Introduction To Genetic Algorithms

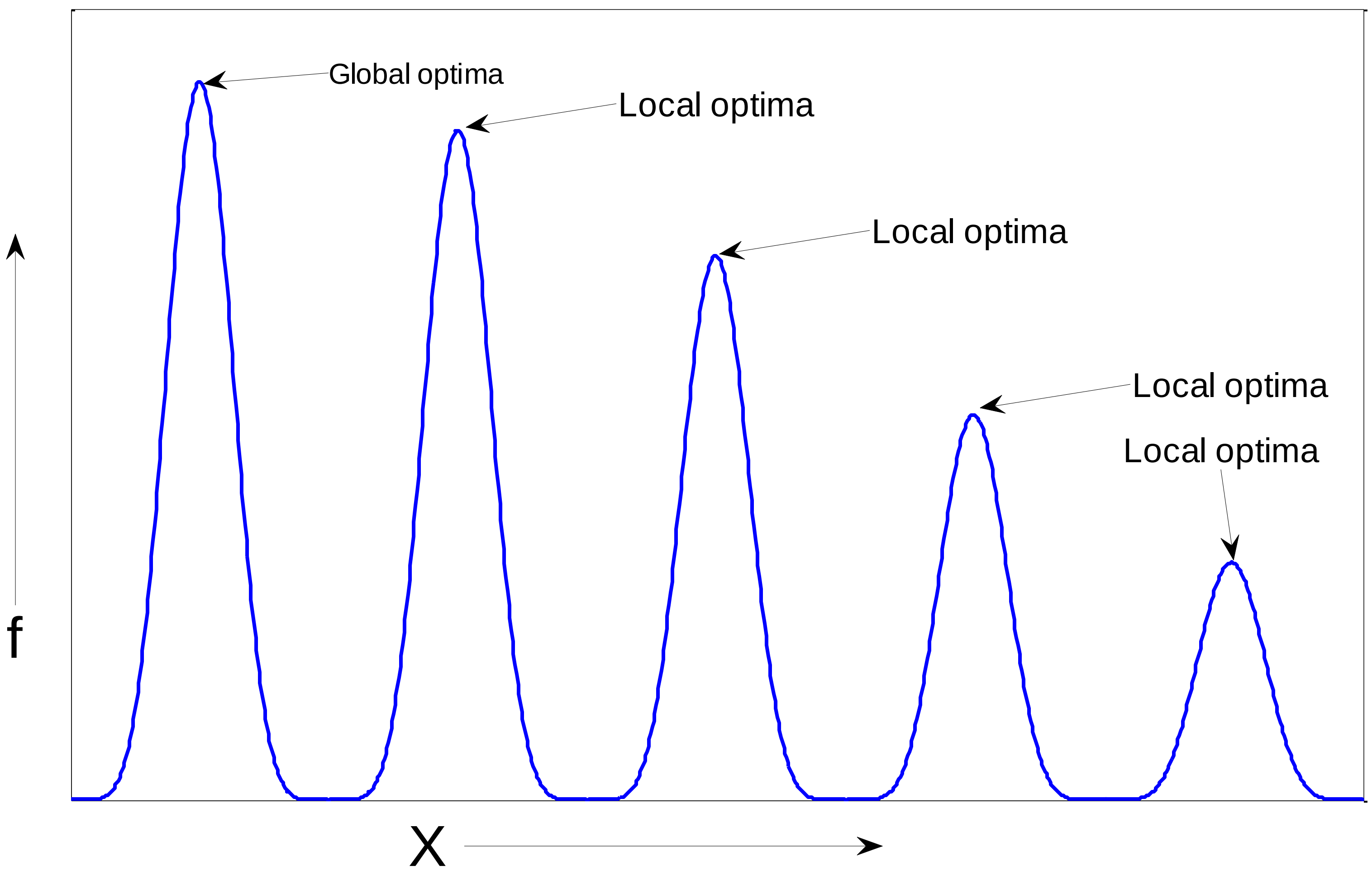
Dr. Rajib Kumar Bhattacharjya
Department of Civil Engineering
IIT Guwahati
Email: rkbc@iitg.ernet.in

16 September 2015

References

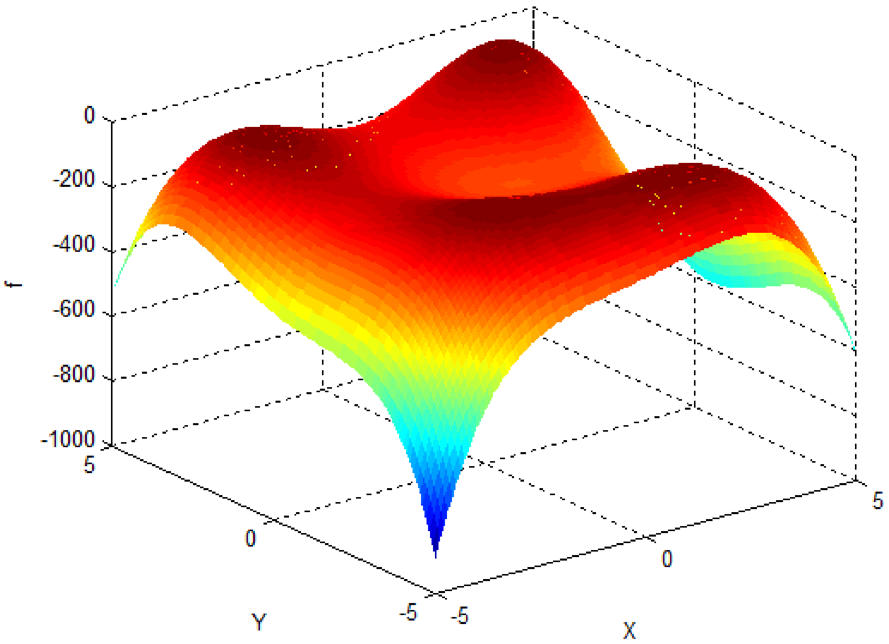
- D. E. Goldberg, ‘Genetic Algorithm In Search, Optimization And Machine Learning’, New York: Addison – Wesley (1989)
- John H. Holland ‘Genetic Algorithms’, Scientific American Journal, July 1992.
- Kalyanmoy Deb, ‘An Introduction To Genetic Algorithms’, Sadhana, Vol. 24 Parts 4 And 5.

