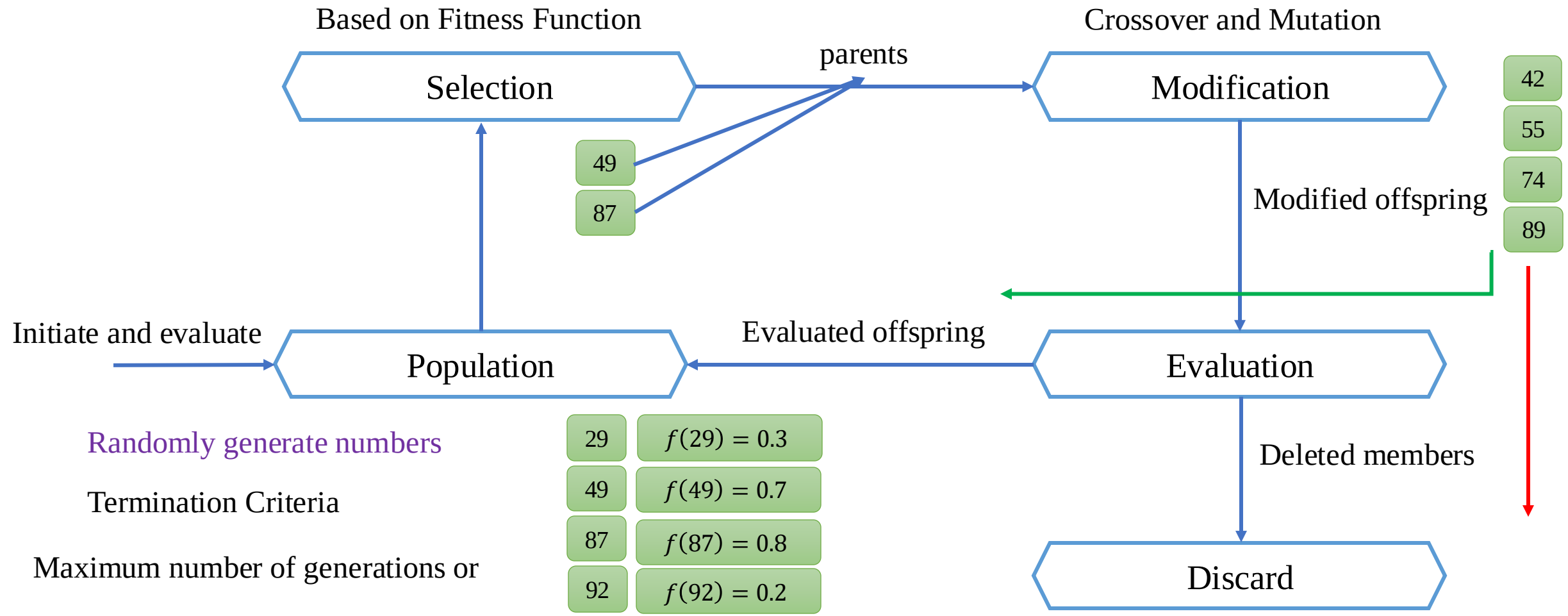
AIFA: Genetic Algorithm

07/04/2025

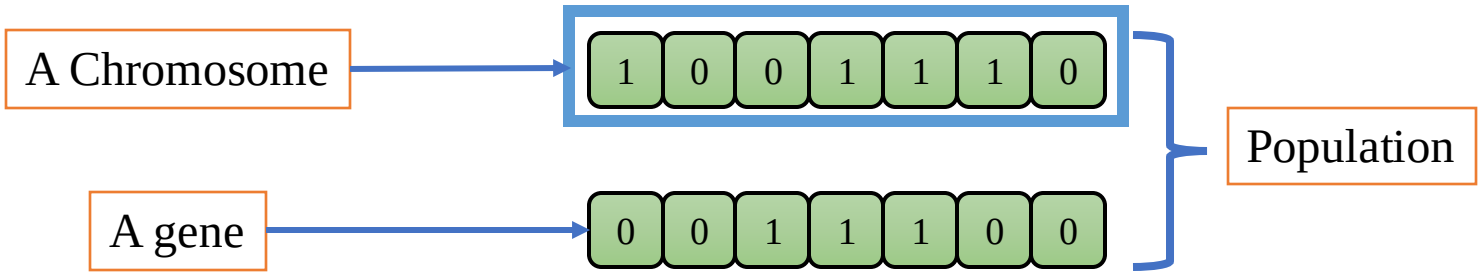
Koustav Rudra

GA: Evolutionary Cycle



Find a number between 1 to 100 that fits functional value f(x)

Genetic Algorithms



- Binary encoding uses 0's and 1's in a chromosome
- Each bit corresponds to a gene
- The values for a given gene are alleles
- A set of chromosomes forms population

