https://github.com/Marouane20/GeneticAlgo

Genetic Algorithm for Solving Linear and Quadratic Equations

Basic Philosophy

Genetic algorithm developed by Goldberg was inspired by Darwin's theory of evolution which states that the survival of an organism is affected by rule "the strongest species that survives". Darwin also stated that the survival of an organism can be maintained through the process of reproduction, crossover and mutation. Darwin's concept of evolution is then adapted to computational algorithm to find solution to a problem called objective function in natural fashion. A solution generated by genetic algorithm is called a chromosome, while collection of chromosome is referred as a population. A chromosome is composed from genes and its value can be either numerical, binary, symbols or characters depending on the problem want to be solved. These chromosomes will undergo a process called fitness function to measure the suitability of solution generated by GA with problem. Some chromosomes in population will mate through process called crossover thus producing new chromosomes named offspring which its genes composition are the combination of their parent. In a generation, a few chromosomes will also mutation in their gene. The number of chromosomes which will undergo crossover and mutation is controlled by crossover rate and mutation rate value. Chromosome in the population that will maintain for the next generation will be selected based on Darwinian evolution rule, the chromosome which has higher fitness value will have greater probability of being selected again in the next generation. After several generations, the chromosome value will converges to a certain value which is the best solution for the problem.

The Algorithm

In the genetic algorithm process is as follows:

Step 1. Determine the number of chromosomes, generation, and mutation rate and crossover rate value

Step 2. Generate chromosome-chromosome number of the population, and the initialization value of the genes chromosome-chromosome with a random value .

Step 3. Process steps 4-7 until the number of generations is met

Step 4. Evaluation of fitness value of chromosomes by calculating objective function

Step 5. Chromosomes selection

Step 6. Crossover

Step 7. Mutation

Step 8. Solution (Best Chromosomes)

https://github.com/Marouane20/GeneticAlgo

The flowchart of algorithm can be seen in Figure 1

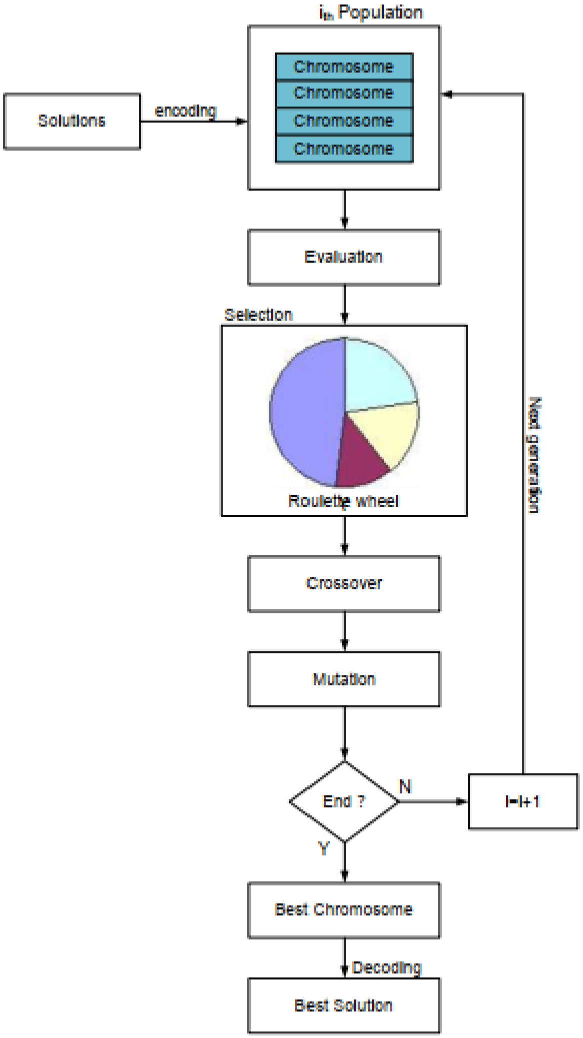

Figure 1. Genetic algorithm flowchart

Numerical Example

Here is example of applications that use genetic algorithms to solve the problem of quadratic equation **$\alpha X^2+\beta X+\lambda$** . Suppose there is equality **$2X^2 + 4X - 4 = 0$**, genetic algorithm will be used to find the value of **X** that satisfy the above equation. First, we should formulate the objective function, but there is a problem which is we have just one value X so we can't do crossover using one value because we need 2 values or more to do the crossover.

