INTRODUCTION TO GENETIC ALGORITHM (GA)

Preamble

- A non-traditional optimization method.
- Stochastic in nature with probabilistic transition rules
- Traditional optimization method is deterministic in nature with specific rule for moving from one solution to the other.
- It is a computerized search and optimization algorithm that mimics natural evolution and genetics.
- It revolves around the genetic reproduction processes and 'survival of the fittest' strategies.
- Uses population of solutions and parallel processing.
- Recently the algorithms are applied to many engineering design optimization problems
- Suitable for complex optimization problems.

The origin of GA

- The concept of these algorithms evolved in the last century mid sixties by Prof. John Holland, University of Michigan, Ann Arbor.

Working Principle

- Genetic algorithms follow the natural order of attaining the maximum.
- Consider the following maximization problem:

Maximize $$f(x), \quad x_i^{(L)} \le x_i \le x_i^{(U)}, i = 1, 2, \ldots, N$$

- Minimization problem can also be handled using GA.

Coding and Decoding

- To use GA, the variables x_i's are first coded in string structure (chromosome).
- Generally variable x_i is represented as a binary number of say l_i bits.
- This type of coding will divide the feasible interval of variable x_i into $2^{l_i} - 1$ intervals.
- The length of string is usually determined according to the desired solution accuracy.

- The obtainable accuracy for a variable x_i with l_i bit coding is $\left(x_i^{(U)} - x_i^{(L)}\right) / 2^{l_i}$.
- For a two-variable function a coded string may be like 1011 0011.
- The first four bits represents x_1 and the remaining four represents x_2 .
- The decoded value of sub string representing x_1 is $((1)2^0 + (1)2^1 + (0)2^2 + (1)2^3)$ or 11
- A variable is in coded string form whose value can be found out by a fixed mapping rule.
- A linear mapping rule is

$$x_i = x_i^{(L)} + \frac{x_i^{(U)} - x_i^{(L)}}{2^{l_i} - 1} \times (\text{decoded value of substring for the variable } x_i)$$

Fitness Function

- GAs mimics the survival-of-the-fittest principle of nature to make a search process.
- Therefore, GAs is naturally suitable for solving maximization problems.
- The operation of GAs begins with a population of strings representing design or decision variables.
- A chromosome's potential as a solution is determined by its fitness function, which evaluates a chromosome with respect to the objective function of the optimization problem.
- Fitness function (T_x) is first derived from objective function and used in successive genetic operations.
- For maximization problems, the fitness function can be considered to be the same as the objective function.
- For minimization problems, the fitness function is an equivalent maximization problem chosen such that the optimum point remains unchanged.
- One such transformation is

$$T_x = 1 / (1 + f(x)) \ .$$

GA Operators - Introduction

- Three main operators are reproduction, crossover and mutation.
- They operate upon the population of solution to create a new population of points.
- The new population is further evaluated and tested for termination.
- The termination criterion is not met; the population is iteratively operated by the operators and evaluated.
- This procedure is continued until the termination criterion is met.
- One cycle of these operations and the subsequent evaluation procedure is known as a generation.

Reproduction

- This operation by which good strings in a population are selected and form a mating pool.
- The essential idea of a reproduction operator is that the above-average strings are picked from the current population and their multiple copies are inserted in the mating pool in probabilistic manner.
- The commonly used reproduction operator (roulette-wheel selection) is the proportionate reproduction operator where a string is selected for the mating pool with a probability proportionate to its fitness.
- Probability of selection for a population of size *n* is

$$p_i = \frac{T_i}{\sum_{i=1}^{n} T_i}$$

- The roulette-wheel selection can be simulated in the following manner.