Population contains a set of feasible and some infeasible solutions, if needed

Step1: Encoding Problem

- Types of encoding a solution of the problem into chromosome

- Binary encoding
  - Difficulty to apply directly
  - Not a natural encoding

1	0	0	1	1	0

- Value or real number encoding
  - For constrained optimization problem

1.23	0.57	0.12	2.34	3.12	2.27

- Permutation encoding
  - For combinatorial optimization problems
  - TS or Quadratic Assignment Problems

3	2	5	1	8	4	7	6

- Tree encoding

Step1: Encoding a Problem

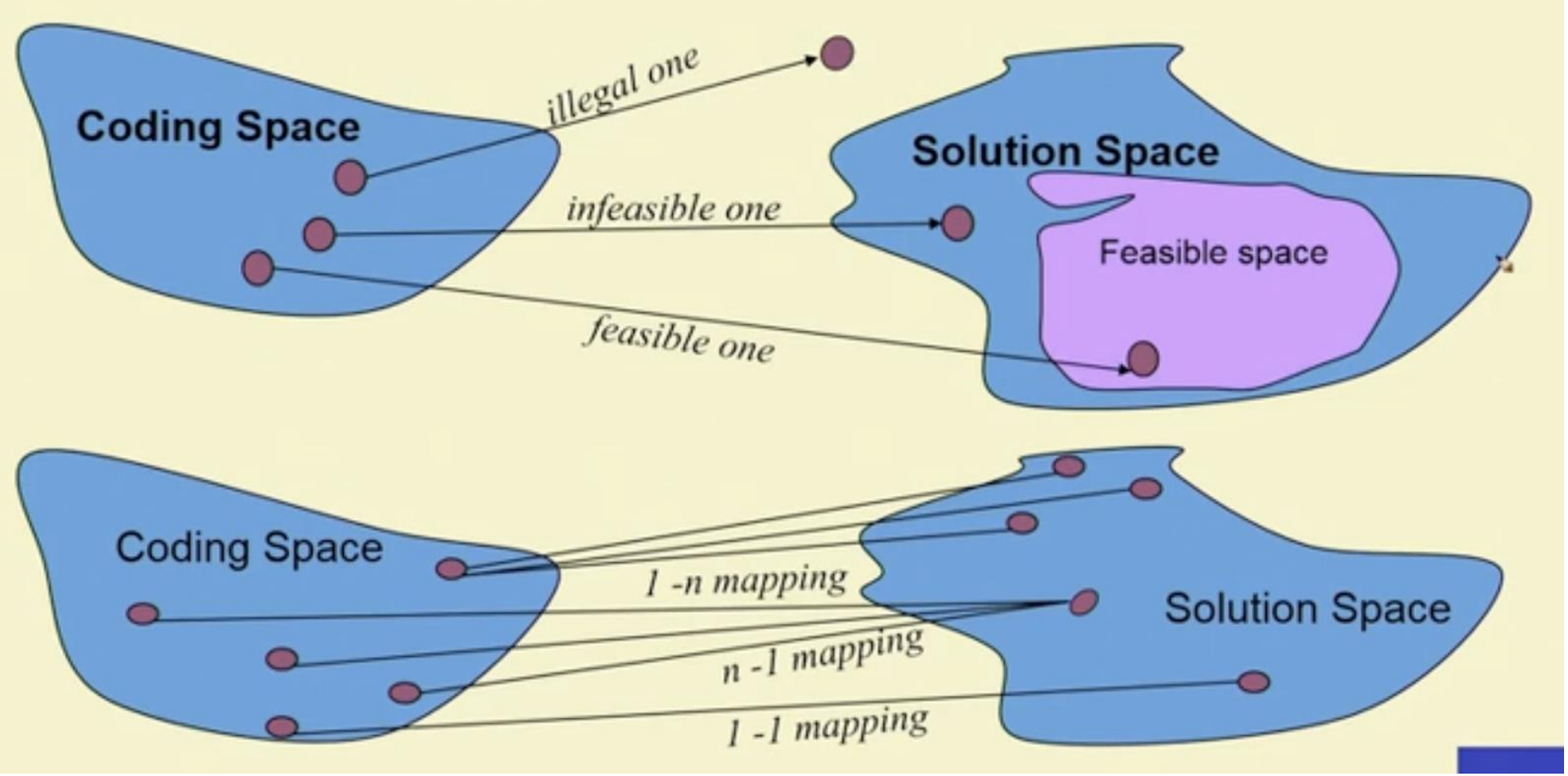
- Critical issues with Encoding

- Feasibility of a chromosome
  - Solution decoded from a chromosome lies in a feasible region of the problem

