

2.2 Evolutionary algorithms: Genetic Algorithm

- Evolutionary algorithms (EA) represent a broad spectrum of heuristic approaches for simulating biological evolution in the search for improved “fitness,” i.e., the best values of decision variables and parameters based on an objective or fitness function.
- Evolutionary algorithms are broadly based on the repeated mutation and recombination and selection: in each generation (iteration) to define new individuals (candidate solutions).
- Primary examples include:
 - Genetic algorithms
 - Evolutionary strategies
 - Evolutionary programming and
 - Genetic programming

Genetic Algorithm (GA)

- GA are randomized general purpose search techniques used for finding the best values of the parameters or decision variables of existing models.
- GAs and their variations are based on the mechanisms of natural selection.
- Unlike conventional optimization search approaches based on gradients, GAs work on populations of possible solutions, attempting to find a solution set that either maximizes or minimizes the value of a function of those parameters and decision variables. This function is called an objective function.

GA

- Some populations of solutions may improve the value of the objective function, others may not.
- The ones that improve its value play a greater role in the generation of new populations of solutions.
- This process continues until no significant improvement in model output is apparent.
- Just how “good” or “fit” a particular population of parameter and decision variable values must be evaluated using a model of the system that contains these parameters and decision variables. This system model is separated from the GA model. This makes GA applicable for estimating the best parameter and decision variable values of a wide variety of simulation models.

GA

- Each individual solution set of a GA model contains values of all the parameters or variables whose best values are being sought. These solutions are expressed as strings of values.
- For example, if the values of three variables x , y , and z are to be obtained, these variables are arranged into a string, xyz . Assuming each variable is expressed using three digits, then the string 056004876 would represent $x = 56$, $y = 4$, and $z = 876$.
- These strings (array of numbers) are called chromosomes. Chromosomes are usually represented by strings of binary numbers. The numbers in the chromosome are called genes.
- Pairs of chromosomes (parents) join together and produce offspring, who in turn inherit some of genes of the parents.

Main Features of GA

- Altered genes that result in improved values of the objective function will tend to survive from generation to generation, while those that are inferior will tend to die and not reappear in future population sets.
- To illustrate the main features of genetic algorithms, consider the problem of finding the best allocations of water to the three water consuming irrigation farms seen in chapter 2.
- The maximum allocation, x_i , to any single user i cannot exceed 5, and the sum of all allocations cannot exceed the value of $Q - R$, say $10 - 2 = 8$.
 - $0 \leq x_i \leq 5$ for $i = 1, 2$ and 3
 - $x_1 + x_2 + x_3 \leq 8$

Example

- The objective is to find the values of each allocation that maximizes the net benefits, $NB(x)$.
- Maximize Net benefit Subject to constraints:
 - Net benefit = Total return – Total cost
 - Total return = $p_1(12 - p_1) + p_2(20 - 1.5p_2) + p_3(28 - 2.5p_3)$
 - Total cost = $3(p_1)^{1.3} + 5(p_2)^{1.2} + 6(p_3)^{1.15}$
 - Where $p_1 \leq 0.4(x_1)^{0.9}$, $p_2 \leq 0.5(x_2)^{0.8}$ and $p_3 \leq 0.6(x_3)^{0.7}$
 - Water-allocation restriction: $R + x_1 + x_2 + x_3 = Q$
- A population of possible feasible solutions is generated randomly. The best size of the sample solution population - the number of solutions being considered - is usually determined by trial and error.

Cross-over and Mutation

- Using numbers to the base 10, a sample individual solution (chromosome) could be 312, representing the allocations $x_1 = 3$, $x_2 = 1$, and $x_3 = 2$. Another individual solution, picked at random, might be 101.
- These two individuals or chromosomes, each containing three genes, can pair up and have two children. The genes of the children are determined by crossover and mutation operations. These pairing, crossover and mutation operations are random.
- Crossover involves splitting the two solution strings into two parts, each string at the same place. Assume the location of the split was randomly determined to be after the first digit,
 - (3|12) and (1|01) the two new individuals are 301 and 112.

