[Volume 2 issue 6 June 2014]
Page No.462-468 ISSN :2320-7167

INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH

How Genetic Algorithm Is Performed In A Single Generation.

Dipanjan kumar Dey
Department of Computer Science & Engineering,
Mail id-dipanjandey75@gmail.com
Assistant professor of Prajnanananda Institute of Technology and Management (PITM)
94/2, Park Street, Dist-Kolkata, Kolkata 700017, West Bengal (India)

Abstract: First part of this work consists of basic information about Genetic algorithm like what are Individual, Population, Crossover, Genes, Binary Encoding, Flipping, Crossover probability Mutation probability**.** What is it used for, what is their aim. A genetic algorithm is one of a class of algorithms that searches a solution space for the optimal solution to a problem. In this article the methods of selection, crossover and mutation are specified. In the second part, solving a maximizing problem using Genetic algorithm in a single generation. A single generation of a Genetic algorithm is performed here with encoding, selection, crossover and mutation.

.**Keywords**: selection, crossover, mutation

I.INTRODUCTION:

Chromosomes are selected from the population to be parents to crossover. The problem is how to select these chromosomes. Genetic algorithm is based on the Darwin's theory of evolutions; the basic rule is "survival of the fittness."I.e According to Darwin's evolution theory the best ones should survive and create new offspring. There are many methods how to select the best chromosomes, for example roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and some others. Here I am solving a problem to select these chromosomes the best ones should survive and create new offspring using Genetic algorithm (GA).

II. INDIVIDUALS:

An individual is a single solution. Each individual has fitness. An individual is encoded as a string of binary digits.

III. GENES:

Genes are the basic instructions for building a GA.A chromosome is a sequence of genes.

IV. POPULATION:

A population is a collection of individuals. Population being a combination of various chromosomes.

Population	Chromosome 1	0 1 1 0 0
	Chromosome 2	1 1 0 0 1
	Chromosome 3	0 0 1 0 1
	Chromosome 4	1 0 0 1 1

This figure shows the population consists of four chromosomes each having five bits.

V. BINARY ENCODING:

In binary encoding, every chromosome encodes a binary string of bits: 0 or 1are mostly used
, like
Chromosome 1: 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 1
Chromosome 2: 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1

VI. SELECTION.

Selection allocates more copies of those solutions with higher fitness values. If the fitness function is higher the better chance that a individual will be selected. The main idea of selection is to prefer better solutions to worse ones.

VII. CROSSOVER

Crossover selects genes from parent chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover selects genes from parent chromosomes and creates a new offspring.

The Crossover operators are of many types.
−one simple way is, One-Point crossover or single-point crossover
. The others are Two Point crossover, Uniform crossover, Arithmetic crossover, precedence preservative crossover (PPX), partially matched crossover (PMX) and Heuristic crossovers.
Here I am discussing only One-Point crossover or single-point crossover because in my work only it is required.
Crossover can then look like this (| is the crossover point):

Chromosome 1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

VII.I. ONE-POINT CROSSOVER

One-Point crossover operator randomly selects one crossover point and then copy everything before this point from the first parent and then everything after the crossover point copy from the second parent. The Crossover would then look as shown below.
Consider the two parents selected for crossover.
Parent 1 1 1 0 1 1 |0 0 1 0 0 1 1 0 1 1 0
Parent 2 1 1 0 1 1 |1 1 0 0 0 0 1 1 1 1 0
Interchanging the parents chromosomes after the crossover points -
The Offspring produced are:
Offspring 1 1 1 0 1 1|1 1 0 0 0 0 1 1 1 1 0
Offspring 2 1 1 0 1 1|0 0 1 0 0 1 1 0 1 1 0
Note: The symbol, a vertical line, |is the chosen crossover point

VIII. MUTATION

After a crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can then be following:

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

The mutation depends on the encoding as well as the crossover. For example when we are encoding permutations, mutation could be exchanging two genes.