- Legality of a chromosome
  - Chromosomes represent a solution to the problem

- Uniqueness of mapping (between chromosomes and solution to the problem)
  - Between one-to-many, many-to-one, and one-to-one, one-to-one mapping is highly desirable
  - With one chromosome representing only one solution to the problem

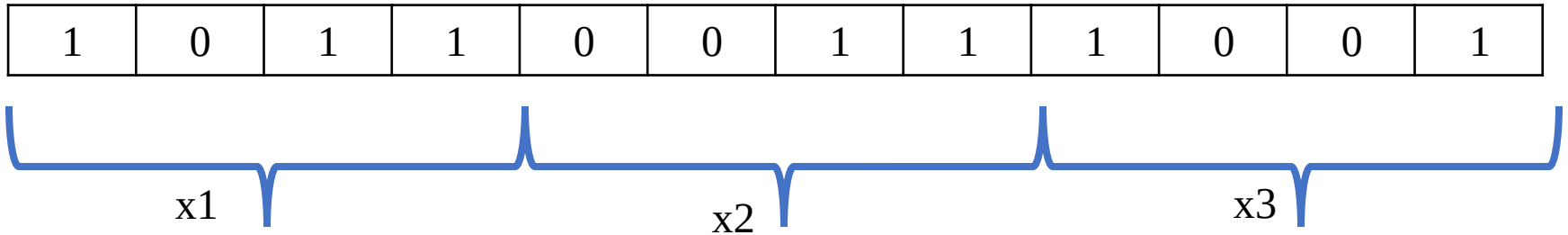
Step1: Encoding a Problem



Step2: Initialization

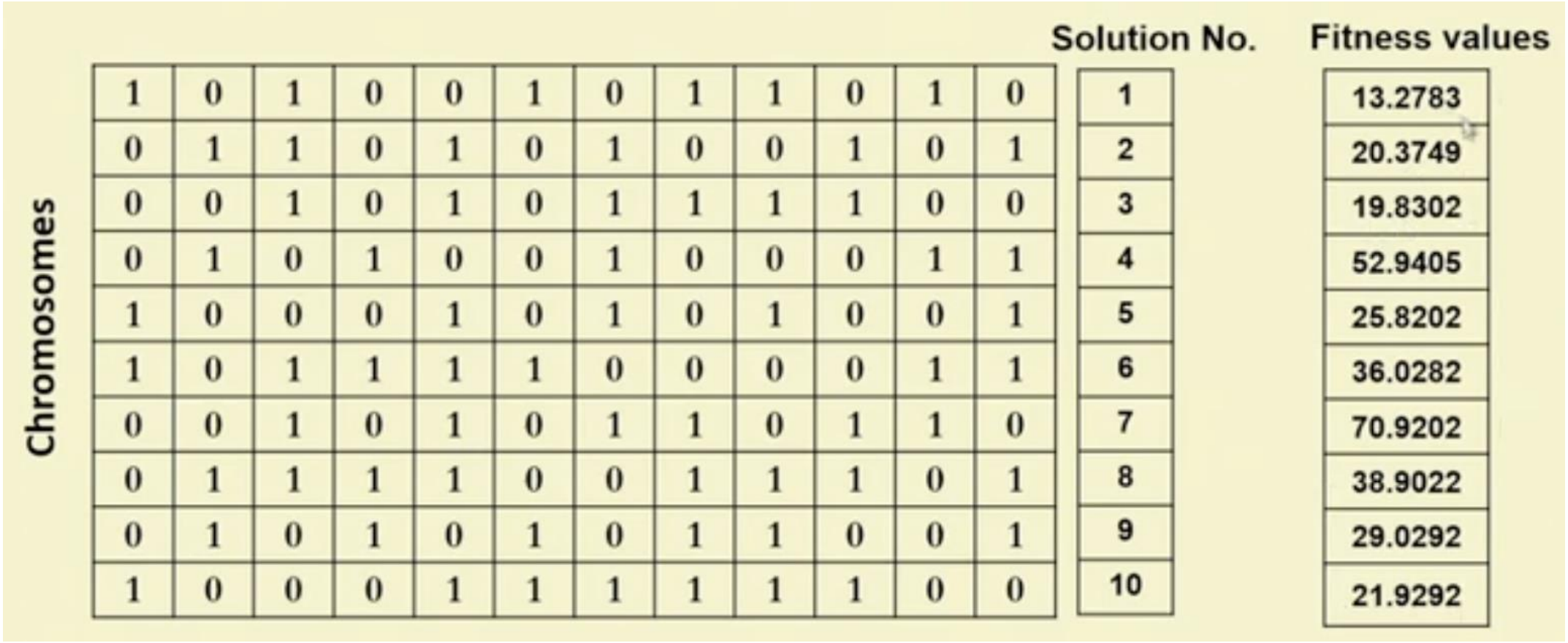
- Create initial population of solutions
  - Through a process of randomization
  - Through a Local search procedure
  - Selecting only feasible solutions