Cross-over and Mutation

- Another crossover approach is to determine for each corresponding pair of genes whether or not they will be exchanged.
- For example, suppose the probability of a crossover was set at 0.30. Thus, an exchange of each corresponding pair of genes in a string or chromosome has a 30% chance of being exchanged. Assume as the result of this “uniform” crossover procedure, only the middle gene in the pair of strings 312 and 101 is exchanged. This would result in 302 and 111.
- NOTE: The probability of a mutation is usually much smaller than that of a crossover.

Mutation

- Random mutation operations can apply to each gene in each string. Mutation involves changing the value of the gene being mutated.
- If these strings contain binary numbers, a 1 would be changed to 0, and a 0 would be changed to 1.
- If numbers to the base 10 are used as they are here, any reasonable mutation scheme can be defined.
- For example, suppose the mutation of a number reduces/increase it by 1, unless the resulting number is infeasible. Hence in this example, a mutation could be such that if the current value of the gene being mutated (reduced) is 0, then the new number is 5.
- Suppose the middle digit 1 of the second new individual, 112, is randomly selected for mutation. Thus, its value changes from 1 to 0. The new string is 102.

Mutation

- The mutation and cross over processes result in a new population of individuals (children) for which each child's fitness, or objective value, can be determined.
- For maximization objective function, the higher the value the better.
- Adding up all the objective values associated with each child in the population, and then dividing each child's objective value by this total sum yields a fraction for each child.
- That fraction is the probability of that child being selected for the new population of possible solutions.
- The higher the objective value of a child, the higher the probability of its being selected to be a parent in a new population.

Example ...

- In this example, the objective is to maximize the total benefit derived from the allocation of water,
- For the maximization equation the string 301 has a total net benefit of 21.0. The string 102 has a total benefit of 22.8.
- Considering just these two children, the sum of these two benefits is 43.8. Thus the child (string) 301 and child (string) 102 have a probability of $21.0/43.8 = 0.48$ and $22.8/43.8 = 0.52$ of being selected for the new population, respectively.
- Drawing from a uniform distribution of numbers ranging from 0 to 1, if a random number is in the range 0–0.48, then the string 301 would be selected. If the random number exceeds 0.48, then the string 102 would be selected.
- A more realistic example the population size should involve hundreds of strings.

Example ...

- This selection or reproduction mechanism tends to transfer to the next generation, the better (more fit) individuals of the current generation.
- Genetic algorithms involve numerous iterations of the operations just described. Each iteration (or generation) produces populations that tend to contain better solutions.
- The best solution of all populations of solutions should be saved.
- The genetic algorithm process can end when there is no significant change in the values of the best solution that has been found.
- In this search process, there is no guarantee this best solution will be the best that could be found, that is, a global optimum.

The code

```
clc, clear
% this code provides genetic algorithm based optimization
% [x,fval] = ga(fitnessfcn,nvars,A,b,Aeq,beq)
% finds a local minimum x to fitnessfcn, subject to the linear
equalities Aeq*x = beq as well as A*x ≤ b.
% (Set Aeq=[] and beq=[] if no linear equalities exist.)
% x is Best point that ga located during its search.
% fval is Fitness function evaluated at x.
% ga evaluates the matrix product Aeq*x as if x is transposed (Aeq*x').
%for the water allocation problem the following describes the inequality
constraint
A=[1 1 1;1 0 0;0 1 0; 0 0 1;-1 0 0;0 -1 0; 0 0 -1]; b=[8;5;5;5;0;0;0];
[x,fval,exitflag,output,population]=ga(@objfun1,3,A,b,[],[]);
fval=-fval;

function obf=objfun1(x)
p1=0.4*x(1,1)^0.9; p2=0.5*x(1,2)^0.8; p3=0.6*x(1,3)^0.7;
c1=3*p1^1.3; c2=5*p2^1.2; c3=6*p3^1.15;
r1=p1*(12-p1); r2=p2*(20-1.5*p2); r3=p3*(28-2.5*p3);
obf=(c1+c2+c3)-(r1+r2+r3); %Note here that ga works for minimization
while our objective... %is to maximize the benefit which is equivalent
to minimization of the cost.. %benefit difference.
```