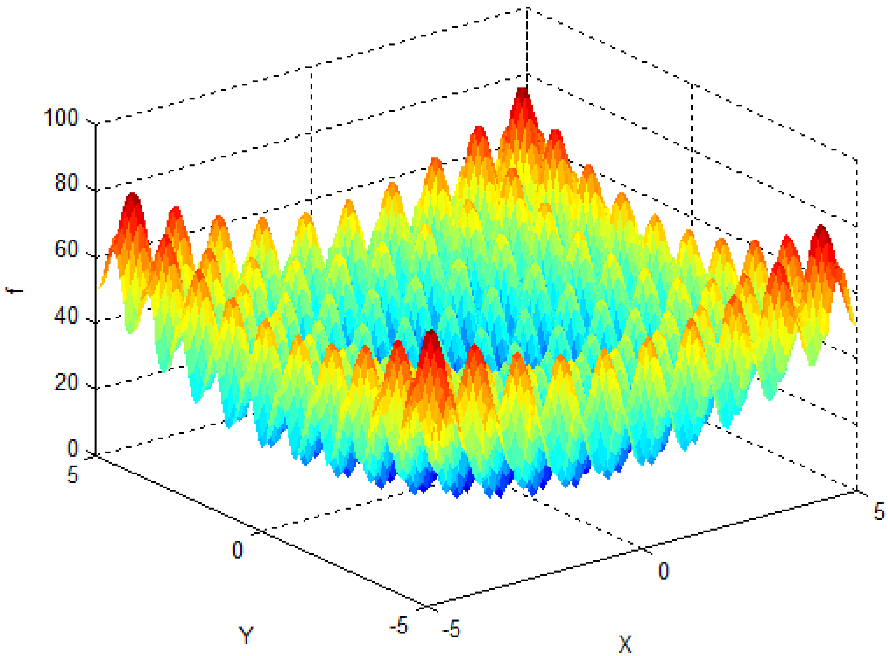
Introduction to optimization



Introduction to optimization

Multiple optimal solutions

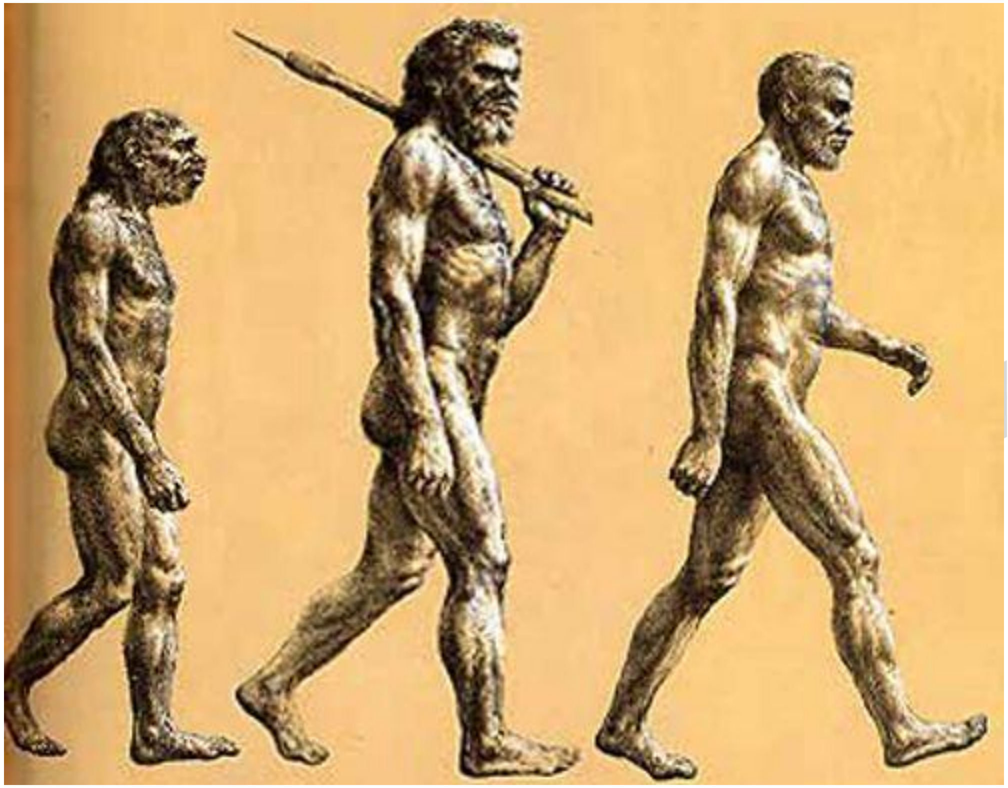




Genetic Algorithms

Genetic Algorithms are the heuristic search and optimization techniques that mimic the process of natural evolution.

Principle Of Natural Selection



"Select The Best, Discard The Rest"

An Example….

Giraffes have long necks

- Giraffes with slightly longer necks could feed on leaves of higher branches when all lower ones had been eaten off.
- They had a better chance of survival.
- Favorable characteristic propagated through generations of giraffes.