- Consider a Problem: Minimize: F(x1, x2, x3) with binary encoding, we have:



Initialization

- Population of solutions
- Fitness of solutions are evaluated (= objective function)



Chromosomes												Solution No.	Fitness values
1	0	1	0	0	1	0	1	1	0	1	0	1	13.2783
0	1	1	0	1	0	1	0	0	1	0	1	2	20.3749
0	0	1	0	1	0	1	1	1	1	0	0	3	19.8302
0	1	0	1	0	0	1	0	0	0	1	1	4	52.9405
1	0	0	0	1	0	1	0	1	0	0	1	5	25.8202
1	0	1	1	1	1	0	0	0	0	1	1	6	36.0282
0	0	1	0	1	0	1	1	0	1	1	0	7	70.9202
0	1	1	1	1	0	0	1	1	1	0	1	8	38.9022
0	1	0	1	0	1	0	1	1	0	0	1	9	29.0292
1	0	0	0	1	1	1	1	1	1	0	0	10	21.9292

Step3: Selection (“Survival of the fittest”)

- Sampling Mechanisms: Select chromosomes from sampling space

- Stochastic Sampling
- Roulette Wheel Selection:
  - Determine survival probability to fitness value
  - Randomly generate number between [0,1] and select the individual

- Deterministic Samplings
  - Select best individuals from the parents and offspring with no duplication of the individuals

- Mixed Sampling
  - both random and deterministic sampling

Step3: Selection ("Survival of the fittest")

- Sampling Mechanisms: Select chromosomes from sampling space

- Stochastic Sampling
- Roulette Wheel Selection:
  - Determine survival probability to fitness value
  - Randomly generate number between [0,1] and select the individual

- Deterministic Samplings
  - Select best individuals from the parents and offspring with no duplication of the individuals

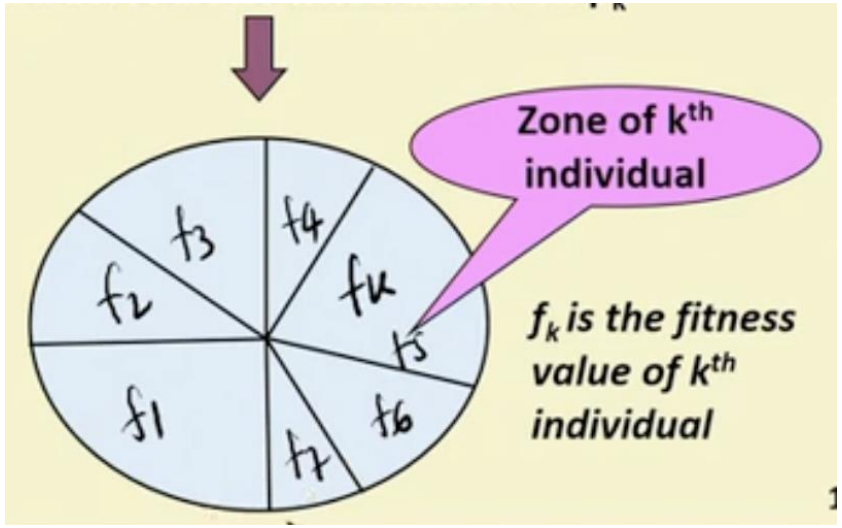
- Mixed Sampling
  - Both random and deterministic sampling

Selection probability for kth individual

$$p_k = \frac{f_k}{\prod_{j=1}^{pop\_size} f_j}$$

How to construct Roulette wheel?

