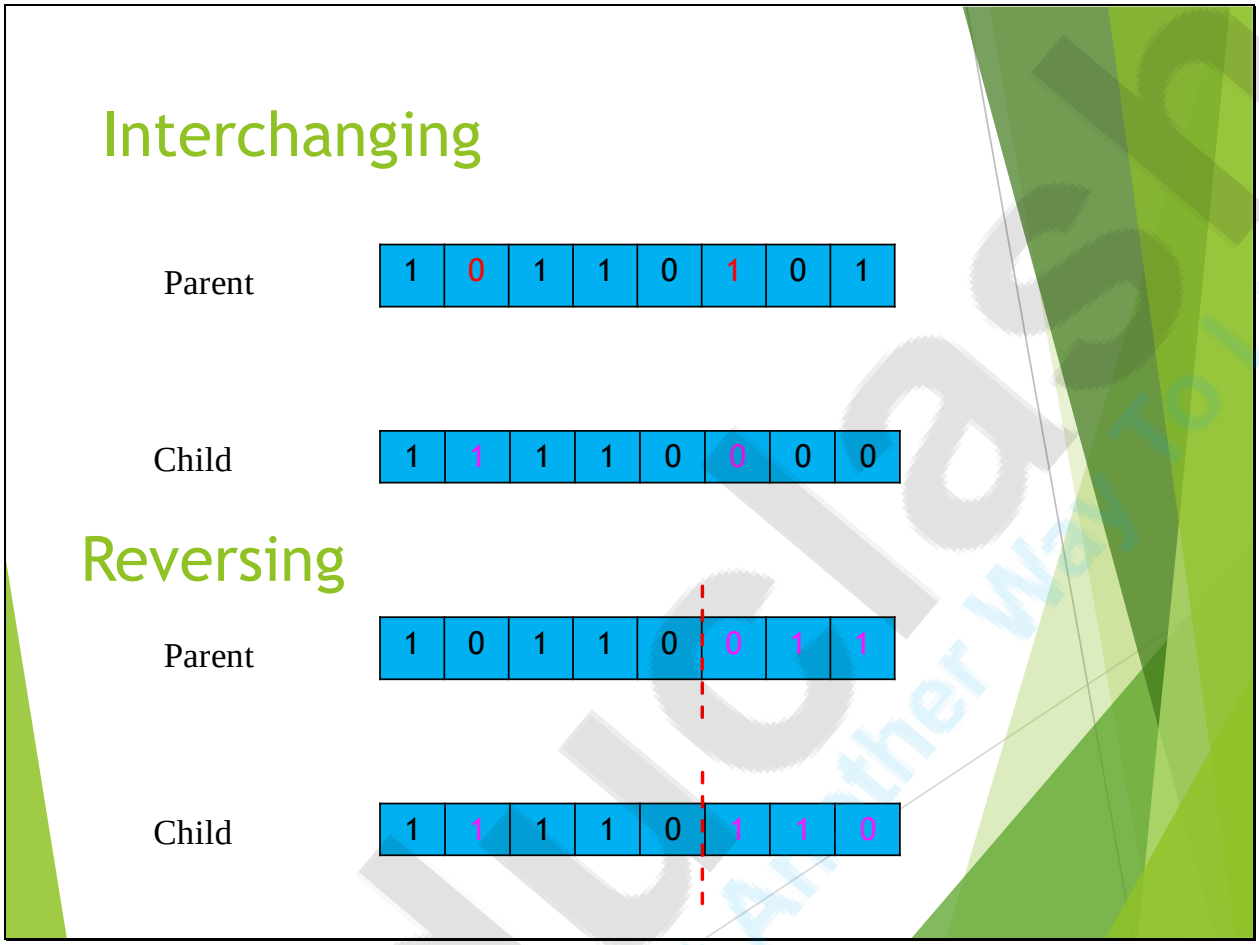
- **Mutation**
  - Mutation is performed after crossover
  - It recovers the lost genetic material
  - It is an insurance policy against loss of genetic material
  - Background operator to maintain genetic diversity in the population
  - Different mutation forms for different representation
- **Mutation of bit**
  1. Flipping
  2. Interchanging
  3. Reversing

- **Flipping**
  - It changes 0 to 1 and 1 to 0 based on a mutation chromosome generated



- **Interchanging**
  - Two random positions of string are chosen
  - Bits corresponding to those positions are interchanged
- **Reversing**
  - A random position is chosen
  - Bits next to that position are reversed
- **Mutation Probability**
  - Pm decides how often parts of chromosome will be mutated
  - If Pm is 100%, whole chromosome is changed
  - If Pm is 0%, nothing is changed
  - Mutation should not occur very often as GA will change to random search

