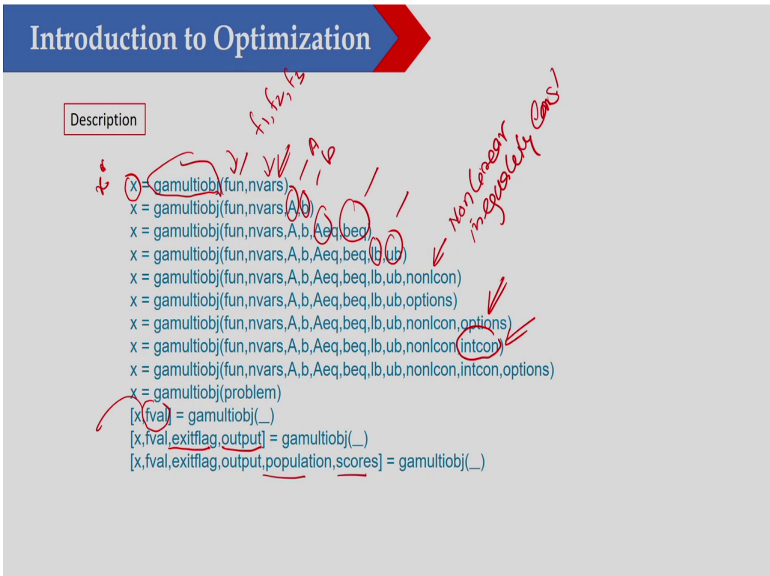
**Optimization Methods for Civil Engineering**
**Prof. Rajib Kumar Bhattacharjya**
**Department of Civil Engineering**
**Indian Institute of Technology, Guwahati**

**Lecture - 37**
**Multi-objective GA using Matlab**

Hello student. Welcome back to the course on Optimization Methods for Civil Engineering. In this particular class I will solve few problems related to Multi-objective optimization problem using MATLAB. Already we have solved multi-objective optimization problem using R software. I hope you could solve some of the problems using R.

So, today in today's class, so, I will solve few problems using genetic algorithm multi objective genetic algorithm using MATLAB platform.

(Refer Slide Time: 01:11)



So, let us see the function that I will use for solving a multi objective optimization problem with constrain or without constrain that function is that function is ga multi objective ok. So, this is the function gamultiobj. So, this is the function and these are the arguments. Suppose I have to put function here and number of variables. So, this is this arguments are similar to the ga arguments or similar to the fmincon function I have used.

So, here in this particular function. So, I can handle a multi-objective optimization problem with linear constrain, non-linear constrain, linear equality constrain, linear non-equality constrain and non-linear equality and non-linear inequality constrain along with integer variable. So, if you have any integer variable that also you can define in this particular function.

So, if you have any problem with integer variables, so, then also you can use this particular function. So, arguments are similar to that you have used in ga or fmincon. So, in function so, here you have to define the objective function here suppose if you have two objective function then you have to define f 1 and f 2, if you have 3 then I will define f 3.

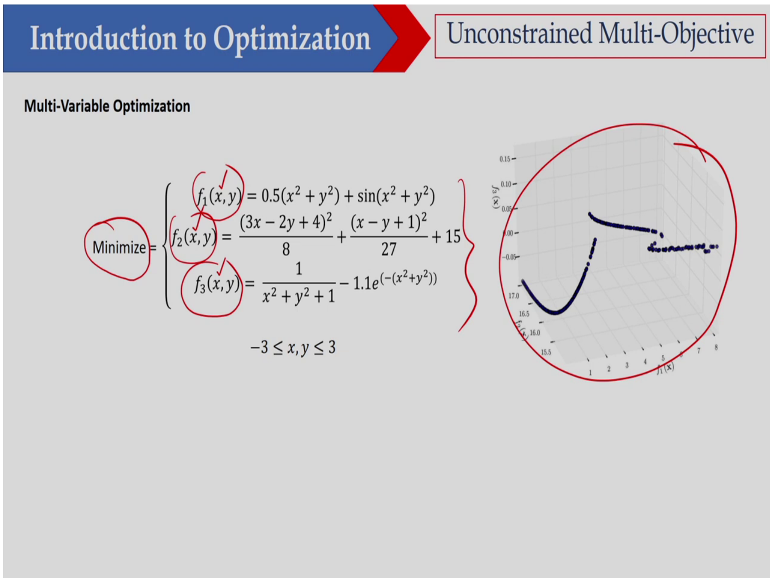
Then I have to define the number of variables then if you have inequality linear constrain then you define A and b and if you have equality linear constrain then also you can define then you can also define lower bound and upper bound; already we have discussed that part. So, you know how to define lower bound and upper bounds. And if you have non-linear inequality constrains; so, this is non-linear inequality constrain.

So, that also you can define. And if you have option; so, you can change the population size you can change the generation you can change the crossover probability, mutation probability and you can also define what type of function you are using for crossover for mutation then for selection. So, that also you can define here in the option under option.

Then if you have any integer your then also you can define here. So, what I am getting from this particular function? So, I will get the optimal value of x; that means, the x star I will get. I can also get the function value then exitflag that I will get; I will get the other output then population, scores. So, that also I can get from this particular function ok. So, that as an output I can get when you are using this multi objective ga function.

So, I hope this is not difficult because already you have learned how to solve a multivariable problem or a single variable problem with a single objective function; that means, single objective function problem you can solve it. So, using ga function and arguments are same. So, the function type is same. Only the name of the function is different. So, there I have used ga function and here it is multi objective ga or this is basically ga multi objective ok. gamultiobj the function name is different ok. So, arguments are same.