- Calculate cumulative probability
- Construct roulette wheel based on p_k

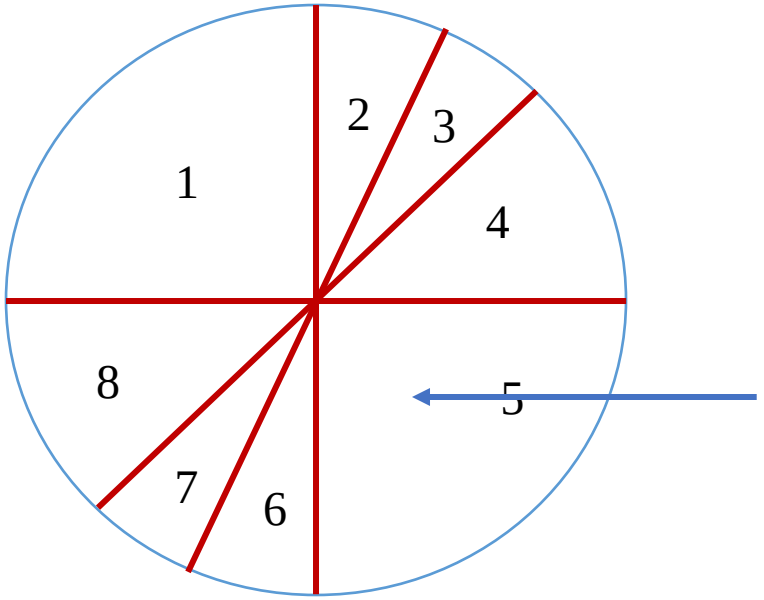


Selection Schemes

- Roulette wheel selection without scaling
- Roulette wheel selection with scaling
- Stochastic tournament selection with a tournament size of two
- Remainder stochastic sampling without replacement
- Remainder stochastic sampling with replacement
- Elitism

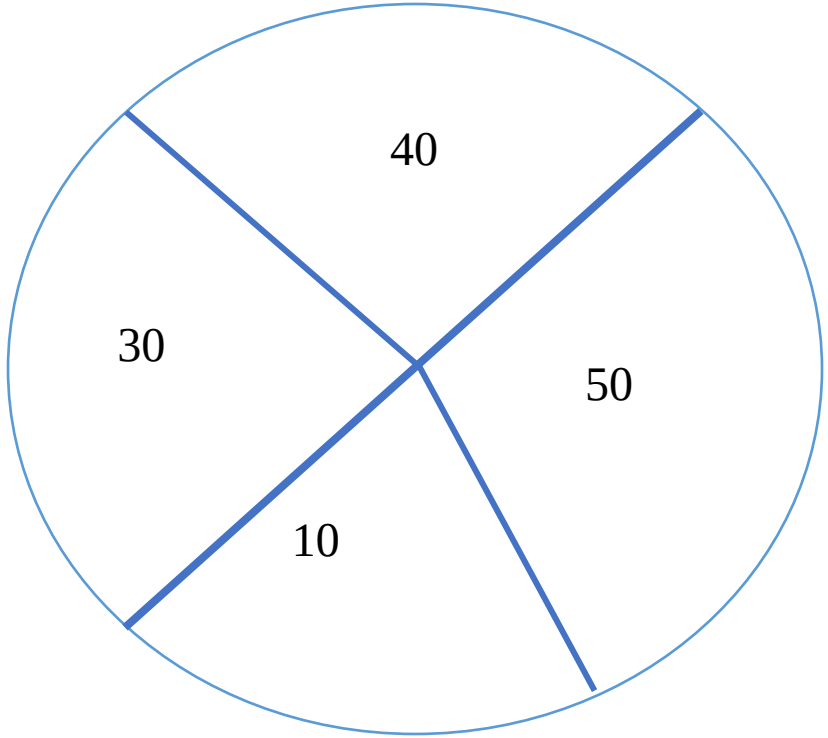
- Which one to use?
- When?

- Balance between population diversity and selection pressure



Roulette wheel selection without scaling

- C1: 10
- C2: 30
- C3: 40
- C4: 50



Roulette wheel selection with scaling

	2			
1		3	7	
				100

	22			
21		23	27	
				120

Tradeoff: Selection pressure (fitted chromosome) vs Population diversity

Remainder stochastic sampling without replacement

- C1: 10
- C2: 30
- C3: 40
- C4: 50
- Average: 32.5

- C1: $^{20}/_{65}$
- C2: $^{60}/_{65}$
- C3: $^{80}/_{65}$
- C4: $^{100}/_{65}$
- Average: 32.5

- C1: $^{20}/_{65}$
- C2: $^{60}/_{65}$
- C3: $1 + {}^{15}/_{65}$
- C4: $1 + {}^{35}/_{65}$
- Average: 32.5

- C3+C4

- C1: $^{20}/_{65}$
- C2: $^{60}/_{65}$
- C3: $^{15}/_{65}$
- C4: $^{35}/_{65}$

Roulette

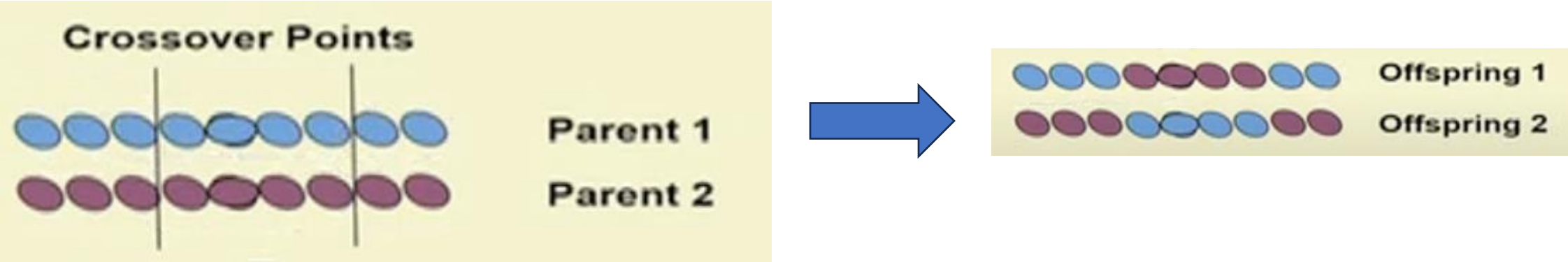
- Average: 32.5

Elitism

- A selected chromosome directly move to next generation
- No crossover or mutation