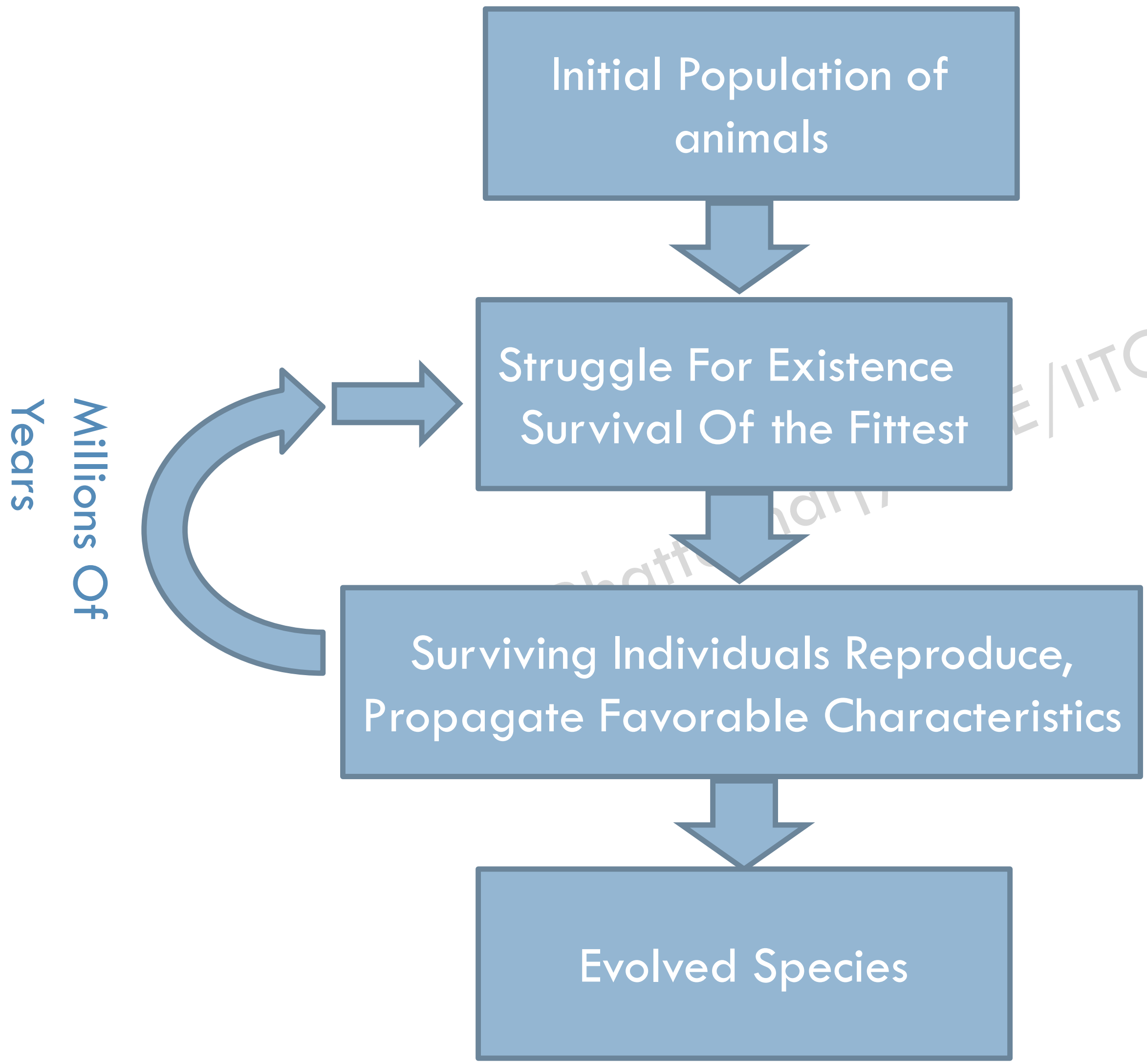
- Now, evolved species has long necks.

An Example….

This longer necks may have due to the effect of mutation initially. However as it was favorable, this was propagated over the generations.