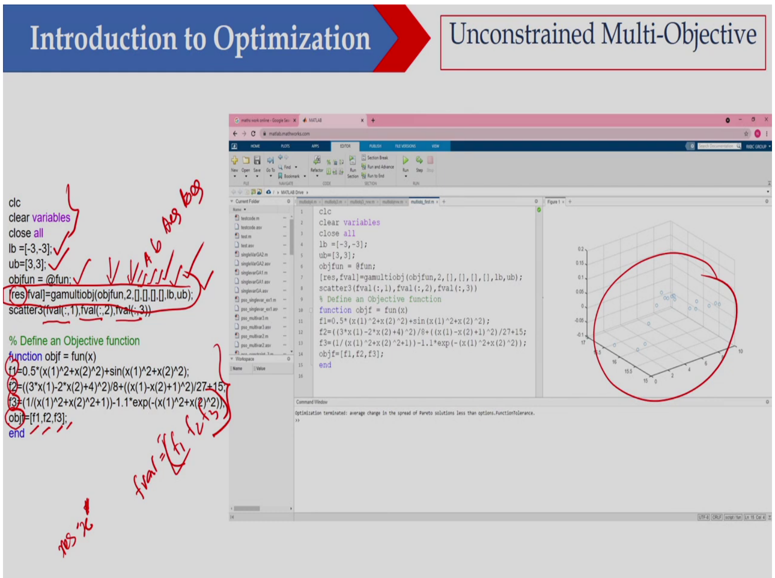
(Refer Slide Time: 05:11)




Now let us see; so, I will solve three problems here. So, let us see the first problem. The first problem we have three objective function. So, we have f 1, we have f 2 and we have f 3 and the functions are minimization types. So, functions are minimization type and we have three objective function. So, how many variable we have? We have two variables; that is x and y ok.

So, f 1 is a function of x and y, f 2 is also a function of x and y and f 3 also a function of x and y. So, this is the function and if I solve this particular problem, so, I should get this Pareto optimal front. So, I should get this Pareto optimal front and let us see whether I will get the same Pareto optimal front here. So, I should get the same Pareto optimal front.

(Refer Slide Time: 06:14)



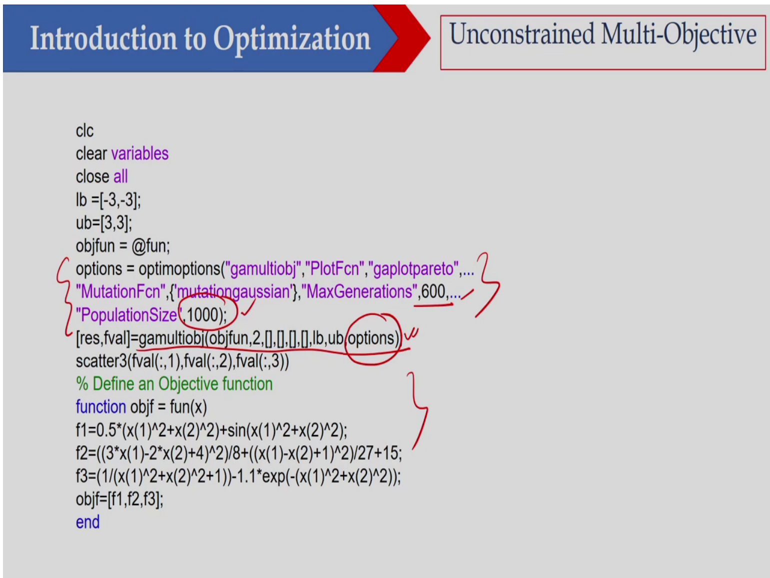

Now, this is the MATLAB code. So, here what you have to do? You have to write the objective function. So, here we have three objective function that is f 1, f 2 and f 3 and after that so, this is the vector. So, objective is a vector so, that is with the where we have f 1, f 2 and f 3. So, this we have to written from this particular function. So, these three variables are to clear the environment then you define lower bound and upper bound and here objective function is your fun.

So, this is the name of the function. So, this is the name of the function and now this is the multi objective ga function. So, ga multi objective and here I have defined objective a number of variables 2 and then we do not have A b A equality then b equality. So, we do not have. So, I am putting null null null. So, empty vector and then I have defined lower bound and upper bound.

So, if you are solving; so, you should get the solution of this problem and then I am plotting this particular objective function that is so that is whatever your solution will be at rest. So, x star will be at basically store at result. So, res here then fval. So, this is basically will contain the f 1 ok, f 2 and f 3 ok. So, here so, I am plotting that the objective function 1, objective function 2 and objective function 3 as a scatter plot then I should get this particular plot.

So, here I did not define any parameter. So, I have used the default parameters. Number of population default population I have used all the parameters I have used default parameters. So, therefore, the population size is less here and I am getting few points on the Pareto optimal front. Now, if I would like to increase the population size, so, what I can do? I can go to option and I can define the population size ok.