Reproduction

- Crossover operation (Based on crossover probability)
- Select parents from population based on crossover probability
- Randomly select two points between strings to perform crossover operation
- Perform crossover operations on selected strings
- Known for Local search operation

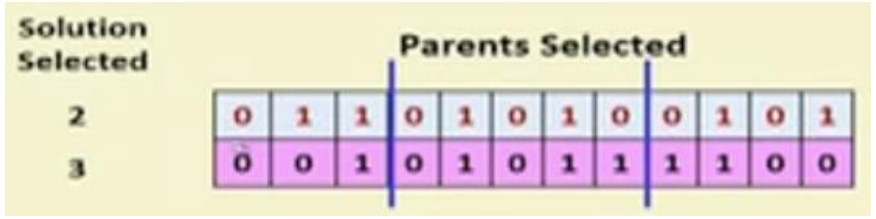


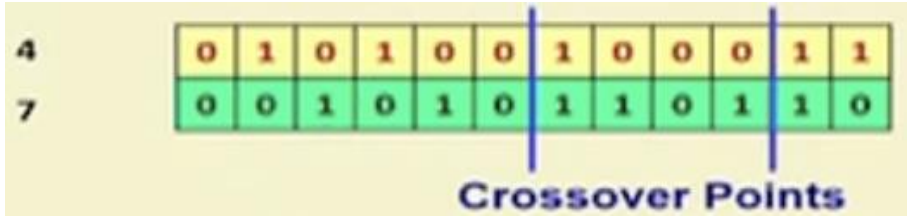
Reproduction

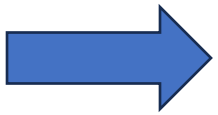
Crossover Probability: 0.8

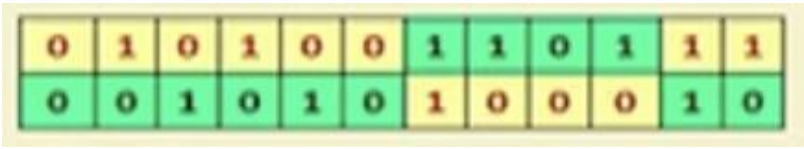
Solution No.	Chromosomes												Random values [0,1]		
1	1	0	1	0	0	1	0	1	1	0	1	0	0.9502	> 0.8	→ No
2	0	1	1	0	1	0	1	0	0	1	0	1	0.2191	< 0.8	→ Yes
3	0	0	1	0	1	0	1	1	1	1	0	0	0.4607	< 0.8	→ Yes
4	0	1	0	1	0	0	1	0	0	0	1	1	0.6081	< 0.8	
5	1	0	0	0	1	0	1	0	1	0	0	1	0.8128	> 0.8	
6	1	0	1	1	1	1	0	0	0	0	1	1	0.9256	> 0.8	
7	0	0	1	0	1	0	1	1	0	1	1	0	0.7779	< 0.8	
8	0	1	1	1	1	0	0	1	1	1	0	1	0.4596	< 0.8	
9	0	1	0	1	0	1	0	1	1	0	0	1	0.9817	> 0.8	
10	1	0	0	0	1	1	1	1	1	1	0	0	0.7784	< 0.8	