To resolve this problem, we generate 2 values or more depend on the input in Init function and we make the average between these values to get X.

For this problem the objective is minimizing the value of function **f**(**x**)
where **f**(**x**) = avg(**a** + **b** + **c** + **d**). Since there are four variables in the equation, namely

a, **b**, **c**, and **d**, we can compose the chromosome as follow: To speed up the computation, we can restrict that the values of variables **a**, **b**, **c**, and **d** are floats between 0 and 100 **(we used floats numbers to make the result of the best chromosome converge to the ideal solution)**

a	b	c	d

Step 1. Initialization
For example we define the number of chromosomes in population are 10, then we generate random value of gene **a**, **b**, **c**, **d** for 6 chromosomes

Chromosome[1] = [**a**;**b**;**c**;**d**] = [11.1935965 83.30917189 33.94997522 57.29563453]
Chromosome[2] = [**a**;**b**;**c**;**d**] = [24.33678075 21.40156641 28.95327074 20.27212823]
Chromosome[3] = [**a**;**b**;**c**;**d**] = [66.25390084 48.34089347 97.34323317 30.22498128]
Chromosome[4] = [**a**;**b**;**c**;**d**] = [51.93800964 79.10142873 11.79686082 82.70688243]
Chromosome[5] = [**a**;**b**;**c**;**d**] = [18.53951476 6.88945018 66.19665062 45.05446386]
Chromosome[6] = [**a**;**b**;**c**;**d**] = [86.58561064 67.99999023 45.15672055 0.73250022]

Chromosome[7] = [**a**;**b**;**c**;**d**] = [81.2526174 89.9003671 89.95338256 38.79150891]
Chromosome[8] = [**a**;**b**;**c**;**d**] = [28.67859849 88.94751862 16.51816212 55.76948657]
Chromosome[9] = [**a**;**b**;**c**;**d**] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[10] = [**a**;**b**;**c**;**d**] = [52.57950247 72.87726874 99.44574003 51.41822224]

Step 2. Evaluation
We compute the objective function value for each chromosome produced in initialization step:
F_obj[1] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(46.43709453)^2 + 4* (46.43709453) - 4)
= Abs(4494.555875)
= 4494.555875
F_obj[2] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(23.74093653)^2 + 4* (23.74093653) - 4)
= Abs(1218.227881)
= 1218.227881
F_obj[3] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(60.54075219)^2 + 4* (60.54075219) - 4)
= Abs(7568.52836)
= 7568.52836
F_obj[4] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(56.3857954)^2 + 4* (56.3857954) - 4)
= Abs(6580.259027)
= 6580.259027

F_obj[5] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)

= Abs(2*(34.17001986)^2 + 4* (34.17001986) - 4)

= Abs(2467.860594)

= 2467.860594

F_obj[6] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)

= Abs(2*(50.11870541)^2 + 4* (50.11870541) - 4)

= Abs(5220.244086)

= 5220.244086

F_obj[7] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)

= Abs(2*(74.97446899)^2 + 4* (74.97446899) - 4)

= Abs(11538.23988)

= 11538.23988

F_obj[8] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)

= Abs(2*(47.47844145)^2 + 4* (47.47844145) - 4)

= Abs(4694.318571)

= 4694.318571

F_obj[9] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)

= Abs(2*(32.73943478)^2 + 4* (32.73943478) - 4)

= Abs(2270.698919)

= 2270.698919

F_obj[10] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)

= Abs(2*(69.08018337)^2 + 4* (69.08018337) - 4)

= Abs(9816.464202)

= 9816.464202

Step 3. Selection

1. The fittest chromosomes have higher probability to be selected for the next generation. To compute fitness probability we must compute the fitness of each chromosome. To avoid divide by zero problem, the value of **F_obj** is added by 1.