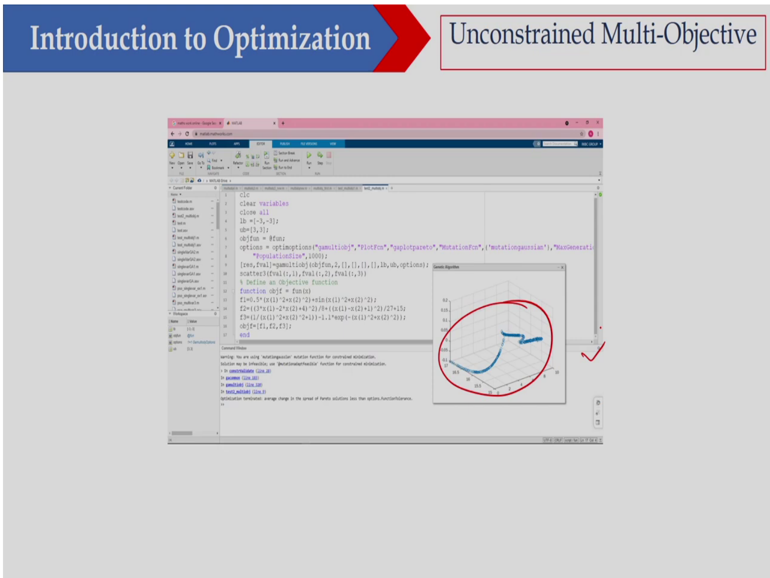
(Refer Slide Time: 08:40)




So, here what I can do? I can put this option. So, here I am putting generation 600 and population size 1000. So, I am putting 1000 population size so that I am getting a continuous Pareto optimal front ok. So, I can increase the population size using this option. So, I have increased here up to 1000 and I am using 600 generation, but if you want you can also change that one. So, what changes you have to do on this particular line that is the multi objective genetic algorithm function.

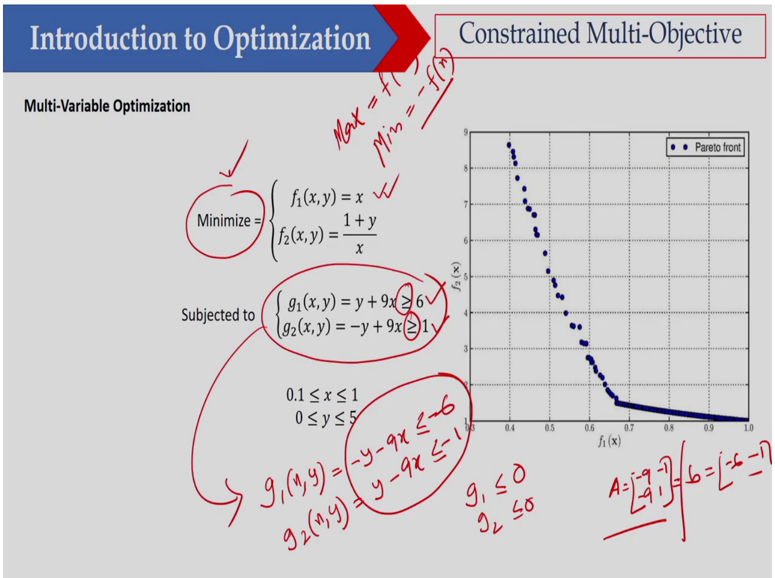
So, I have to put this option here ok. So, option and then these options will be taken. Now, if I execute this particular m file. So, I will get the continuous Pareto optimal front ok. So, we will see this one. So, when I will use the MATLAB online MATLAB. So, there actually I will try to run this particular function and to get the Pareto optimal front.

(Refer Slide Time: 09:50)




So, this is basically. So, this you should get ok so, if you are executing that particular m file.

(Refer Slide Time: 09:56)




So, the next problem is a multi objective optimization problem with constrain. So, we have two constrains here; this is the first constrain and this is the second constrain. Here the objective functions are minimization types so, this is fine. So, if your problem is a maximization type problem then you have to convert it to minimization type problem; that means what you have to do?

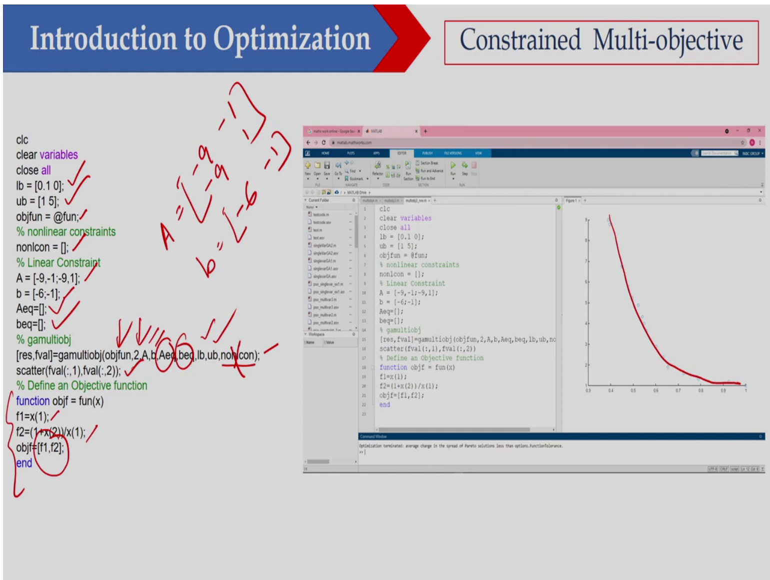
You have to minimum you have to convert it to minimization type problem which is equal to minus of your f x ok. So, suppose I have maximization of f x; so, that I can do minimization of minus of f x ok. So, I have to do that, but in this case this is a objective functions are minimization type.

Now look at the constrain so, constrain function should be that g 1 should be less than equal to 0 and g 2 should be less than equal to 0, but here this is greater than equality type. So, what I will do basically? So, I will convert this one that is g 1 x, y ok.