IX. CROSSOVER AND MUTATION PROBABILITY

There are two basic parameters of GA - crossover probability and mutation probability.

IX.I. CROSSOVER PROBABILITY (P_C)

Crossover probability (P_c) says how often will be crossover performed. If there is no crossover, offspring is exact copy of parents. If there is a crossover, offspring is made from parts of parents' chromosome.

If crossover probability **(P_c)** is **100%**, then all offspring is made by crossover.

If crossover probability **(P_c)** is **0%**, whole new generation is made from exact copies of chromosomes from old population (but this does not mean that the new generation is the same).

Crossover is made in hope that new chromosomes will have good parts of old chromosomes and maybe the new chromosomes will be better. However it is good to leave some part of population survive to next generation.

IX.II. MUTATION PROBABILITY (P_M)

Mutation probability (P_m) says how often will be parts of chromosome mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. If **(P_m)** is **100%**, whole chromosome is changed, if **(P_m)** is **0%**, nothing is changed.
Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to **random search**.

X. FLIPPING:

Flipping of a bit involves changing 0 to 1 and 1 to 0 based on a mutation chromosome generated.
Consider a parent and a mutation chromosome is randomly generated. For a 1 in mutation chromosome, the corresponding bit in parent chromosome is flipped (0 to1and 1to 0).In the following table 1 occurs at 3 places of mutation chromosome, the corresponding bits in parent chromosome are flipped and the child is generated.

Parent	1011 0101
Mutation chromosome	1000 1001
Child	0011 1100

Figure-Mutation flipping concept

XI. SOLVING A MAXIMIZING PROBLEM USING GENETIC ALGORITHM:

Let us consider a maximizing problem,
The objective function $f(x) = x^2$, which is to be maximized, where x can take values 0 and 31.x is also known as decision variable. $f(x) = x^2$ is also known as fitness function.
Here I am using five bits (binary integer) numbers between 0(00000) and 31(11111).
A single generation of a Genetic algorithm is performed here with encoding, selection, crossover and mutation.
Here initial population of size 4 is randomly chosen (01100, 11001, 00101, 10011).Note that any number of populations can be selected according to the requirement and application.

XI.I TABLE-1: THE PRESENTATION OF SELECTION:

String no.	Initial population (randomly selected)	Value of the variable x	fitness function $f(x) = x^2$	Probability (p_i) of selection	% Probability of selection	Expected count	Actual count
S_1	01100	12	$144=f(x_1)$	$0.1247=P_1$	12.47%	0.4987	1
S_2	11001	25	$625= f(x_2)$	$0.5411= P_2$	54.11%	2.1645	2
S_3	00101	5	$25= f(x_3)$	$0.0216= P_3$	2.16%	0.0866	0
S_4	10011	19	$361= f(x_4)$	$0.3126= P_4$	31.26%	1.2502	1
Sum			$1155= \sum f(x_i)$	$1.0000= \sum p_i$	100	4.0000	4
Average			288.75	0.2500	25	1.0000	1
Maximum			625	0.5411	54.11	2.1645	2

Calculations:
For string S_1 01100, the x value of $01100=0*2^4+1*2^3+1*2^2+0*2^1+0*2^0=0+8+4+0+0=12$
For string S_2 11001, the x value of $11001=1*2^4+1*2^3+0*2^2+0*2^1+1*2^0=16+8+0+0+1=25$
For string S_3 00101, the x value of $00101=0*2^4+0*2^3+1*2^2+0*2^1+1*2^0=0+0+4+0+1=5$
For string S_4 10011, the x value of $10011=1*2^4+0*2^3+0*2^2+1*2^1+1*2^0=16+0+0+2+1=19$

Now for fitness function $f(x) = x^2$, Calculate the fitness value,
For string S_1, x=12, $f(x) = x^2 = (12)^2 = 144$
For string S_2, x=25, $f(x) = x^2 = (25)^2 = 625$
For string S_3, x=5, $f(x) = x^2 = (5)^2 = 25$
For string S_4, x=19, $f(x) = x^2 = (19)^2 = 361$