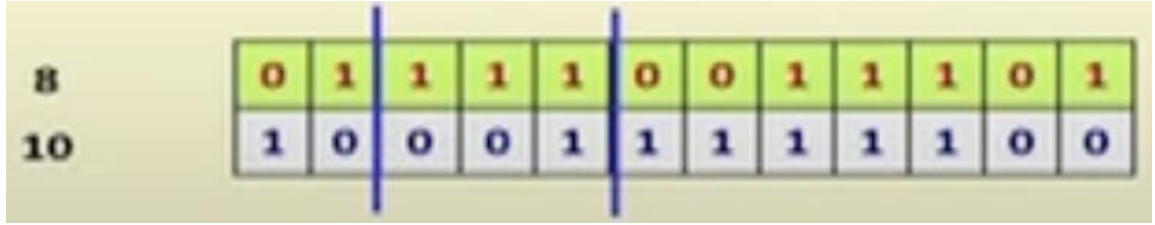
Reproduction











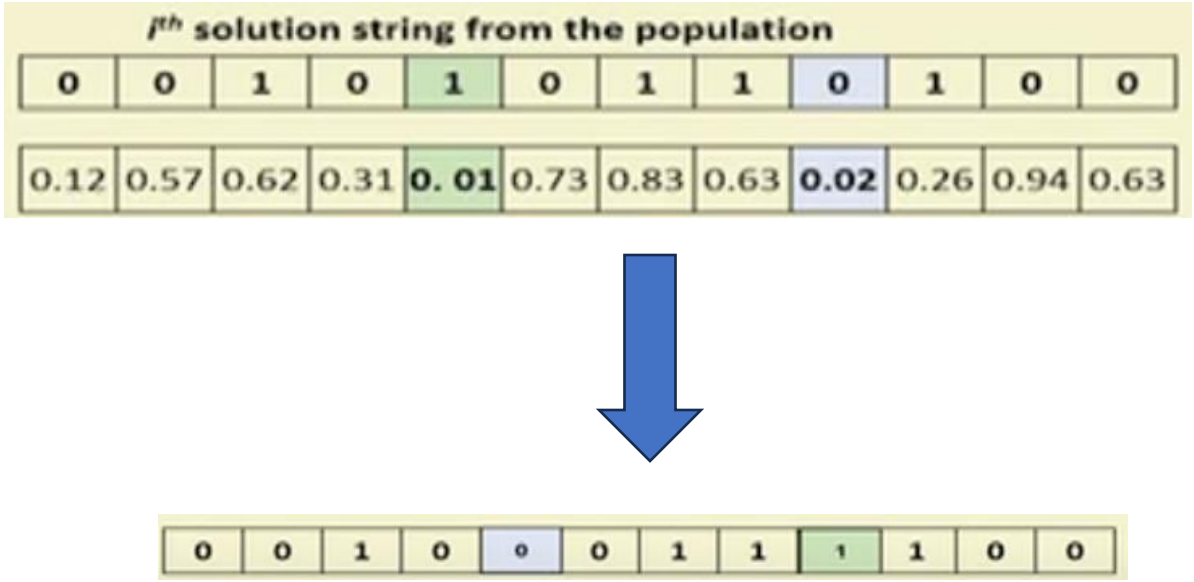
Reproduction

- Mutation operation (based on mutation probability Pm)
- Each bit of every individual is modified with probability Pm
- Main operator for global search (looking at new areas of the search space)
- Pm usually small {0.001,…,0.01}
- Rule of thumb Pm=$^{1}/_{no.of\ bits\ in\ chromosome}$

- 1010 ➔ 0010
- Boost population diversity where solution might be localized
- Make a drastic change

Reproduction

- Minimize: F(x1,x2,x3)
- Pm = 1/12 = 0.083
- Generate Random number [0,1] for each bit
- Select bits having probability less than Pm
- Interchange the bits with each other