So, this is equal to minus y minus 9 x. So, this is less than equal to minus 6, similarly that g 2 x, y. So, what I will do? I will convert I will multiply it by minus 1 then I will get y minus 9 x less than equal to minus 1. So, now, this is your less than equality type constrain. So, I can define what is the value of A and what is the value of b. So, in this case A will be; A will be this is minus 9 minus 1 and this is minus 9 plus 1 and b will be your minus 6 minus 1 ok.

So, this is your A and b. So, I can put this A and b in the ga multi objective function. So, that I will use so, this is the problem with constrains. So, we have two constrain and this constrains are linear inequality constrains ok, so, linear inequality constrain. So, I have to put what is the value of A and I have to put what is the value of b.

(Refer Slide Time: 12:34)



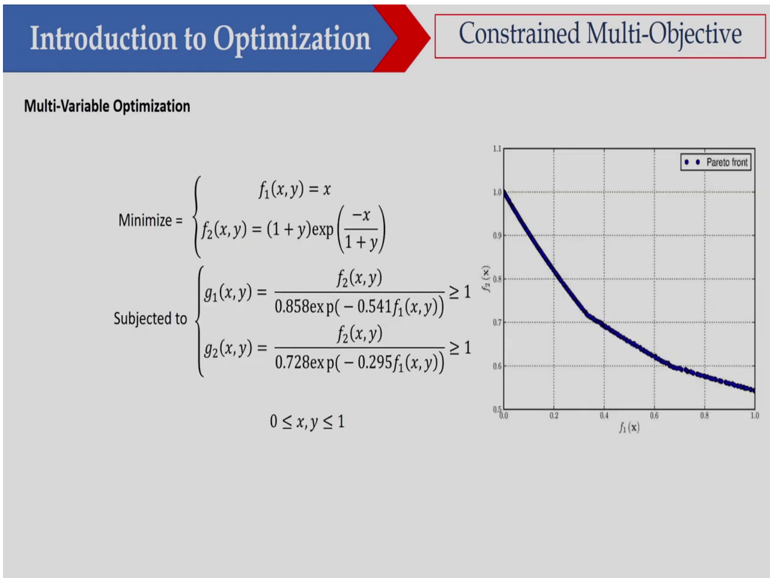

So, let us see the m file; here I am defining what is lower bound and upper bound. So, lower bound is 0.1 and 0 upper bound is 1 and 5 then objective function. So, this is the objective function. So, objective function f 1 equal to x 1 f 2 equal to 1 plus x 2 by x 1 and we have two objective function.

And so, we do not have any non-linear constrain. So, we have A and b so, that is equality type linear constrain. So, here this is your A. So, A is minus 9 minus 1 then minus 9 plus 1 ok and b is minus 6 and basically minus 1. So, this is your b. So, I have defined A and b and then this is the ga multi objective.

So, this is objective function then number of variables are 2 then I have defined what is A what is b. A equality b equality is not there. So, I am putting null and then lower bound upper bound we have defined and we do not have any non-linear constrain ok. So, non-linear constrain we do not have. So, you can put null here or I can also delete this portion that is not required.

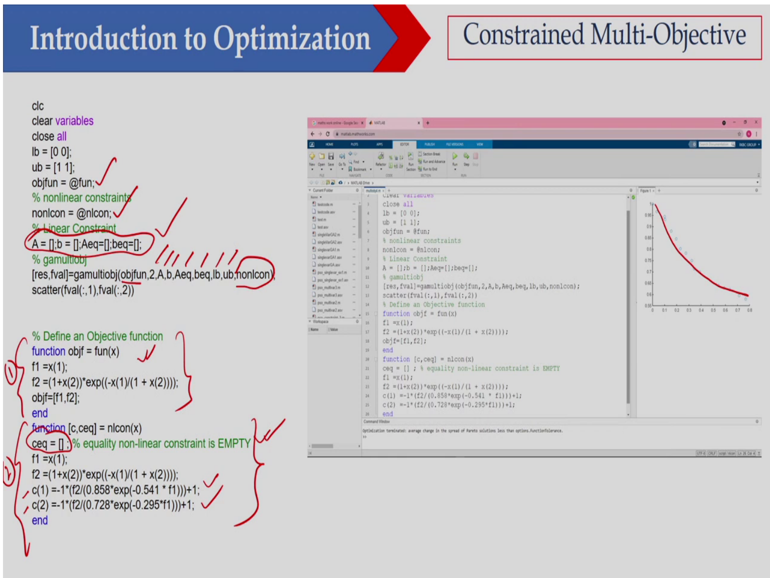
And then I am plotting the objective function 1 and objective function 2. So, I will get the Pareto optimal front. So, this is the code then you should get this Pareto optimal front ok. So, this you should get. So, this is the second problem.

(Refer Slide Time: 14:22)




The third problem is a multi-objective problem with non-linear inequality constrains. So, we have two non-linear inequality constrains. So, what we have to do? We have to write the inequality constrain function also non-linear inequality constrain function also. So, we have to write a function where we have defined this non-linear inequality constrain. So, we do not have any equality constrain. So, we will just pass the null vector.