Fitness[1] = 1 / (1+**F_obj**[1])

= 1 / 4495.555875

= 2.22441902×10−4

Fitness[2] = 1 / (1+**F_obj**[2])

= 1 / 1219.227881

= 8.20191217×10−4

Fitness[3] = 1 / (1+**F_obj**[3])

= 1 / 7569.52836

= 1.32108627×10−4

Fitness[4] = 1 / (1+**F_obj**[4])

= 1 / 6581.259027

= 1.5194661×10−4

Fitness[5] = 1 / (1+**F_obj**[5])

= 1 / 2468.860594
= 4.05045146×10−4
Fitness[6] = 1 / (1+**F_obj**[6])
= 1 / 5221.244086
= 1.91525235×10−4
Fitness[7] = 1 / (1+**F_obj**[7])
= 1 / 11539.23988
= 8.66608209×10−5
Fitness[8] = 1 / (1+**F_obj**[8])
= 1 / 4695.318571
= 2.12978094×10−4
Fitness[9] = 1 / (1+**F_obj**[9])
= 1 / 2271.698919
= 4.40199179×10−4
Fitness[10] = 1 / (1+**F_obj**[10])
= 1 / 9817.464202
= 1.01859297×10−4

Total = 2.22441902×10−4 + 8.20191217×10−4 + 1.32108627×10−4 + 1.5194661×10−4 + 4.05045146×10−4 + 1.91525235×10−4 + 8.66608209×10−5 + 2.12978094×10−4 + 4.40199179×10−4 + 1.01859297×10−4
= 29.44149415 ×10−4

The probability for each chromosomes is formulated by: **P**[i] = **Fitness**[i] / **Total**
P[1] = 2.22441902/ 29.44149415
= 0.07555387674
P[2] = 8.20191217/ 29.44149415
= 0.2785834214
P[3] = 1.32108627/ 29.44149415
= 0.04487157694
P[4] = 1.5194661/ 29.44149415
= 0.05160968028
P[5] = 4.05045146/ 29.44149415
= 0.1375762874
P[6] = 1.91525235/ 29.44149415
= 0.06505282443
P[7] = 0.866608209/ 29.44149415
= 0.02943492625
P[8] = 2.12978094/ 29.44149415
= 0.07233943118
P[9] = 4.40199179/ 29.44149415

= 0.1495165893
P[10] = 1.01859297/ 29.44149415
= 0.0345971901

From the probabilities above we can see that Chromosome 4 that has the highest fitness, this chromosome has highest probability to be selected for next generation chromosomes. For the selection process we use roulette wheel, for that we should compute the cumulative probability values:

C[1] = 0.07555387674×10−4
C[2] = 0.07555387674+ 0.2785834214
= 0.3541372981
C[3] = 0.07555387674+ 0.2785834214+ 0.04487157694
= 0.3990088751
C[4] = 0.07555387674+ 0.2785834214+ 0.04487157694+ 0.1375762874
= 0.5365851625
C[5] = 0.07555387674+ 0.2785834214+ 0.04487157694+ 0.1375762874+ 0.1375762874
= 0.6741614499
C[6] = 0.07555387674+ 0.2785834214+ 0.04487157694+ 0.1375762874+ 0.1375762874+ 0.06505282443
= 0.7392142743

C[7] = 0.0755538767+ 0.2785834214+ 0.04487157694+ 0.1375762874+ 0.1375762874+ 0.06505282443+ 0.02943492625
= 0.7686492005

C[8] = 0.07555387674+ 0.2785834214+ 0.04487157694+ 0.1375762874+ 0.1375762874+ 0.06505282443+ 0.02943492625+ 0.07233943118
= 0.8409886317