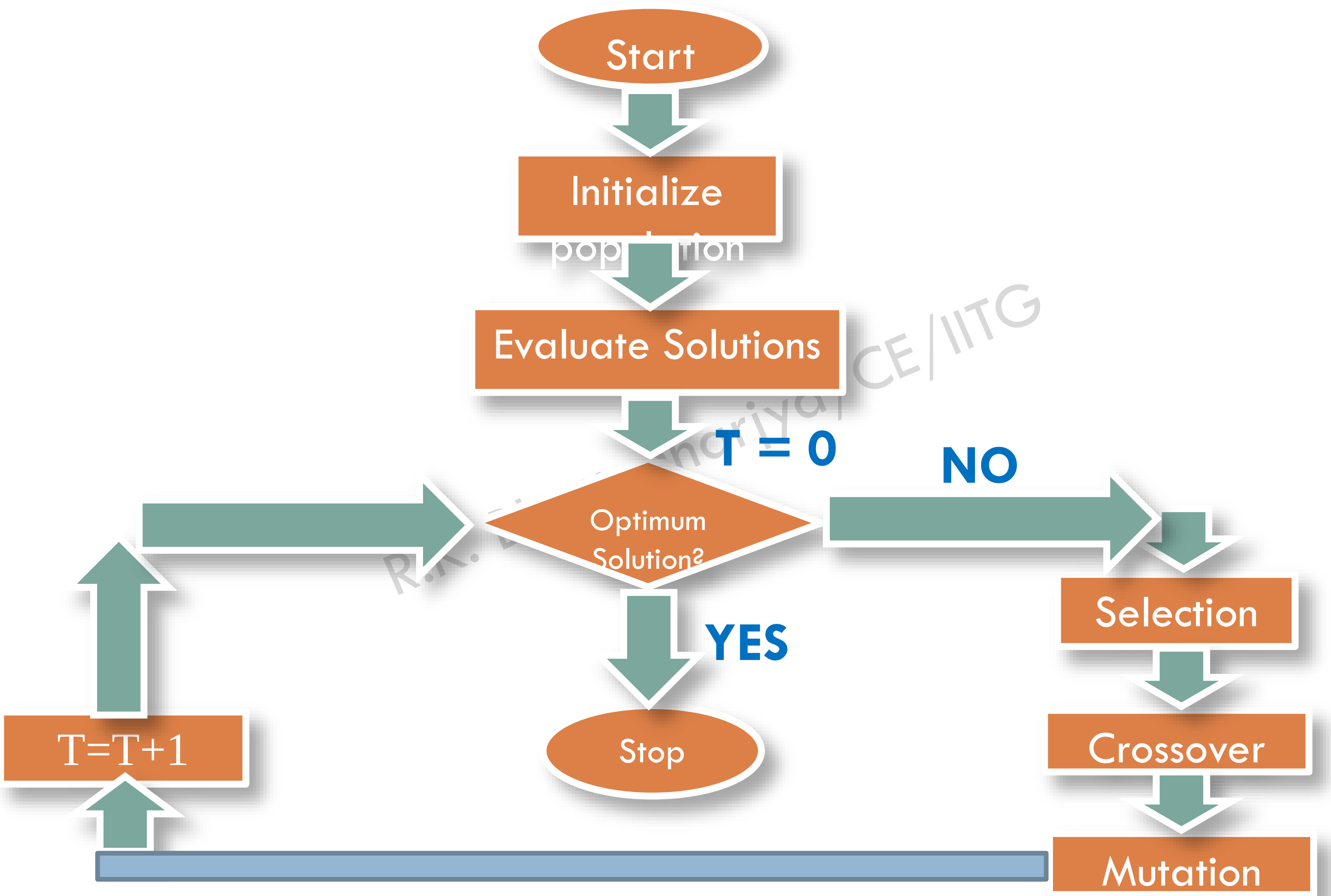
Evolution of species

R.K. Bhattacharjya/CE/IITG



Thus genetic algorithms implement the optimization strategies by simulating evolution of species through natural selection

Simple Genetic Algorithms



Simple Genetic Algorithm


```
function sga ()
        {
        Initialize population;
        Calculate fitness function;

        While(fitness value != termination criteria)
                {
                   Selection;

                   Crossover;

                   Mutation;

                Calculate fitness function;
                }
        }
```

GA Operators and Parameters

- Selection
- Crossover
- Mutation
- Now we will discuss about genetic operators

Selection

The process that determines which solutions are to be preserved and allowed to reproduce and which ones deserve to die out.

The primary objective of the selection operator is to emphasize the good solutions and eliminate the bad solutions in a population while keeping the population size constant.

"Selects the best, discards the rest"

Functions of Selection operator

Identify the good solutions in a population

Make multiple copies of the good solutions

Eliminate bad solutions from the population so that multiple copies of good solutions can be placed in the population

Now how to identify the good solutions?

Fitness function

A fitness value can be assigned to evaluate the solutions

A fitness function value quantifies the optimality of a solution. The value is used to rank a particular solution against all the other solutions

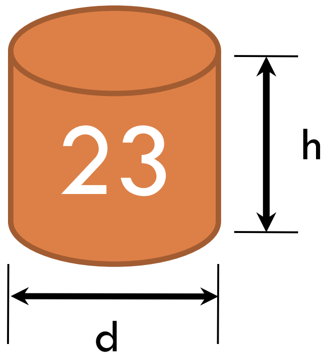
A fitness value is assigned to each solution depending on how close it is actually to the optimal solution of the problem

Assigning a fitness value

$$\begin{aligned}
\text{Minimize} \quad & f(d,h) = c((\pi d^2/2) + \pi d h), \\
\text{Subject to} \quad & g_1(d,h) \equiv (\pi d^2 h/4) \geq 300, \\
\text{Variable bounds} \quad & d_{\min} \leq d \leq d_{\max}, \\
& h_{\min} \leq h \leq h_{\max}.
\end{aligned}$$

Considering c = 0.0654

$$\begin{aligned}
F(s) &= 0.0654(\pi(8)^2/2 + \pi(8)(10)), \\
&= 23,
\end{aligned}$$



Selection operator

- There are different techniques to implement selection in Genetic Algorithms.