(Refer Slide Time: 14:59)

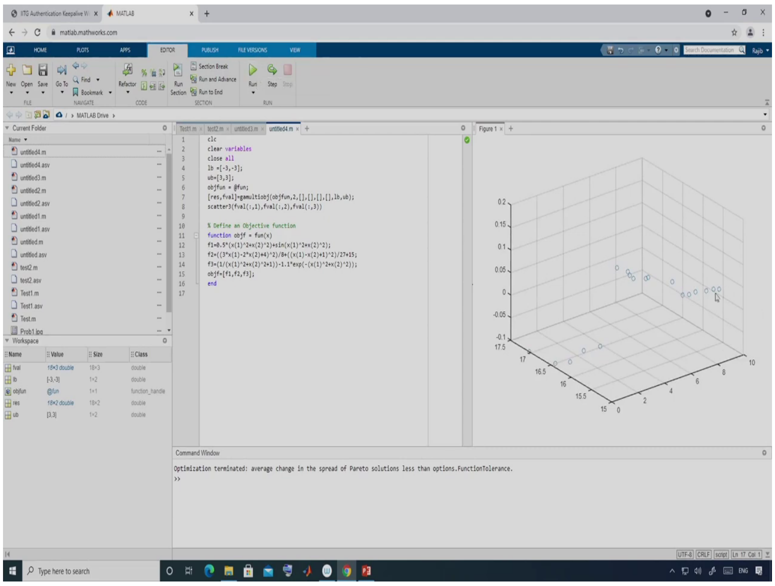



So, here we have to define two function, this is the first function, this is the first function this is function 1 and this is function 2. Function 1 is the objective function like that as we have non-linear inequality constrain. So, therefore, I am also defining this is c and c 1 and c 2. So, we have 2 constrains c 1 and c 2 and we do not have any equality type constrain. So, I am just returning an empty vector.

And so, this is the function I have defined and this is the objective function I have defined and then now I am using objective function then constrain then we do not have any linear constrain. So, this is we are putting empty and then I am defining this objective function number of variable is 2 then A, b, Aeq, beq is not there then lower bound upper bound I have defined and then I have also defined non-linear constrain ok.

So, we have non-linear inequality type constrain. Now, if I run this particular function so, I should get this Pareto optimal front ok. So, I can increase the population size by using option. So, then I will get a continuous line or otherwise you will get this particular your Pareto optimal front ok. Then let us go to MATLAB and just see whether I am getting the similar solution or not. So, let us run this particular m file in MATLAB. Let us solve the first problem. So, this is the m file.

(Refer Slide Time: 16:56)

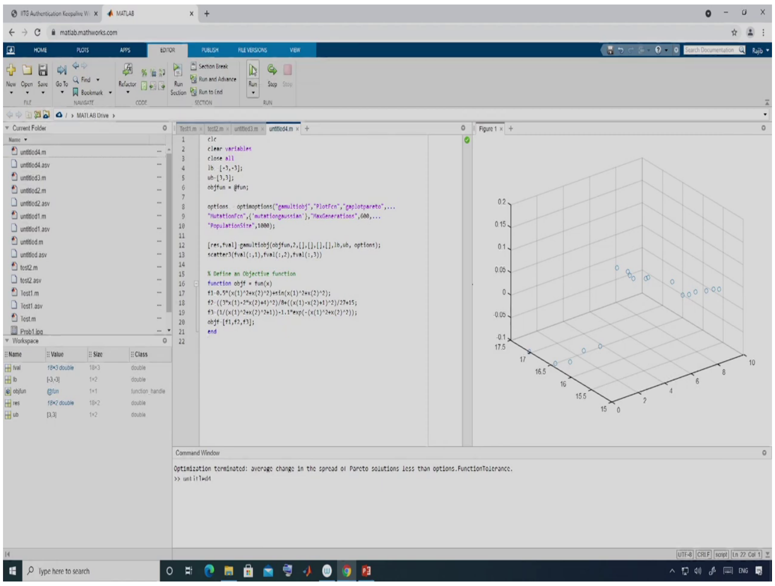



So, I would just copy this one and I will go to; I will go to MATLAB. And here you just see so, these are just to clean the environment and then I have defined lower bound that is minus 3 and minus 3, upper bound this is 3 and 3, the objective function is your fun ok. So, this is the objective function name ok. And then I have used this particular function to solve this problem.

So, before that I have to write this objective function. So, here this is the first objective function f 1, this is your f 2 and this is your f 3 and then you have to put in this particular objective function vector. So, here we have f 1, f 2 and f 3 ok. Now, I have defined this objective function. Now, if I execute this particular file, so, I should get the solution of this problem. So, let me run this one ok.

So, I am getting that one, but what is happening? The population size is less. So, therefore, I am not getting; I am not getting a continuous Pareto optimal front. So, what I can do? I can increase the population size ok. So, how I will do that? I will put the option. So, here is the option. So, I can put it. So, this is the option.

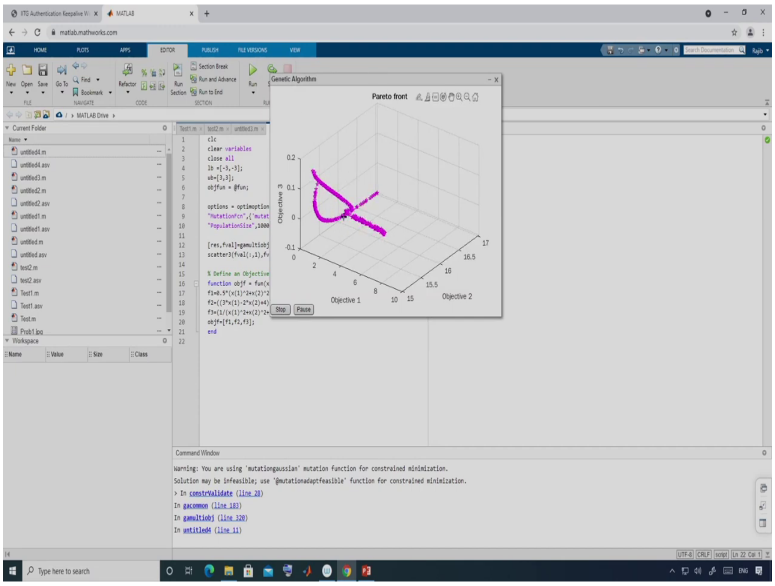
(Refer Slide Time: 19:02)



So, I am defining here the option. So, this is the option and here I have to define options ok. So, now, what I am doing here? So, I am using maximum generation equal to 600 and population size is 1000. I can also use some different population size let me see that 1 then I am using plot function. So, this is gaplotpareto. So, it will plot the Pareto optimal front in every iteration.