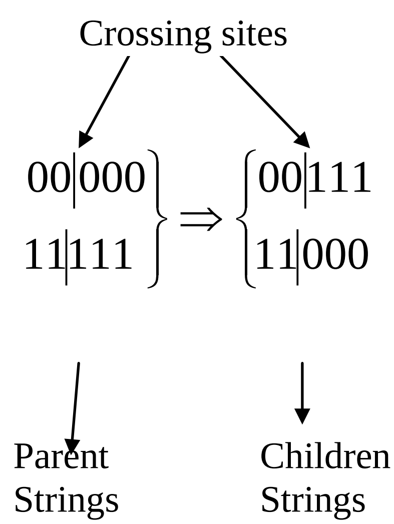
  Consider 5 strings with fitness shown in column 2 of the table below:

String Number	Fitness	Cumulative Fitness	P_i	Cumulative P_i	Random Number	Mating Pool
1	25	25	0.25	0.25	0.9501	5
2	5	30	0.05	0.30	0.2311	1
3	40	70	0.40	0.70	0.6068	3
4	10	80	0.10	0.80	0.4860	3
5	20	100	0.20	1.00	0.8913	5

- Large number of copies of good strings in a population is probabilistically assigned to form a mating pool.

Crossover

- This operator is mainly responsible for the search of new strings.
- New strings are created by exchanging information among strings of the mating pool.
- In most crossover operators, two strings are picked from the mating pool at random and some portions of the strings are exchanged between the strings.
- A single-point crossover operator is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the crossing site.



- In order to preserve some of the good strings that are already present in the mating pool, not all strings in the mating pool are used in crossover.
- Crossover is carried out with a probability, p_c

Mutation

- It also generates new strings but it is used sparingly.
- Mutation operator changes 1 to 0 and vice versa with a small probability, p_m .
- Mutation is used to maintain diversity in the population.

For example, consider the following population having four eight-bit strings, which undergo mutation with $p_m = 0.05$:

Strings	Random number	Mutated string
0110 1011	0.7621 0.4447 0.7382 0.9169 0.3529 0.2028 0.1988 0.9318	0110 1011
0011 1101	0.4565 0.6154 0.1763 0.4103 0.8132 0.1987 0.0153 0.4660	0011 1111
0001 0110	0.0185 0.7919 0.4057 0.8936 0.0099 0.6038 0.7468 0.4186	1001 1110
0111 1100	0.8214 0.9218 0.9355 0.0579 0.1389 0.2722 0.4451 0.8462	0111 1100

- Notice that all four strings have a 0 in the left-most bit position.
- If the true optimum solution requires 1 in that position, then neither reproduction nor crossover operator will be able to create 1 in that position.

Algorithm

Step 1. Choose a coding to represent problem parameters, a selection operator, a crossover operator, and a mutation operator.

Choose population size, *n*, crossover probability, p_c, and mutation probability, p_m.

Initialise a random population of strings of size *l*.

Set a maximum allowable generation number $t_{\max}$. Set $t = 0$.

Step 2. Evaluate each string in the population.

Step 3. If $t > t_{\max}$ or other termination criteria is satisfied, terminate.

Step 4. Perform reproduction on the population.

Step 5. Perform crossover on random pairs of strings.

Step 6. Perform mutation on every string.

Step 7. Evaluate strings in the new population. Set $t = t + 1$ and go to step 3.

Exercise Problem

- Minimize $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2, \quad 0 \le x_1, x_2 \le 6$

  - Himmelblau function

  Note: - The true solution to this problem is (3, 2) having a function value equal to zero.