C[9] = 0.07555387674+ 0.2785834214+ 0.04487157694+ 0.1375762874+ 0.1375762874+ 0.06505282443+ 0.02943492625+ 0.07233943118+ 0.1495165893
= 0.970505221

C[10] = 0.07555387674+ 0.2785834214+ 0.04487157694+ 0.1375762874+ 0.1375762874+ 0.06505282443+ 0.02943492625+ 0.07233943118+ 0.1495165893+ 0.0345971901

= 1.0

Having calculated the cumulative probability of selection process using roulette-wheel can be done. The process is to generate random number **R** in the range 0-1 as follows.
R[1] = 0.56177803

R[2] = 0.84863847
R[3] = 0.9182119
R[4] = 0.91172082
R[5] = 0.83018734
R[6] = 0.75299129

R[7] = 0.60776276
R[8] = 0.3538906
R[9] = 0.79336161
R[10] = 0.71450196

If random number **R**[1] is greater than **C**[1] and smaller than **C**[2] then select **Chromosome**[2] as a chromosome in the new population for next generation:

NewChromosome[1] = **Chromosome**[5]
NewChromosome[2] = **Chromosome**[9]
NewChromosome[3] = **Chromosome**[9]
NewChromosome[4] = **Chromosome**[9]
NewChromosome[5] = **Chromosome**[8]
NewChromosome[6] = **Chromosome**[7]

NewChromosome[7] = **Chromosome**[5]
NewChromosome[8] = **Chromosome**[2]
NewChromosome[9] = **Chromosome**[8]
NewChromosome[10] = **Chromosome**[6]

Chromosomes in the population thus became:

Chromosome[1] = [18.53951476 6.88945018 66.19665062 45.05446386]
Chromosome[2] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[3] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[4] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[5] = [28.67859849 88.94751862 16.51816212 55.76948657]
Chromosome[6] = [81.2526174 89.9003671 89.95338256 38.79150891]
Chromosome[7] = [18.53951476 6.88945018 66.19665062 45.05446386]
Chromosome[8] = [24.33678075 21.40156641 28.95327074 20.27212823]
Chromosome[9] = [28.67859849 88.94751862 16.51816212 55.76948657]
Chromosome[10] = [86.58561064 67.99999023 45.15672055 0.73250022]

In this example, we use one-cut point, i.e. randomly select a position in the parent chromosome then exchanging sub-chromosome. Parent chromosome which will mate is randomly selected and the number of mate Chromosomes is controlled using **crossover_rate** (**ρc**) parameters. Pseudo-code for the crossover process is as follows:

https://github.com/Marouane20/GeneticAlgo

```
begin
    k← 0;
    while(k<population) do
        R[k] = random(0-1);
        if(R[k]< ρc) then
            select Chromosome[k] as parent;
        end;
        k = k + 1;
    end;
end;
```

Chromosome **k** will be selected as a parent if **R**[**k**]<**ρc**. Suppose we set that the crossover rate is 33%, then Chromosome number **k** will be selected for crossover if random generated value for Chromosome **k** below 0.33. The process is as follows: First we generate a random number **R** as the number of population.

R[1] = 0.91245173
R[2] = 0.14305563
R[3] = 0.52689373
R[4] = 0.32925696
R[5] = 0.9448646
R[6] = 0.36109722

R[7] = 0.96023052
R[8] = 0.79299481
R[9] = 0.13507707
R[10] = 0.95035777

For random number **R** above, parents are **Chromosome**[2], **Chromosome**[4] and **Chromosome**[9] will be selected for crossover.

Chromosome[2] >< **Chromosome**[4]
Chromosome[4] >< **Chromosome**[9]
Chromosome[9] >< **Chromosome**[2]

After chromosome selection, the next process is determining the position of the crossover point. This is done by generating random numbers between 1 to (length of Chromosome – 1).
In this case, generated random numbers should be between 1 and 3. After we get the crossover point, parents Chromosome will be cut at crossover point and its gens will be interchanged. For example we generated 3 random number and we get:

C[1] = 1
C[2] = 1
C[3] = 2

Then for first crossover, second crossover and third crossover, parent's gens will be cut at gen number 1, gen number 1 and gen number 3 respectively, e.g.