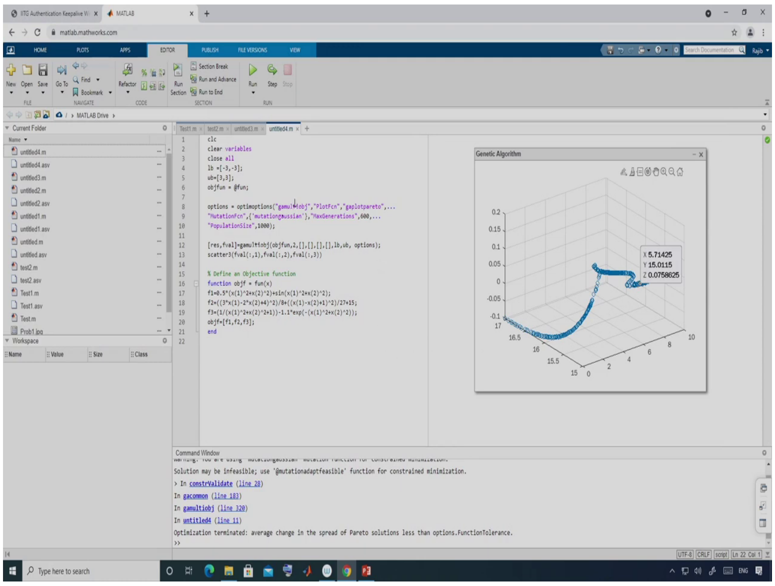
And mutation function I have used Gaussian mutation. So, this is your real coded ga and I have used Gaussian mutation function. So, you can also define what type of crossover function you want to use. So, you can define in this option line. So, you can define and also you can define the probability of crossover probability of mutation then what type of function you would like to use for selection. So, that also you can define. Otherwise the default function will be used.

(Refer Slide Time: 20:22)



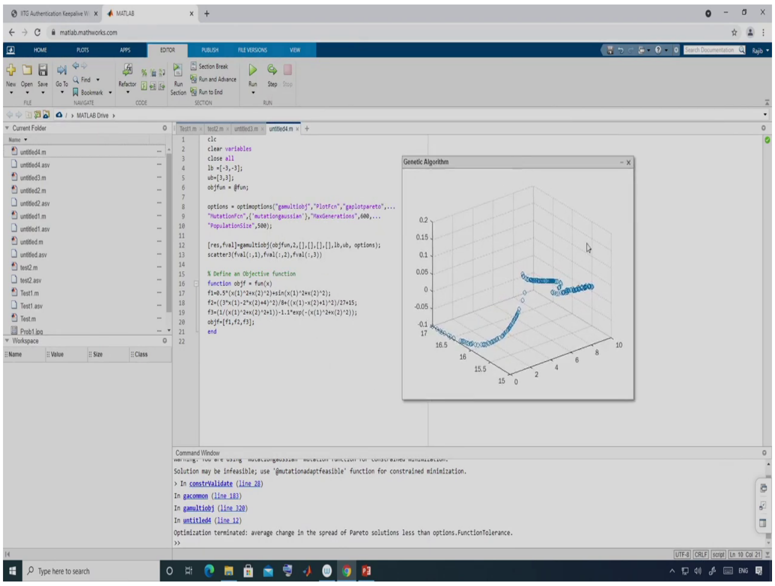

So, let me execute this particular m file. Now, you just see that I am getting this Pareto optimal front ok and this is the final Pareto optimal front ok.

(Refer Slide Time: 20:29)



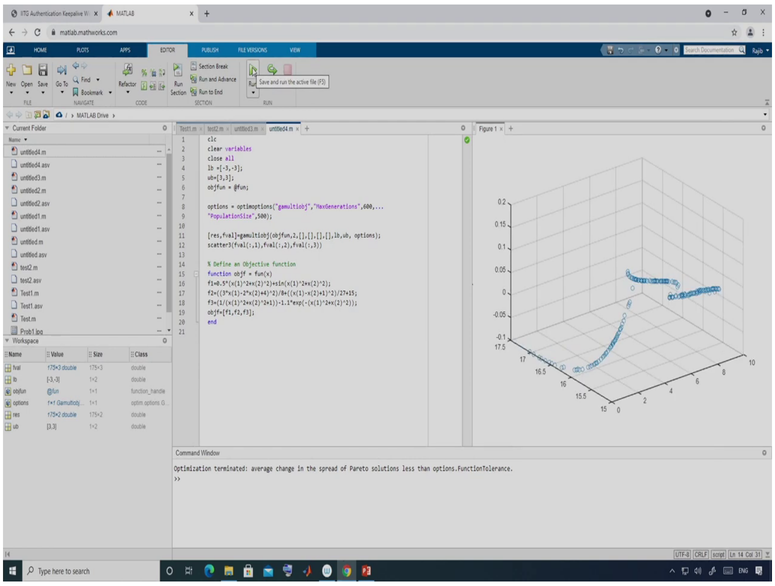
So, this is similar to what actually I have shown you earlier. So, now, our population size is more. So, therefore, you are getting a continuous Pareto optimal you are getting a continuous Pareto optimal front. So, let me check if I want to suppose the population size 500. So, 500 then what will happen? Yeah.