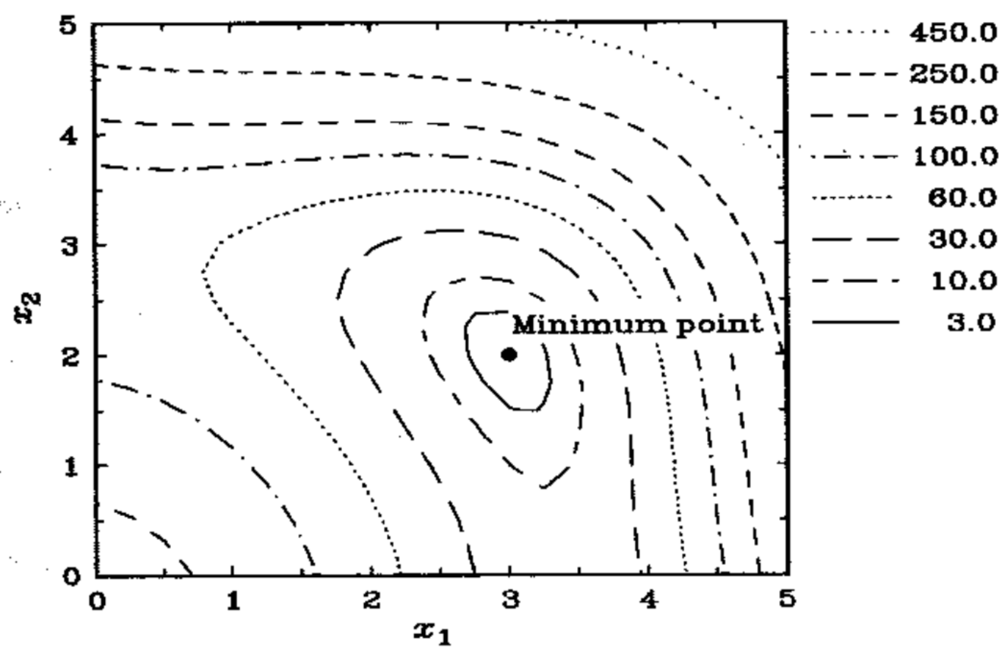


Fig.1 Contour plot of the Himmelblau function

- The function value corresponding to each contour is listed in the right side

Step 1. Coding details: Binary coding with 10 bits for each variable and thus total string length=20.

Operators: Roulette-wheel selection, Single point crossover and bit wise mutation.

$$n = 20,\ p_m = 0.05,\ p_c = 0.8, t_{\max} = 30$$

Initialise the generation counter, $t = 0$.

Table 1 Evaluation and Reproduction Phases on a Random Population

	String												Mating pool	
	Substring-2	Substring-1	x_2	x_1	$f(x)$	$T(x)$	A	B	C	D	E	F	Substring-2	Substring-1
1	1110010000	1100100000	5.349	4.692	959.680	0.001	0.13	0.007	0.007	0.472	10	0	0010100100	1010101010
2	0001001101	0011100111	0.452	1.355	105.520	0.009	1.10	0.055	0.062	0.108	3	1	1010100001	0111001000
3	1010100001	0111001000	3.947	2.674	126.685	0.008	0.98	0.049	0.111	0.045	2	1	0001001101	0011100111
4	1001000110	1000010100	3.413	3.120	65.026	0.015	1.85	0.093	0.204	0.723	14	2	1110011011	0111000010
5	1100011000	1011100011	4.645	4.334	512.197	0.002	0.25	0.013	0.217	0.536	10	0	0010100100	1010101010
6	0011100101	0011111000	1.343	1.455	70.868	0.014	1.71	0.086	0.303	0.931	19	2	0011100010	1011000011
7	0101011011	0000000111	2.035	0.041	88.273	0.011	1.34	0.067	0.370	0.972	19	1	0011100010	1011000011
8	1110101000	1110101011	5.490	5.507	1436.563	0.001	0.12	0.006	0.376	0.817	17	0	0111000010	1011000110
9	1001111101	1011100111	3.736	4.358	265.556	0.004	0.49	0.025	0.401	0.363	7	1	0101011011	0000000111
10	0010100100	1010101010	0.962	4.000	39.849	0.024	2.96	0.148	0.549	0.189	4	3	1001000110	1000010100
11	1111101001	0001110100	5.871	0.680	814.117	0.001	0.14	0.007	0.556	0.220	6	0	0011100101	0011111000
12	0000111101	0110011101	0.358	2.422	42.598	0.023	2.84	0.142	0.698	0.288	6	3	0011100101	0011111000
13	0000111110	1110001101	0.364	5.331	318.746	0.003	0.36	0.018	0.716	0.615	12	1	0000111101	0110011101
14	1110011011	0111000010	5.413	2.639	624.164	0.002	0.24	0.012	0.728	0.712	13	1	0000111110	1110001101
15	1010111010	1010111000	4.094	4.082	286.800	0.003	0.37	0.019	0.747	0.607	12	0	0000111101	0110011101
16	0100011111	1100111000	1.683	4.833	197.556	0.005	0.61	0.030	0.777	0.192	4	0	1001000110	1000010100
17	0111000010	1011000110	2.639	4.164	97.699	0.010	1.22	0.060	0.837	0.386	9	1	1001111101	1011100111
18	1010010100	0100001001	3.871	1.554	113.201	0.009	1.09	0.054	0.891	0.872	18	1	1010010100	0100001001
19	0011100010	1011000011	1.326	4.147	57.753	0.017	2.08	0.103	0.994	0.589	12	2	0000111101	0110011101
20	1011100011	1111010000	4.334	5.724	987.955	0.001	0.13	0.006	1.000	0.413	10	0	0010100100	1010101010