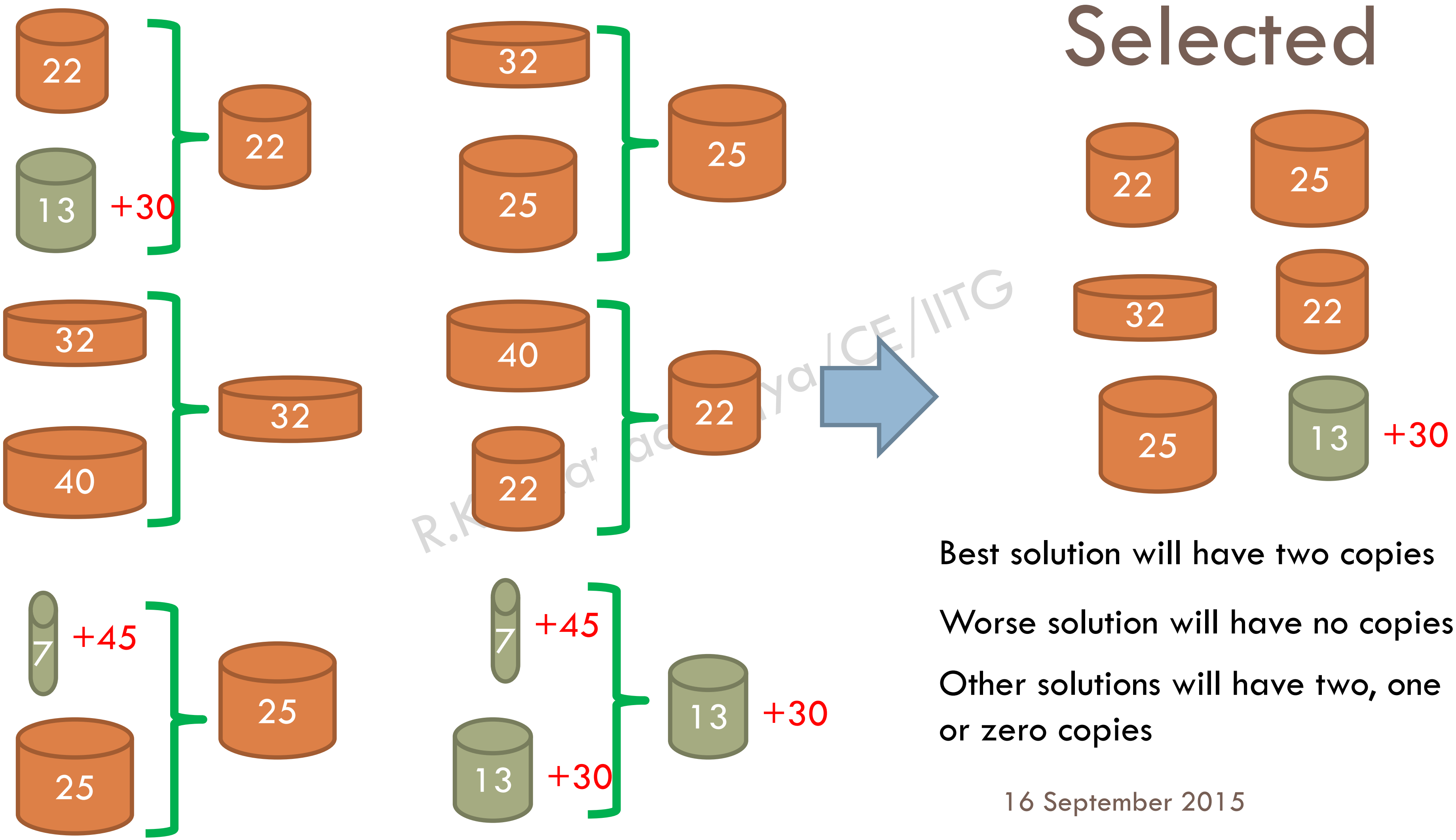
- They are:

  - Tournament selection
  - Roulette wheel selection
  - Proportionate selection
  - Rank selection
  - Steady state selection, *etc*

Tournament selection

- In tournament selection several tournaments are played among a few individuals. The individuals are chosen at random from the population.
- The winner of each tournament is selected for next generation.
- Selection pressure can be adjusted by changing the tournament size.
- Weak individuals have a smaller chance to be selected if tournament size is large.

Tournament selection



Best solution will have two copies

Worse solution will have no copies

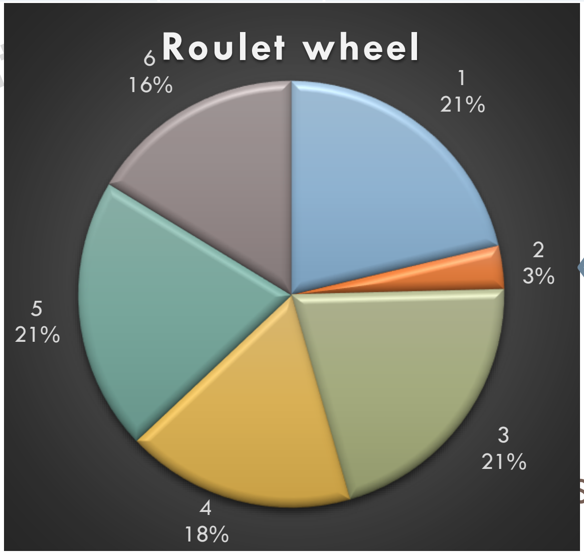
Other solutions will have two, one or zero copies

Roulette wheel and proportionate selection

Chrom #	Fitness	% of RW	EC	AC
1	50	26.88	1.61	2
2	6	3.47	0.19	0
3	36	20.81	1.16	1
4	30	17.34	0.97	1
5	36	20.81	1.16	1
6	28	16.18	0.90	1
	186	100.00	6	6