Sum of the fitness value=$\sum f(x_i) = 144+625+25+361=1155$

Probability of selection:

For string_S$_1$, Probability $P_1= \frac{f(x_1)}{\sum f(x_i)} = \frac{144}{1155}$ =0.1247, ∴% Probability= 0.1247*100=12.47

For string_S$_2$, Probability $P_2= \frac{f(x_1)}{\sum f(x_i)} = \frac{625}{1155}$ =0.5411, ∴% Probability=0.5411*100=54.11

For string_S$_3$, Probability $P_3= \frac{f(x_1)}{\sum f(x_i)} = \frac{25}{1155}$ =0.0216, ∴% Probability=0.0216*100=2.16

For string_S$_4$, Probability $P_4= \frac{f(x_1)}{\sum f(x_i)} = \frac{361}{1155}$ =0.3126, ∴% Probability=0.3126*100=31.26

∴ $\sum p_i = P_1+ P_2+ P_3+ P_4$=0.1247+0.5411+0.0216+0.3126=1.0000

Average of $f(x_i) = \frac{\sum f(x_i)}{N} = \frac{1155}{4}$=288.75, N=4=number of population,

Expected count $=\frac{Fitness}{Average}$

The expected count gives an idea of which population can be selected for further processing in the mating pool.

For string_S$_1$, Expected count $=\frac{Fitness}{Average}=\frac{f(x_1)}{288.75}=\frac{144}{288.75}$=0.4987

For string_S$_2$, Expected count $=\frac{Fitness}{Average}=\frac{f(x_2)}{288.75}=\frac{625}{288.75}$=2.1645

For string_S$_3$, Expected count $=\frac{Fitness}{Average}=\frac{f(x_3)}{288.75}=\frac{25}{288.75}$=0.0866

For string_S$_4$, Expected count $=\frac{Fitness}{Average}=\frac{f(x_4)}{288.75}=\frac{361}{288.75}$=1.2502

Sum of the expected count =0.4987+2.1645+0.0866+1.2502=4.0000

Thus we see from table-1,

For string_S$_1$, Probability of selection is 12.47%, Expected count=2.1645, so there is a chance for it to participate in the crossover cycle is at least once. Hence it actual count can be consider as 1.

For string_S$_2$, Probability of selection is 54.11%, Expected count=0.4987, so there is a fair chance for it to participate in the crossover cycle twice. Hence it actual count can be consider as 2.

For string_S$_3$, Probability of selection is 2.16%, Expected count=0.0866, so there is a very poor chance for it to participate in the crossover cycle. Hence it actual count can be consider as 0.

For string_S$_4$, Probability of selection is 31.26%, Expected count=1.2502, so there is a chance for it to participate in the crossover cycle is once. Hence it actual count can be consider as 1.

XI.II TABLE-2: THE PRESENTATION OF THE CROSSOVER:

String no.	Mating pool	Crossover point	Offspring after crossover	X value	fitness function $f(x) = x^2$
S_1	01100	4	01101	13	169
S_2	11001	4	11000	24	576
S_3	11001	2	11011	27	729
S_4	10011	2	10001	17	289
Sum					1763
Average					440.75
Maximum					729

Explanation of table-2

The mating pool in table-2 is formed on the basis of actual count.

The actual count of string_S_1 is 1; hence string_S_1 occurs once in mating pool.

The actual count of string_S_2 is 2; hence string_S_2 occurs twice in mating pool.

The actual count of string_S_3 is 0; hence string_S_3 does not occur in mating pool.

The actual count of string_S_4 is 1; hence string_S_4 occurs once in mating pool.

Now Crossover point is specified .On the basis of crossover point, a single- point crossover is performed and new offspring (children) is produced.