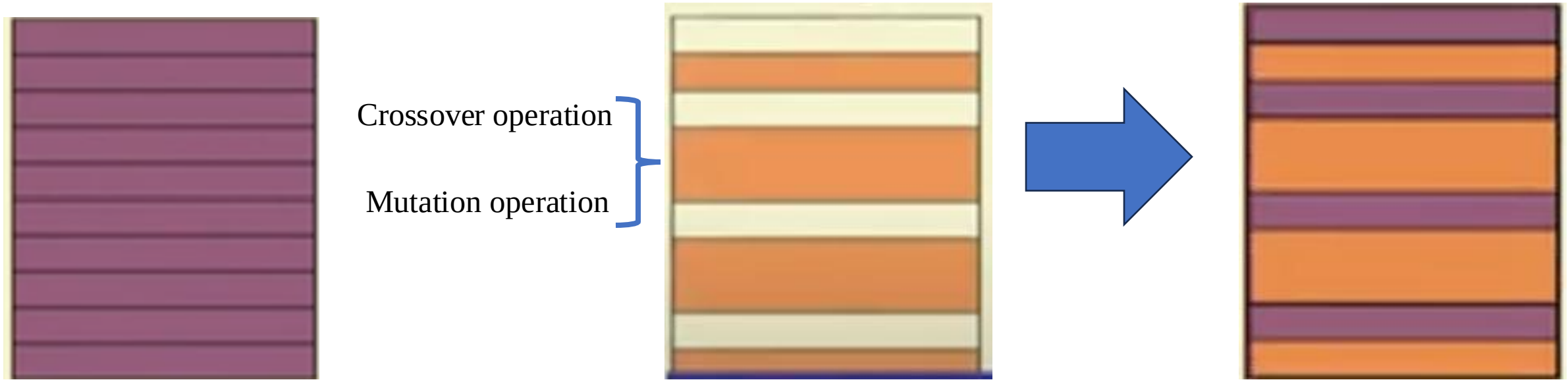


Generating offspring

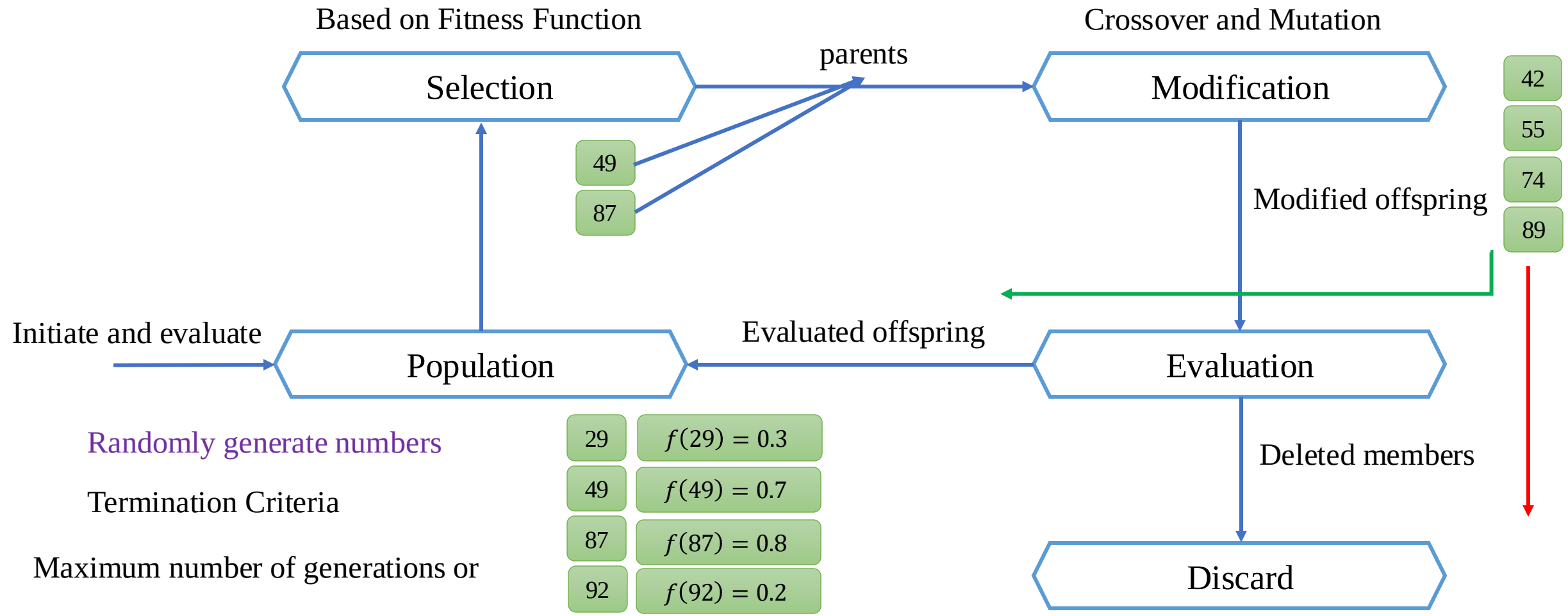
- Directs the search towards promising regions in the search space
- Basic issues involved in selection phase
- Sampling space: Parents and Offspring
- Regular sampling space: all offspring + few parent = pop_size
- Enlarged sampling space: all offspring + all parent

Generating offspring

- Directs the search towards promising regions in the search space
- Basic issues involved in selection phase
- Sampling space: Parents and Offspring
- Regular sampling space: all offspring + few parent = pop_size
- Enlarged sampling space: All offspring + All parent



GA: Evolutionary Cycle



Find a number between 1 to 100 that fits functional value f(x)

Summary of Genetic Algorithms

- Begin
- {
  - Initialize population;
  - Evaluate population;
  - While(Termination Criteria Not Satisfied)
  - {
    - Select parents for reproduction
    - Perform crossover and mutation
    - Evaluate population;
  - }
- }

Parameters for GA

- Empirical studies of Gas show the following:
- Crossover rate should be high generally, about 80-95%
- Mutation rate should be very low
- Crossover and mutation type: operators depend on the chosen encoding and on the problem
- Population size:
  - Very big population size usually does not improve performance of GA – speed actually reduces
  - Good population size is about 20-30
  - The best population size depends on the size of encoded string (chromosomes)
  - More the encoded sizes, more should be the size of population

Parameters for GA

- Selection:
  - Basic Roulette wheel selection can be used, but sometimes rank selection can be better
  - Elitism should be used for sure if you do not use other method for saving the best found solution

- Encoding
  - Depends on the problem and also on the size of instance of the problem

Thank You