Parents are selected according to their fitness values

The better chromosomes have more chances to be selected



Rank selection

R.K. Bhattacharjya/CE/IITG

Chrom #	Fitness
1	37
2	6
3	36
4	30
5	36
6	28

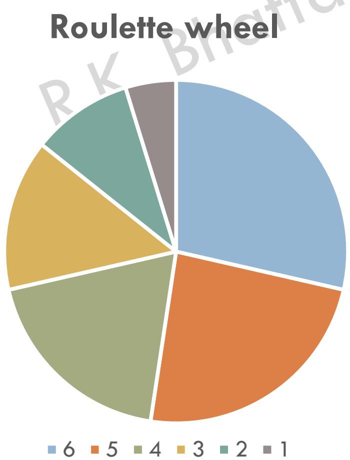
Sort according to fitness

Chrom #	Fitness
1	37
3	36
5	36
4	30
6	28
2	6

Assign raking

Chrom #	Rank
1	6
3	5
5	4
4	3
6	2
2	1

Chrom #	% of RW
1	29
3	24
5	19
4	14
6	10
2	5



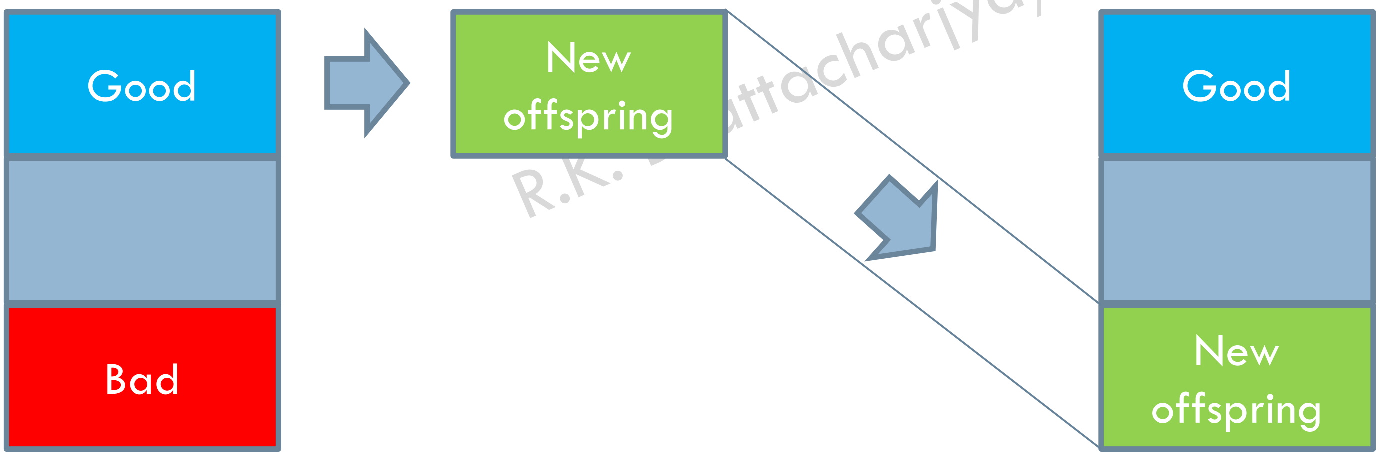
Chrom #	EC	AC
1	1.714	2
3	1.429	1
5	1.143	1
4	0.857	1
6	0.571	1
2	0.286	0

Steady state selection

In this method, a few good chromosomes are used for creating new offspring in every iteration.

Then some bad chromosomes are removed and the new offspring is placed in their places

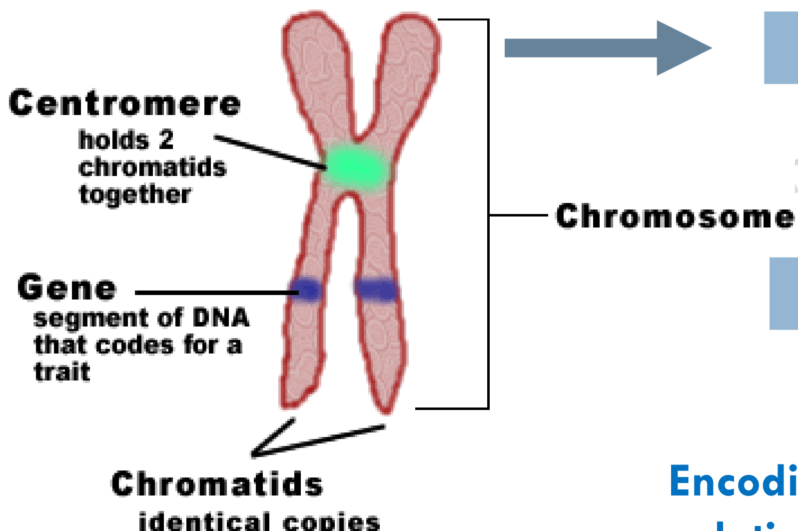
The rest of population migrates to the next generation without going through the selection process.

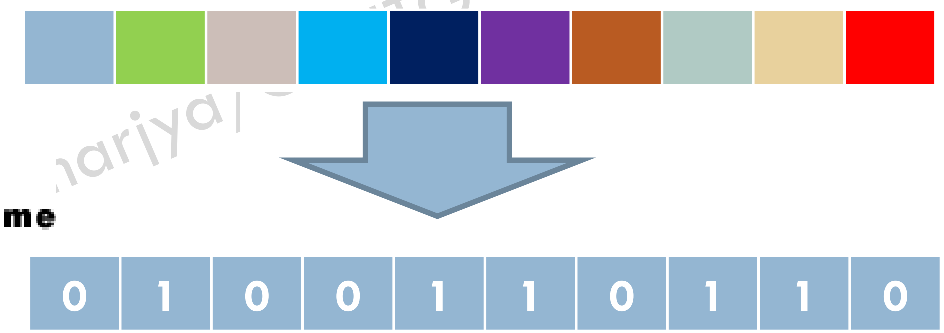


How to implement crossover

The crossover operator is used to create new solutions from the existing solutions available in the mating pool after applying selection operator.

This operator exchanges the gene information between the solutions in the mating pool.





Encoding of solution is necessary so that our solutions look like a chromosome

Source: http://www.biologycorner.com/bio1/celldivision-chromosomes.html

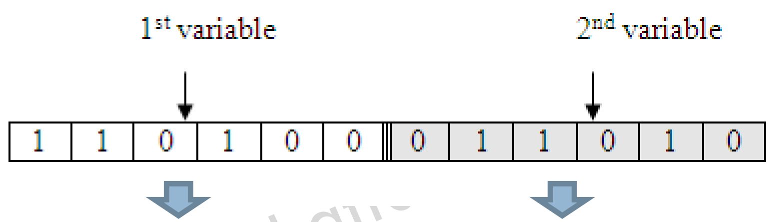
Encoding

The process of representing a solution in the form of a string that conveys the necessary information.