Interchanging



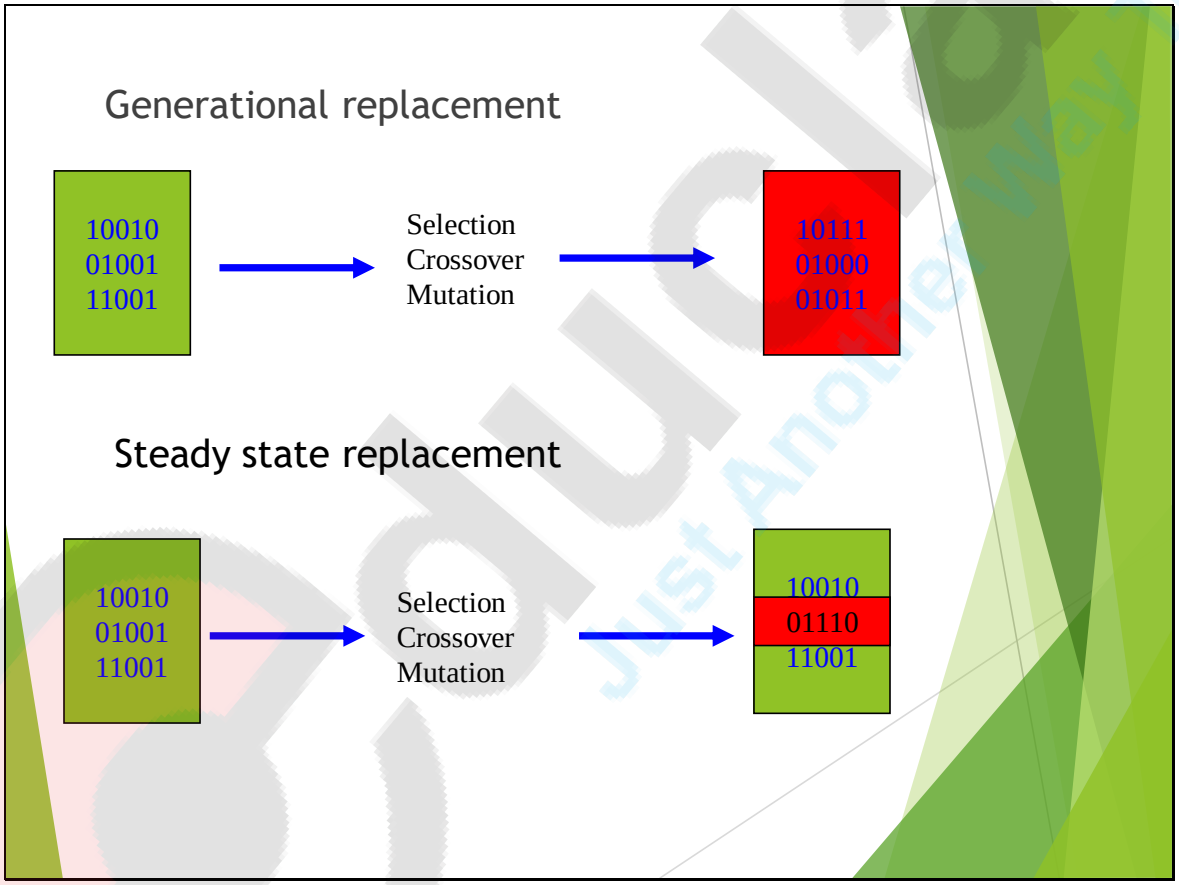
Replacement Schemes

Generational replacement

- Entire population is replaced each generation
- Non-overlapping population

Steady state replacement

- Single individual (worst, random) is replaced by
- One offspring
- Overlapping population



Example

- Maximize function

$$f(x) = x^2$$

where x is permitted to vary between 0 and 31

eduCash
Just Another Way To Learn

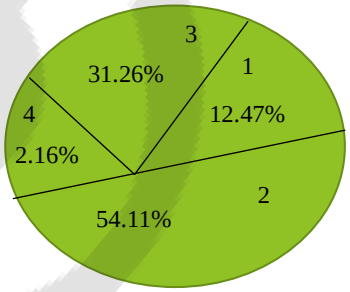
Example

- Initial population is of size 4 (randomly chosen)
- Evaluate fitness function i.e. $f(x) = x^2$

String no.	Initial population	x	fitness	Prob	Percentage prob	Expected count	Actual count
1	01100	12	144	0.1247	12.47	0.4987	1
2	11001	25	625	0.5411	54.11	2.1645	2
3	00101	5	25	0.0216	2.16	0.0866	0
4	10011	19	361	0.3126	31.26	1.2502	1
Sum			1155	1	100	4	4
Avg.			288.75	0.2500	25	1	1
Max.			625	0.5411	54.11	2.1645	2

Example

- Selection using Roulette wheel



Example

- Crossover

String no.	Mating pool	Cross over point	Offspring after crossover	X	Fitness value
1 2 3 4	01100 11001 11001 10011	4 4 2 2	01101 11000 11011 10001	13 24 27 17	169 576 729 289
Sum Avg. Max.					1763 440.75 729

eduCBA
Just Another Way To Learn

Example

- Mutation

String no.	Offspring after crossover	Mutation chromosomes For flipping	Offspring after mutation	X	Fitness value
1 2 3 4	01101 11000 11011 10001	10000 00000 00000 00100	11101 11000 11011 10100	29 24 27 20	841 576 729 400
Sum Avg. Max.					2546 636.5 841

- From the tables, it can be observed that after mutation at random, a new offspring (11101) is produced which is an excellent choice

Edureka! Just Another Way To Learn