Chromosome[2] = **Chromosome**[2] >< **Chromosome**[4]
= [48.18506185 33.3495559 23.53558623 25.88753512]>< [48.18506185 33.3495559 23.53558623 25.88753512]
= [48.18506185 33.3495559 23.53558623 25.88753512]

Chromosome[4] = **Chromosome**[4] >< **Chromosome**[9]
= [48.18506185 33.3495559 23.53558623 25.88753512]>< [28.67859849 88.94751862 16.51816212 55.76948657]

= [48.18506185 88.94751862 16.51816212 55.76948657]

Chromosome[9] = **Chromosome**[9] >< **Chromosome**[2]
= [28.67859849 88.94751862 16.51816212 55.76948657]>< [48.18506185 33.3495559 23.53558623 25.88753512]
= [28.67859849 88.94751862 23.53558623 25.88753512]

Thus Chromosome population after experiencing a crossover process:

Chromosome[1] = [18.53951476 6.88945018 66.19665062 45.05446386]
Chromosome[2] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[3] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[4] = [48.18506185 88.94751862 16.51816212 55.76948657]
Chromosome[5] = [28.67859849 88.94751862 16.51816212 55.76948657]
Chromosome[6] = [81.2526174 89.9003671 89.95338256 38.79150891]
Chromosome[7] = [18.53951476 6.88945018 66.19665062 45.05446386]
Chromosome[8] = [24.33678075 21.40156641 28.95327074 20.27212823]
Chromosome[9] = [28.67859849 88.94751862 23.53558623 25.88753512]
Chromosome[10] = [86.58561064 67.99999023 45.15672055 0.73250022]

Step 5. Mutation
Number of chromosomes that have mutations in a population is determined by the **mutation_rate** parameter. Mutation process is done by replacing the gen at random

position with a new value. The process is as follows. First we must calculate the total length of gen in the population. In this case the total length of gen is

total_gen = **number_of_gen_in_Chromosome** * **number of population**
= 4 * 10
= 40
Mutation process is done by generating a random integer between 1 and total_gen (1 to 24).
If generated random number is smaller than mutation_rate(ρm) variable then marked the position of gen in chromosomes. Suppose we define ρm 10%, it is expected that 10% (0.1) of total_gen in the population that will be mutated:
number of mutations = 0.1 * 40
= 4
≈ 4
Suppose generation of random number yield 4,12 ,18 and 25 then the chromosome which have mutation are Chromosome number 1 gen number 4 and Chromosome 3 gen number 4 and Chromosome 5 gen number 2 and Chromosome 7 gen number 1. The value of mutated gens at mutation point is replaced by random number between 0-30.
Suppose generated random number are 4.00689182 , 8.23107495 , 5.51887826 and 21.40156641 then Chromosome composition after mutation
are:

Chromosome[1] = [18.53951476 6.88945018 66.19665062 4.00689182]
Chromosome[2] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[3] = [48.18506185 33.3495559 23.53558623 8.23107495]
Chromosome[4] = [48.18506185 88.94751862 16.51816212 55.76948657]
Chromosome[5] = [28.67859849 5.51887826 16.51816212 55.76948657]
Chromosome[6] = [81.2526174 89.9003671 89.95338256 38.79150891]
Chromosome[7] = [21.40156641 6.88945018 66.19665062 45.05446386]
Chromosome[8] = [24.33678075 21.40156641 28.95327074 20.27212823]
Chromosome[9] = [28.67859849 88.94751862 23.53558623 25.88753512]
Chromosome[10] = [86.58561064 67.99999023 45.15672055 0.73250022]

Finishing mutation process then we have one iteration or one generation of the genetic algorithm. We can now evaluate the objective function after one generation:

F_obj[1] = $Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)$
= $Abs(2*(23.90812685)^2 + 4* (23.90812685) - 4)$
= Abs(1234.829566)
= 1234.829566
F_obj[2] = $Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)$
= $Abs(2*(32.73943477)^2 + 4* (32.73943477) - 4)$
= Abs(2270.698917)
= 2270.698917

F_obj[3] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(28.32531973)^2 + 4* (28.32531973) - 4)
= Abs(1713.948755)
= 1713.948755
F_obj[4] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(52.35505729)^2 + 4* (52.35505729) - 4)
= Abs(5687.524277)
= 5687.524277
F_obj[5] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(26.62128136)^2 + 4* (26.62128136) - 4)
= Abs(1519.870368)
= 1519.870368
F_obj[6] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(74.97446899)^2 + 4* (74.97446899) - 4)
= Abs(11538.23988)
= 11538.23988
F_obj[7] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(34.88553277)^2 + 4* (34.88553277) - 4)
= Abs(2569.542924)
= 2569.542924
F_obj[8] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(23.74093653)^2 + 4* (23.74093653) - 4)
= Abs(1218.227881)
= 1218.227881
F_obj[9] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(41.76230961)^2 + 4* (41.76230961) - 4)
= Abs(3651.230246)
= 3651.230246
F_obj[10] = Abs(2*avg(a + b + c + d)^2 + 4*avg(a + b + c + d) - 4)
= Abs(2*(50.11870541)^2 + 4* (50.11870541) - 4)
= Abs(5220.244086)
= 5220.244086

From the evaluation of new Chromosome we can see that the objective function is decreasing, this means that we have better Chromosome or solution compared with previous Chromosome generation. New Chromosomes for next iteration are:

Chromosome[1] = [18.53951476 6.88945018 66.19665062 4.00689182]
Chromosome[2] = [48.18506185 33.3495559 23.53558623 25.88753512]
Chromosome[3] = [48.18506185 33.3495559 23.53558623 8.23107495]
Chromosome[4] = [48.18506185 88.94751862 16.51816212 55.76948657]
Chromosome[5] = [28.67859849 5.51887826 16.51816212 55.76948657]
Chromosome[6] = [81.2526174 89.9003671 89.95338256 38.79150891]
Chromosome[7] = [21.40156641 6.88945018 66.19665062 45.05446386]

https://github.com/Marouane20/GeneticAlgo

Chromosome[8] = [24.33678075 21.40156641 28.95327074 20.27212823]
Chromosome[9] = [28.67859849 88.94751862 23.53558623 25.88753512]
Chromosome[10] = [86.58561064 67.99999023 45.15672055 0.73250022]

These new Chromosomes will undergo the same process as the previous generation of Chromosomes such as evaluation, selection, crossover and mutation and at the end it produce new generation of Chromosome for the next iteration. This process will be repeated until a predetermined number of generations. For this example, after running 150 generations,

Best chromosome is obtained:

Chromosome = [1.08186958 0.13502474 0.96939692 0.81144178]
This means that:

a = 1.08186958

b = 0.13502474

c = 0.96939692

d = 0.81144178

If we use the number in the problem equation:

$\mathbf{2X^2} + 4\mathbf{X} - 4 = 0$

Than X will be the average of gens (1.08186958+0.13502474+0.96939692+0.81144178)/4

X= 0.7494

$2*\text{avg}(a + b + c + d)^2 + 4*\text{avg}(a + b + c + d) – 4$

$=2*(0.749433255)^2 + 4*0.749433255 – 4$

=0.1210

We can see that the value of variable **a**, **b**, **c** and **d** generated by genetic algorithm can almost satisfy that equality.

To solve linear equation wee can just take **$\alpha = 0$ in $\alpha X^2+\beta X+\lambda$** and repeat the same process.

REFERENCE

Genetic Algorithm for Solving Simple Mathematical Equality Problem [Denny Hermawanto Indonesian Institute of Sciences (LIPI), INDONESIA] arxiv.org

https://github.com/Marouane20/GeneticAlgo