(Refer Slide Time: 21:05)



So, I am getting so, now, you just see you are getting, but there are some gap, but if you want to increase or if you want to see the continuous line. So, in that case you have to increase the number of population. Or suppose I would like to change only the population size and number of iteration then you can delete this part.

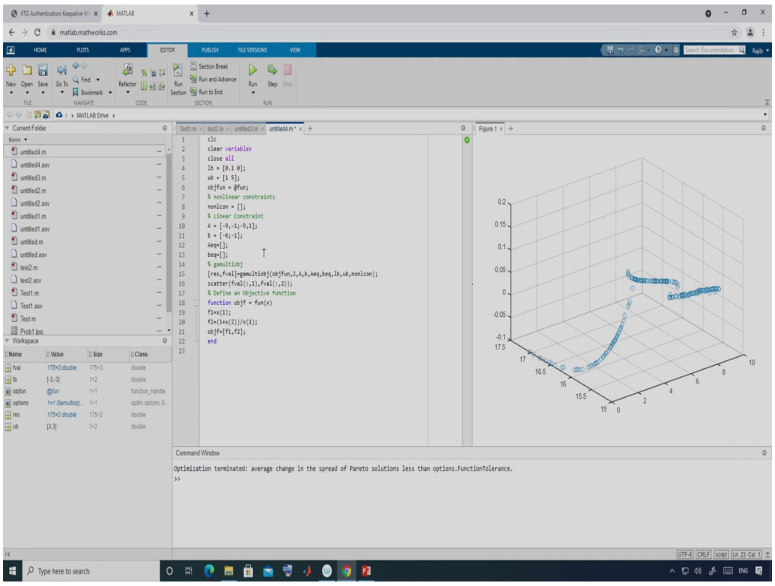
(Refer Slide Time: 21:38)



So, suppose I do not like to see the plot in every generation. So, what I am doing? I am just defining maximum generation that is 600 and population size 500. So, let me then this plot will not be shown. So, only you will get the scatter plot at the end yeah. So, you are getting this scatter plot.

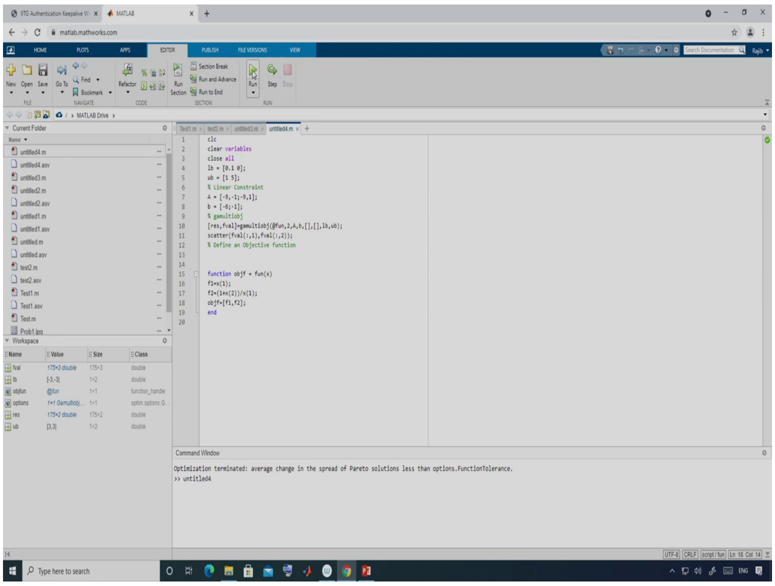
So, it is not showing the plot after every iteration ok. So, it is not showing that one. So, you are getting the final Pareto optimal front. So, using option so, I can change the parameters of the genetic algorithm. Now, let me go to the next problem. So, let us see the second problem. So, this is the second problem so, it has 2 linear constrains. So, I will show you how you can define this linear constrain. So, here this is the code ok.

(Refer Slide Time: 22:51)




So, this is the code, this is the objective function.

(Refer Slide Time: 22:56)

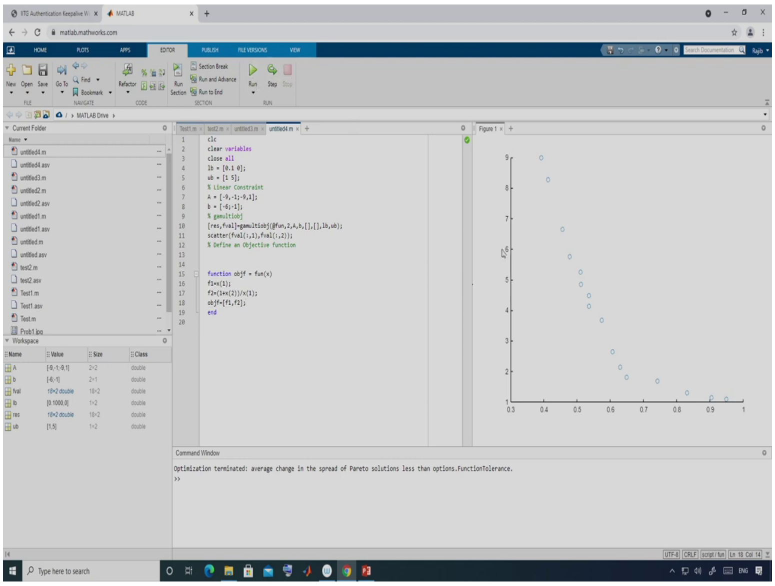



This is the objective function and then I have defined lower bound, I have defined upper bound and then I do not have any non-linear constrains. So, me I can delete this part. So, this is not required actually. So, as I do not have this one. So, I can also remove this part. So, I have 2 inequality type linear constrains.