A : Expected count
B : Probability of selection
C : Cumulative probability of selection
D : Random number between 0 and 1
E : String number
F : True count in the mating pool

- Initial population and mating pool on a contour plot of the objective function
- Best point in the population has a function value = 39.849
- Average function value of initial population = 360.54

Step 2. Evaluation, see Table 1. Average fitness is 0.008

Step 3. Since $t = 0 < t_{\max} = 30$, proceed to step 4.

Step 4. Reproduction, see Table 1. The mating pool is filled in random order (as per the random number).

Step 5. Crossover operation, see Table 2. The expected number of crossover in this case is $0.8 \times 20/2 = 8$.

Actual crossover performed =7

Step 6. Mutation operation, see Table 2. Expected number of bits altered is $0.05 \times 20 \times 20 = 20$.

Actually altered is 16 bits.

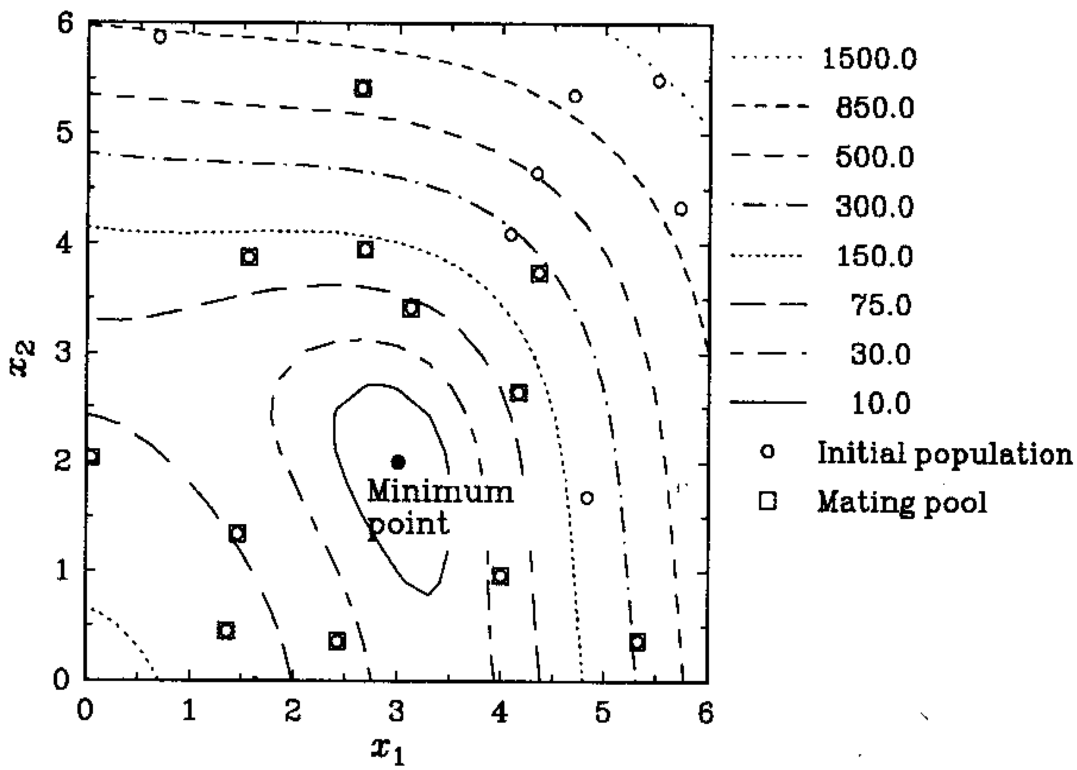


Fig. 2 Initial Population and Mating Pool

Table 2 Crossover and Mutation Operators

Mating pool				Intermediate population		Mutation					
Substring-2	Substring-1	G	H	Substring-2	Substring-1	Substring-2	Substring-1	x_1	x_2	$f(x)$	$T(x)$
0010100100	1010101010	Y	9	0010100101	0111001000	001010$\boxed{1}$101	0111001000	1.015	2.674	18.886	0.050
1010100001	0111001000	Y	9	1010100000	1010101010	101010000$\boxed{1}$	1010101010	3.947	4.000	238.322	0.004
0001001101	0011100111	Y	12	0001001101	0011000010	0001001101	00$\boxed{0}$1000010	0.452	0.387	149.204	0.007
1110011011	0111000010	Y	12	1110011011	0111100111	1110011011	01$\boxed{0}$1100$\boxed{0}$11	5.413	2.082	596.340	0.002