Thus a single- point crossover

Parent 1 0 1 1|0 0
Parent 2 1 1 0|0 1

Offspring 1 0 1 1 0 1
Offspring 2 1 1 0 0 0

And

Parent 1 1 1| 0 0 1
Parent 2 1 0| 0 1 1

Offspring 1 1 1 0 1 1
Offspring 2 1 0 0 0 1
Hence after a single- point crossover new offspring (children) are produced. Now "x" values are decoded as follows

For string_S_1 01101, the x value of $01101=0*2^4+1*2^3+1*2^2+0*2^1+1*2^0=0+8+4+0+1=13$
For string_S_2 11000, the x value of $11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0=16+8+0+0+0=24$
For string_S_3 11011, the x value of $11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=16+8+0+2+1=27$
For string_S_4 10001, the x value of $10001=1*2^4+0*2^3+0*2^2+0*2^1+1*2^0=16+0+0+0+1=17$

Now for fitness function $f(x) = x^2$, Calculate the fitness value,
For string_S_1, x=13, $f(x) = x^2 = (13)^2 = 169$
For string_S_2, x=24, $f(x) = x^2 = (24)^2 = 576$
For string_S_3, x=27, $f(x) = x^2 = (27)^2 = 729$
For string_S_4, x=17, $f(x) = x^2 = (17)^2 = 289$
Sum of the fitness value=$\sum f(x_i)$ =169+576+729+289=1763
Average of the fitness value=$\frac{\text{Sum of the fitness value}}{4} = \frac{1763}{4}$ =440.75

XI.III. TABLE-3: THE PRESENTATION OF THE MUTATION:

String no.	Offspring (children) after crossover	Mutation Chromosome for flipping	Offspring (children) after mutation	X value	Fitness function $f(x) = x^2$
S_1	01101	10000	11101	29	841
S_2	11000	00000	11000	24	576
S_3	11011	00000	11011	27	729
S_4	10001	00100	10101	21	441
Sum					2587
Average					646.75
Maximum					841

Explanation of table-3

After crossover operation, mutation operation is performed and new offspring (children) are produced. I have discussed mutation flipping concept in section 11.Now mutation flipping operation is performed and offspring (children) are produced.

Parent	01101	11000	11011	10001
Mutation chromosome for flipping	10000	00000	00000	00100

New children(offspring)	11101 11000 11011 10101

Hence after mutation new offspring (children) are produced. Now "x" values are decoded as follows

For string S_1 11101, the x value of $11101=1*2^4+1*2^3+1*2^2+0*2^1+1*2^0=16+8+4+0+1=29$
For string S_2 11000, the x value of $11000=1*2^4+1*2^3+0*2^2+0*2^1+0*2^0=16+8+0+0+0=24$
For string S_3 11011, the x value of $11011=1*2^4+1*2^3+0*2^2+1*2^1+1*2^0=16+8+0+2+1=27$
For string S_4 10101, the x value of $10101=1*2^4+0*2^3+1*2^2+0*2^1+1*2^0=16+0+4+0+1=21$
Now for fitness function $f(x) = x^2$, Calculate the fitness value,
For string S_1, x=29, $f(x) = x^2 = (29)^2 = 841$
For string S_2, x=24, $f(x) = x^2 = (24)^2 = 576$
For string S_3, x=27, $f(x) = x^2 = (27)^2 = 729$
For string S_4, x=21, $f(x) = x^2 = (21)^2 = 441$
Sum of the fitness value=$\sum f(x_i) = 841+576+729+441=2587$
Average of the fitness value=$\frac{\text{Sum of the fitness value}}{4} = \frac{2587}{4} = 646.75$

.

XII. **CONCLUSION:**

Once selection, crossover and mutation are performed the new population is now ready to be tested. In table-1 the expected count gives an idea of which population can be selected for further processing in the mating pool. The actual count gives an idea to select the individuals who would participate in the crossover cycle. From table-1, table-2, table-3 we observed that how maximum fitness and the population average fitness performances have improved in the new population. In one generation, the population average fitness has improved from 288.75to 646.75.During the same period the maximum fitness has improved 625 to 841.The best string from initial population (randomly selected)01100,11001,00101,10011 is11001, it receives two chances for its existence because of its high, above-average performances.