Example: decimal to binary and vice versa

Successive Division by 2

$$\begin{array}{r}
 2 \overline{) 29} \\
 \underline{2 \overline{) 14}} \\
 2 \overline{) 7} \\
 \underline{2 \overline{) 3}} \\
 2 \overline{) 1} \\
 \underline{ 0}
 \end{array}$$

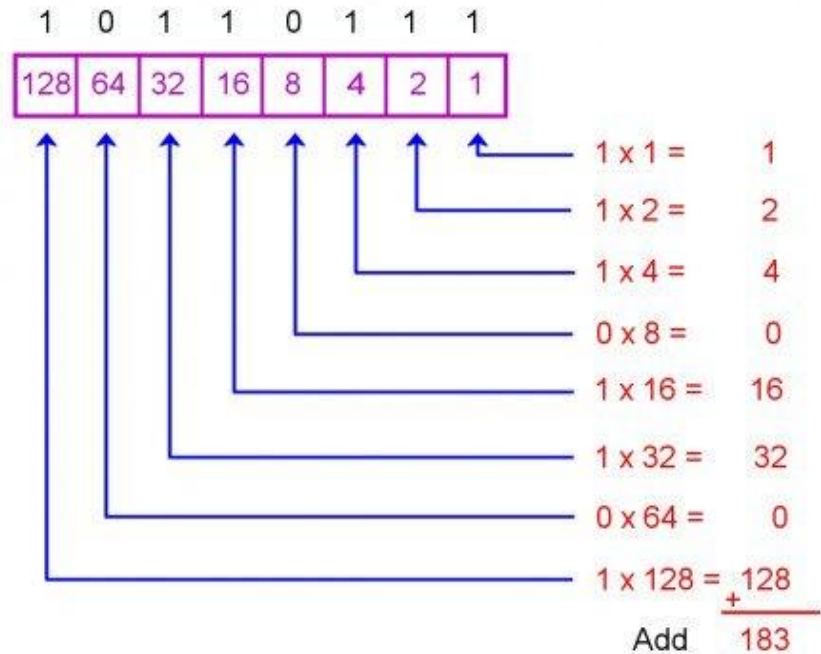
Remainders

1 **LSB**
 0
 1
 1
 1 **MSB**

Read the remainders
 from the bottom up

29 decimal = 11101 binary

Convert 10110111 to Decimal



10110111 = 183 decimal

Exercise - 1

- Using a genetic algorithm program to find the allocations X_i that maximize the total benefits to the three water users i along a stream, whose individual benefits are
 - User 1: $6X_1 - X_1^2$
 - User 2: $7X_2 - X_2^2$
 - User 3: $8X_3 - X_3^2$
- Assume the available stream flow is some known value of 8.

Exercise - 2

- Consider the stream pollution problem where the stream receives waste, W_i from sources located at sites $i = 1$ and 2 .
- Without some waste treatment at these sites, the pollutant concentrations at sites 2 and 3 will exceed the maximum desired concentration.
- The problem is to find the fraction of wastewater removal, x_i , at sites $i = 1$ and 2 required to meet the quality standards at sites 2 and 3 at a minimum total cost.
- The data used for the problem shown in next slide

Exercise - 2

	Parameter	Unit	Value	Remark
Flow	Q_1	m^3/s	10	Flow just u/s of site 1
	Q_2	m^3/s	12	Flow just u/s of site 2
	Q_3	m^3/s	13	Flow at park
Waste	W_1	Kg/day	250,000	Pollutant mass produced at site 1
	W_2	Kg/day	80,000	Pollutant mass produced at site 2
Pollutant Concentration	P_1	mg/l	32	Concentration Just upstream of site 1
				Maximum Allowable concentration upstream of site 2
	P_3	mg/l	20	Maximum Allowable concentration at site 3
Decay Fraction	σ_{12}	-	0.25	Fraction of site 1 pollutant mass at site 2
	σ_{13}	-	0.15	Fraction of site 1 pollutant mass at site 3
	σ_{23}	-	0.6	Fraction of site 2 pollutant mass at site 3