0010100100	1010101010	Y	5	0010100010	1011000011	0010100010	1011000011	0.950	4.147	54.851	0.018
0011100010	1011000011	Y	5	0011100100	1010101010	0011100100	1$\boxed{1}$10101010	1.337	5.501	424.583	0.002
0011100010	1011000011	N		0011100010	1011000011	001110001$\boxed{1}$	1011$\boxed{1}$00011	1.331	4.334	83.929	0.012
0111000010	1011000110	N		0111000010	1011000110	01$\boxed{0}$10$\boxed{1}$0010	1011000110	1.982	4.164	70.472	0.014
0101011011	0000000111	Y	14	0101011011	0000010100	0101011011	0000010100	2.035	0.117	87.633	0.011
1001000110	1000010100	Y	14	1001000110	1000000111	10010$\boxed{1}$0110	1000000111	3.507	3.044	72.789	0.014
0011100101	0011111000	Y	1	0011100101	0011111000	0011100101	0011111000	1.343	1.455	70.868	0.014
0011100101	0011111000	Y	1	0011100101	0011111000	0011100101	0011111000	1.343	1.455	70.868	0.014
0000111101	0110011101	N		0000111101	0110011101	00001$\boxed{0}$1101	011$\boxed{1}$01110$\boxed{0}$	0.264	2.792	25.783	0.037
0000111110	1110001101	N		0000111110	1110001101	0000111110	1110001101	0.364	5.331	318.746	0.003
0000111101	0110011101	Y	18	0000111101	0110011100	0000111101	0110011100	0.358	2.416	42.922	0.023
1001000110	1000010100	Y	18	1001000110	1000010101	1001000110	$\boxed{0}$000010101	3.413	0.123	80.127	0.012
1001111101	1011100111	Y	10	1001111101	0100001001	1001111101	0100001001	3.736	1.554	95.968	0.010
1010010100	0100001001	Y	10	1010010100	1011100111	1010010100	101$\boxed{0}$100111	3.871	3.982	219.426	0.005
0000111101	0110011101	N		0000111101	0110011101	0000111101	0110011101	0.358	2.422	42.598	0.023
0010100100	1010101010	N		0010100100	1010101010	0010100100	1010101010	0.962	4.000	39.849	0.024