Just as in a chromosome, each gene controls a particular characteristic of the individual, similarly, each bit in the string represents a characteristic of the solution.

Encoding Methods

- Most common method of encoding is binary coded. Chromosomes are strings of 1 and 0 and each position in the chromosome represents a particular characteristic of the problem



Decoded value 52 26

Mapping between decimal and binary value

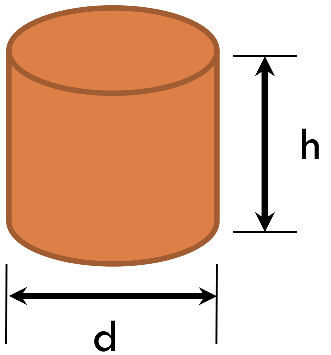
$$x_i = x_i^{\min} + \frac{x_i^{\max} - x_i^{\min}}{2^{l_i} - 1} \mathrm{DV}(s_i)$$

Encoding Methods

$$
\begin{aligned}
\text{Minimize} \quad & f(d,h) = c((\pi d^2/2) + \pi dh), \\
\text{Subject to} \quad & g_1(d,h) \equiv (\pi d^2 h/4) \geq 300, \\
\text{Variable bounds} \quad & d_{\min} \leq d \leq d_{\max}, \\
& h_{\min} \leq h \leq h_{\max}.
\end{aligned}
$$

Defining a string [0100001010]

d h



(d,h) = (8,10) cm

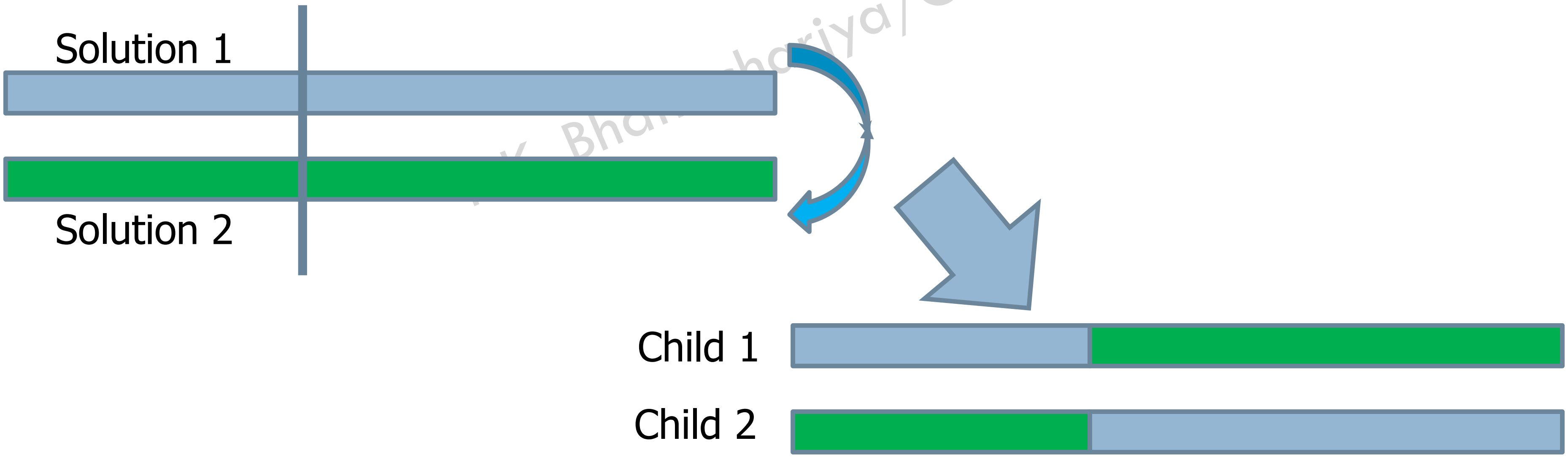
Chromosome = [0100001010]

Crossover operator

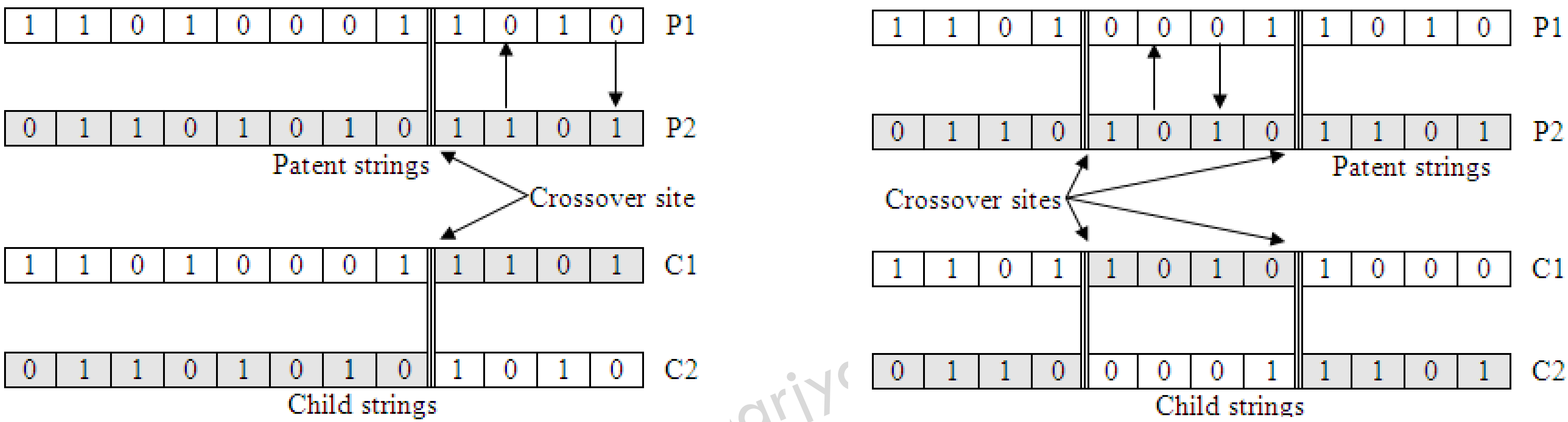
The most popular crossover selects any two solutions strings randomly from the mating pool and some portion of the strings is exchanged between the strings.