Thus after mutation (table-3) a new Offspring (11101) is produced which is an excellent choice.

This completes the genetic algorithm is performed in one (single) generation.

XIII. REFERENCES

[1] – Genetic algorithms for optimization – application in the controller synthesis task – Popov A., diploma thesis, department Systems and Control, faculty Automatics, Technical University Sofia, 2003

[2] – Practical Optimization, Gill Ph., Marry W., Wright M., Academic Press, 1981

[3]- "An Overview of Genetic Algorithms: Part 1, Fundamentals", by David Beasley, David R. Bull, and Ralph R. Martin. University Computing, volume 15(2), pages 58-69, 1993.

[4] Whitley D 1991 Fundamental Principles of Deception in Genetic Search Foundations of Genetic Algorithms G Rawlins ed Morgan Kaufmann

[5] Stark weather, T. Whitley D and Mathias K 1991 Optimization Using Distributed Genetic Algorithms

[6] de Toro, F., Ortega, J., Fernandez, J., and Diaz, A. PSFGA: a parallel genetic algorithm for multiobjective optimization. In Proceedings 10th Euro micro Workshop on Parallel, Distributed and Network-based Processing, 9-11 Jan. 2002. 2002. Canary Islands, Spain: IEEE Comput. Soc.

[7] Fonseca, C.M. and Fleming, P.J. Multiobjective genetic algorithms. in IEE Colloquium on `Genetic Algorithms for Control Systems Engineering' (Digest No. 1993/130), 28 May 1993. 1993. London, UK: IEE]

[8] Goldberg, D. E., 2002, Design of Innovation: Lessons From and For Competent Genetic Algorithms, Kluwer, Boston, MA.

[9] Goldberg, D. E. and Voessner, S., 1999, Optimizing global-local search Hybrids, in: Proc. of the Genetic and Evolutionary Computation Conf.,pp. 220–228.

[10] Krasnogor, N., Hart, W. and Smith, J. (eds), 2004,Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, Vol. 166, Springer,Berlin.

[11]K. Deb and S. Argrawal, "Understanding interactions among genetic algorithm parameters,"in Foundations of Genetic Algorithms 5, 1998, pp. 265-286.D. E. Goldberg,Genetic Algorithms in Search, Optimization & Machine Learning,Addison Wesley, 1989.

[12] D. E. Goldberg,Genetic Algorithms in Search, Optimization & Machine Learning,Addison Wesley, 1989.

[13]A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms,"IEEE Transactions on Evolutionary Computation, Vol. 3, 1999, pp.124-141

[14]K. A. D. Jong, "An analysis of the behavior of a class of genetic adaptive systems,"PhD thesis, University of Michigan, 1975.

[15]T. C. Fogarty, "Varying the probability of mutation in genetic algorithms," in Proceedings of the Third International Conference on Genetic Algorithms, 1989, pp.104-109.

Author: 

Dipanjan Kumar Dey, graduated from Calcutta University, India. M.sc (Mathematics) and M.Tech (Computer Science &Engineering) from M.C.K.V Institute of Engineering (under West Bengal University & Technology, India). He is currently Assistant Professor of Mathematics & Computer Science in Prajnanananda Institute of Technology & Management, West Bengal, India. He is also Faculty member of Institute of Chartered financial Analysis of India (ICFAI) and Academic Counselor, Assistant Coordinator of Indira Gandhi National Open University (IGNOU) study center 2804, Kolkata, India. He is a Science Journalist having Post Graduate certificate course on journalism and media practice from National Council for the Science and Technology Communication, GOVT. OF INDIA, New Delhi.
His research interests in Genetic Algorithms, Soft Computing, Fuzzy Set, Artificial intelligence, Mobile computing.