G : Whether crossover (Y yes, N no), H : Crossing site

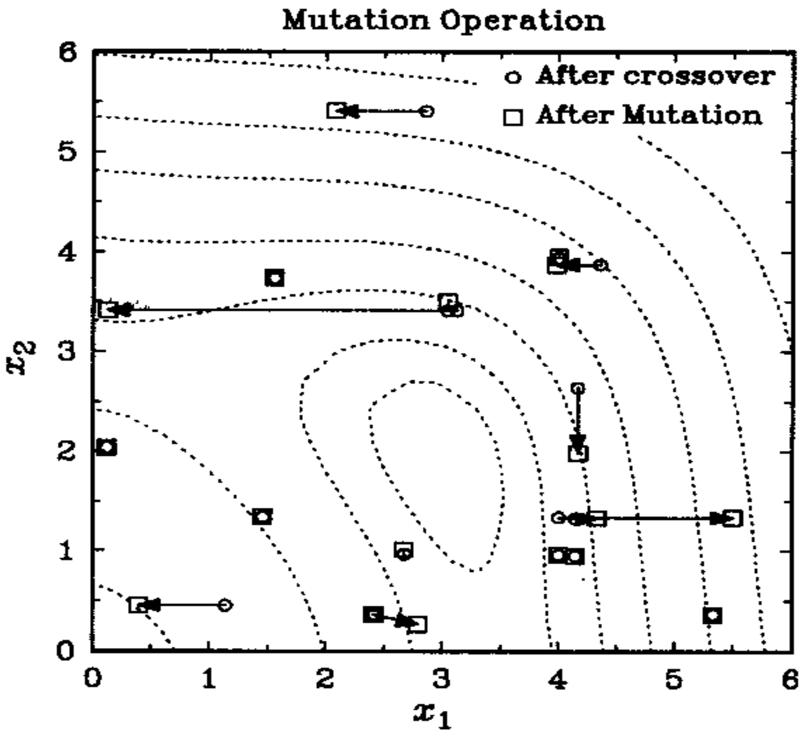


Fig. 3. Population after Mutation Operation

- Some points do not get mutated and remain unaltered
- Current best point has a function value = 18.886
- Average function value = 140.210
- Improvement – over 60 %

Step 7 Resulting population becomes the new population. Generation number 1.

Average fitness is 0.015, improved remarkably from initial average of 0.008.

The best point in this population is found to have fitness equal to 0.05, which is greater than the best (0.024) in the initial population. After 25 generations, the best point is found to be (3.003, 1.994) with a function value 0.001. The fitness value at this point is equal to 0.999 and the average population fitness value is 0.474.

- Most points are clustered around the true minimum

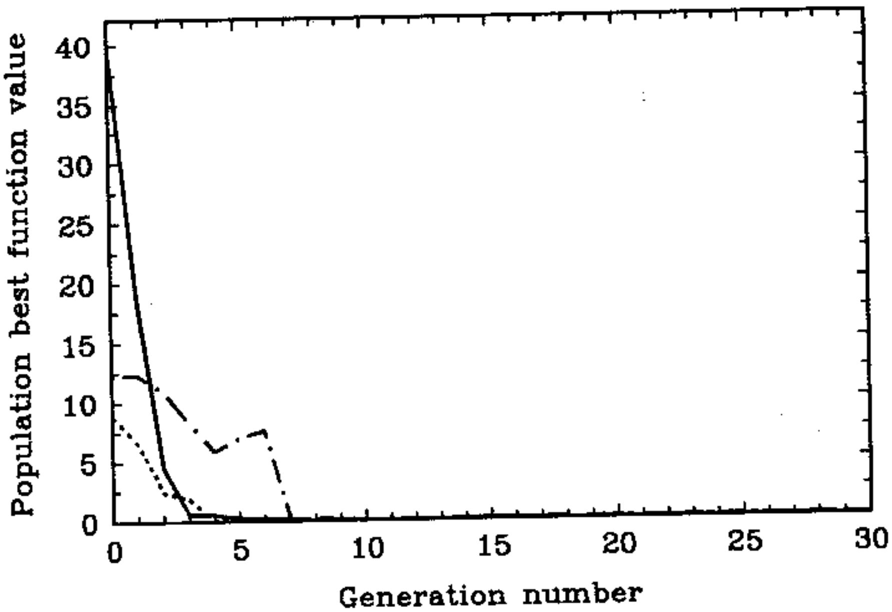


Fig. 5 Three Independent GA Runs

- Fig. 5 shows the function value of the best point in the population for three independent GA runs
- All runs are quickly converge to a point near the optimum.