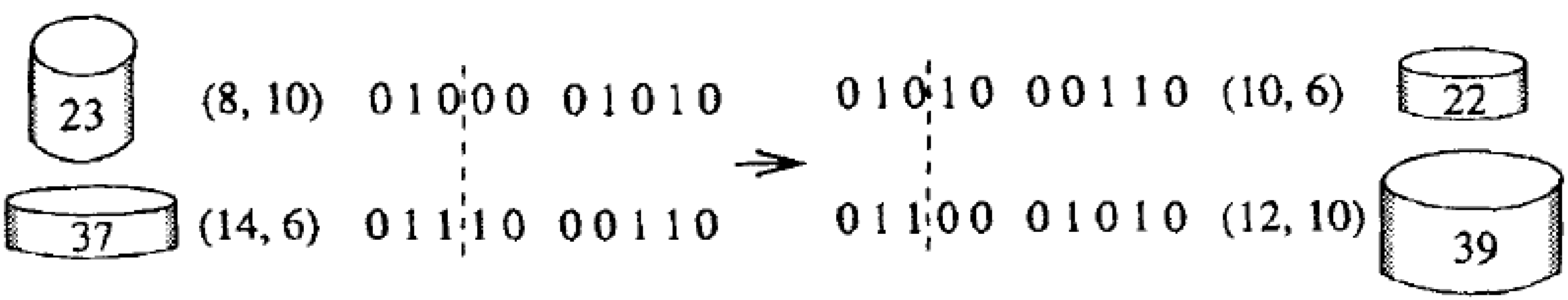
The selection point is selected randomly.

A probability of crossover is also introduced in order to give freedom to an individual solution string to determine whether the solution would go for crossover or not.



Binary Crossover





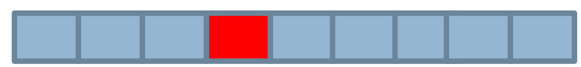
Source: Deb 1999

Mutation operator

Mutation is the occasional introduction of new features in to the solution strings of the population pool to maintain diversity in the population.

Though crossover has the main responsibility to search for the optimal solution, mutation is also used for this purpose.

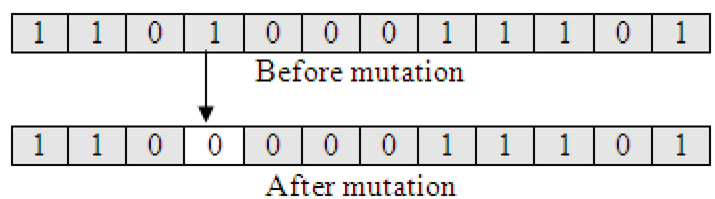
Before mutation



After mutation

Binary Mutation

- Mutation operator changes a 1 to 0 or vise versa, with a mutation probability of .
- The mutation probability is generally kept low for steady convergence.
- A high value of mutation probability would search here and there like a random search technique.



Source: Deb 1999

Elitism

- Crossover and mutation may destroy the best solution of the population pool
- Elitism is the preservation of few best solutions of the population pool
- Elitism is defined in percentage or in number

Nature to Computer Mapping

Nature	Computer
Population	Set of solutions
Individual	Solution to a problem
Fitness	Quality of a solution
Chromosome	Encoding for a solution
Gene	Part of the encoding solution

An example problem

$$Maximize\ f(x) = \sin(x)$$

$$0 \leq x \leq \pi$$

Consider 6 bit string to represent the solution, then
000000 = 0 and 111111 = π

Assume population size of 4

Let us solve this problem by hand calculation

An example problem

Initialize population → Calculate decoded value → Calculate real value → Calculate objective function value → Calculate fitness value → Calculate relative fitness value → Calculate expected count → Calculate actual count

Initial population		Decoding		Fitness calculation		Selection: Proportionate selection		
Sol No	Binary String	DV	x value	f	F	Relative Fitness	Expected count	Actual count
1	100101	37	0.587	0.96	0.96	0.38	1.53	2
2	001100	12	0.19	0.56	0.56	0.22	0.89	1
3	111010	58	0.921	0.25	0.25	0.10	0.39	0
4	101110	46	0.73	0.75	0.75	0.30	1.19	1
				Avg	0.563			
				Max	0.96			

An example problem: Crossover

Matting pool → Random generation of crossover site → New population

Crossover: Single point

Sol No	Matting pool	CS	New Binary String	DV	x value	f	F
1	100101	3	100100	36	0.57	0.97	0.97
2	001100	3	001101	13	0.21	0.60	0.60
3	100101	2	101110	46	0.73	0.75	0.75
4	101110	2	100101	37	0.59	0.96	0.96
						Avg	0.8223
						Max	0.97

An example problem: Mutation

	Mutation					
Sol No	Population after crossover	Population after mutation	DV	x value	f	F
1	100100	100000	32	0.51	1.00	1.00
2	001101	101101	45	0.71	0.78	0.78
3	101110	100110	38	0.60	0.95	0.95
4	100101	101101	45	0.71	0.78	0.78
					Avg	0.878
					Max	1.00

THANKS