Conclusions

- GA searches are robust that allows them to be applied to a wide variety of problems
- Suitable for multimodal functions
- Final solution of a GA may be optimum or near optimum

Bibliography

•Belegundu, A. D., and Chandrupatla, T. R., Optimisation Concepts and Applications in Engineering, 1999, Pearson Education (Singapore) Pte. Ltd.

•Goldberg, D. E., Genetic Algorithms in Search Optimisation & Machine Learning, 2000, Pearson Education Asia Pte. Ltd.

•Kalyanmoy Deb, Optimisation for Engineering Design: Algorithms and Examples, 1995, Prentice-Hall of India Pvt. Ltd.

•Singh, N., and Rajamani, D., Cellular Manufacturing Systems: Design, Planning and Control, 1996, Chapman & Hall, London.

QUESTIONS:

1. What is fitness function? How is it developed? Write the commonly used transformations to get fitness function from objective function of minimization problems.
2. What is the use of fitness function in a genetic algorithm? How is this function formed from objective function of a problem?
3. What are the parameters of a genetic algorithm? How does a genetic algorithm initialized?
4. What is the objective of reproduction operation? Describe the proportionate reproduction operator (roulette-wheel selection).
5. Explain the selection and reproduction stage of genetic algorithm.
6. Write down the steps involved in a genetic algorithm.
7. A population of eight chromosomes is used in a genetic algorithm. The fitness value of these chromosomes is given in a Table below. Determine the mating pool using roulette-wheel selection. Take necessary random numbers from the Table given below.

Table: Fitness value of chromosomes

Chromosome No.	1	2	3	4	5	6	7	8
Fitness function value	4	15	26	17	8	19	16	10

Table: Random numbers

	1	2	3	4	5	6	7	8	9	10
Random No.	0.9501	0.2311	0.6068	0.4860	0.8913	0.7621	0.4565	0.0185	0.8214	0.4447

8. A Genetic Algorithm (GA) is applied to a minimization problem. A population of nine chromosomes is used in the GA. The objective function value of these chromosomes is given in the Table. Determine the mating pool using roulette-wheel selection. Take necessary random numbers from the Table given below.

Table: Objective function value of chromosomes

Chromosome No.	1	2	3	4	5	6	7	8	9
Objective function value	15	4	26	8	17	19	16	10	14

Table: Random numbers

	1	2	3	4	5	6	7	8	9	10
Random No.	0.9501	0.2311	0.6068	0.4860	0.8913	0.7621	0.4565	0.0185	0.8214	0.4447

9. A Cellular manufacturing system design problem uses genetic algorithm as a solution procedure to the mathematical program and this mathematical model has a minimisation function. The objective function value for the chromosomes generated for a population size of 10 is given below.

Table: Objective function value

Chromosome No.	1	2	3	4	5	6	7	8	9	10
Costs	101800	104350	102150	110500	87350	104500	116250	90400	95300	107250

Identify the chromosomes for the mating pool using the proportionate reproduction operator. Use the random numbers given below. (Use the consecutive random numbers.)

Table: Random Numbers

	1	2	3	4	5	6	7	8	9	10
Random No.	0.9501	0.2311	0.6068	0.4860	0.8913	0.7621	0.4565	0.0185	0.8214	0.4447
	11	12	13	14	15	16	17	18	19	20
Random No.	0.6154	0.7919	0.9218	0.7382	0.1763	0.4057	0.9355	0.9169	0.4103	0.8936