So, I have defined A. So, A is minus 9 minus 1 then minus 9 plus 1 and the right hand side b is minus 6 and 1 A equality b equality is not there. So, I can also define here. So, I do not have A equality and b equality. So, lower bound upper bound I have defined and so, I can remove this part in order to reduce the number of lines. So, I have defined what is A and what is b. So, that I have defined and then objective function is your this one. So, I can also directly define this objective function here.

So, I can yeah so, I can define here. So, I have defined the objective function that is your fun ok then number of variables is 2 then I have defined A I have defined b and after that so, I am plotting the Pareto optimal front using this particular line. And in the objective function I have only two objective that f 1 equal to x 1 and f 2 equal to 1 plus x 2 by x 1 and this is f 1 plus f 1 and this is f 1 and f 2. So, this is the objective function. So, let me run this particular file yeah.

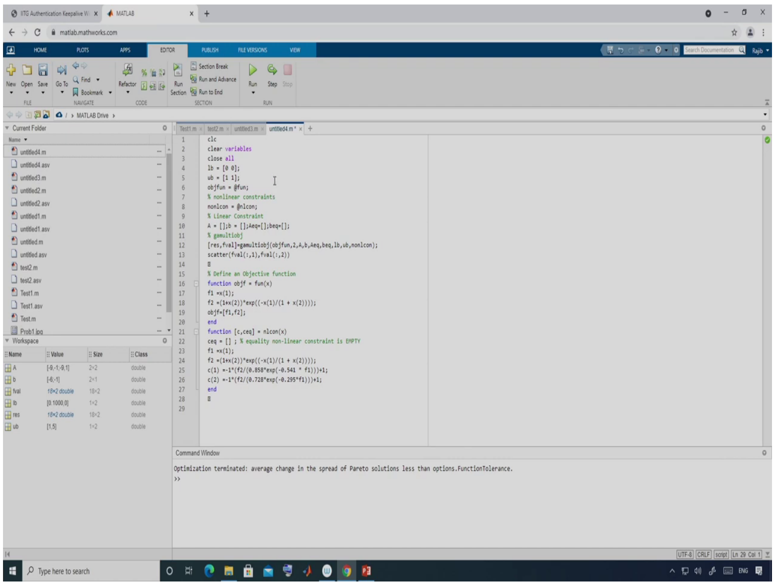
(Refer Slide Time: 25:10)



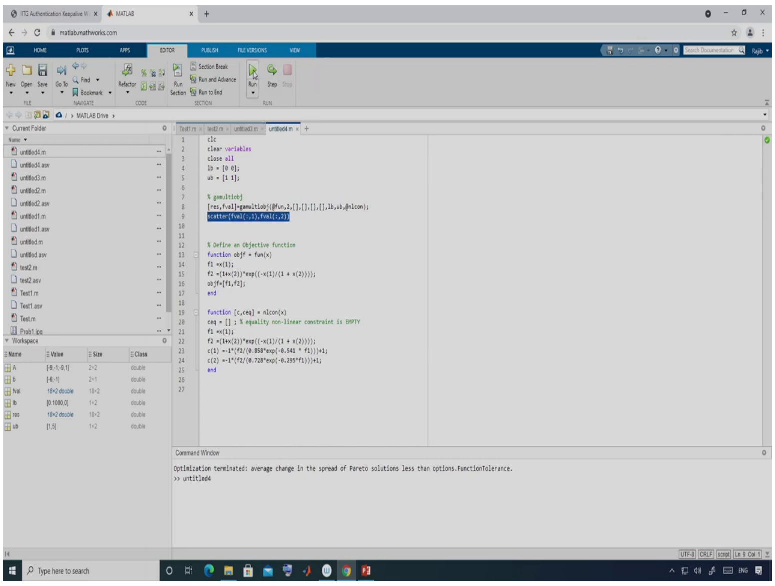

So, you are getting this Pareto optimal front. So, if you increase your population. So, you will get a continuous line. I hope this is clear. So, if you have linear constrain. So, you can define that A b A equality b equality. So, in this particular problem we have only two inequality type linear constrains. So, that I have defined what is A, what is b basically.

So, and other things are similar to what I have solved earlier. Now, let us go to the next problem the this problem we have. So, this problem we have two non-linear inequality type constrain. So, therefore, I have to write the constrain function. So, let me go to the m code. So, this is the m code ok so, MATLAB code dot m file.

(Refer Slide Time: 26:08)



(Refer Slide Time: 26:27)




So, this is the MATLAB code. So, here first I have to define the objective function. So, objective functions are defined here. So, this is the objective function. So, here this is the f 1; f 1 equal to x 1 and f 2 equal to 1 plus x 2 into e x p. So, minus x 1 divided by 1 plus x 2 and we have two objective. So, we are returning this value of f 1 and f 2 and you also need to define the non-linear constrain here.

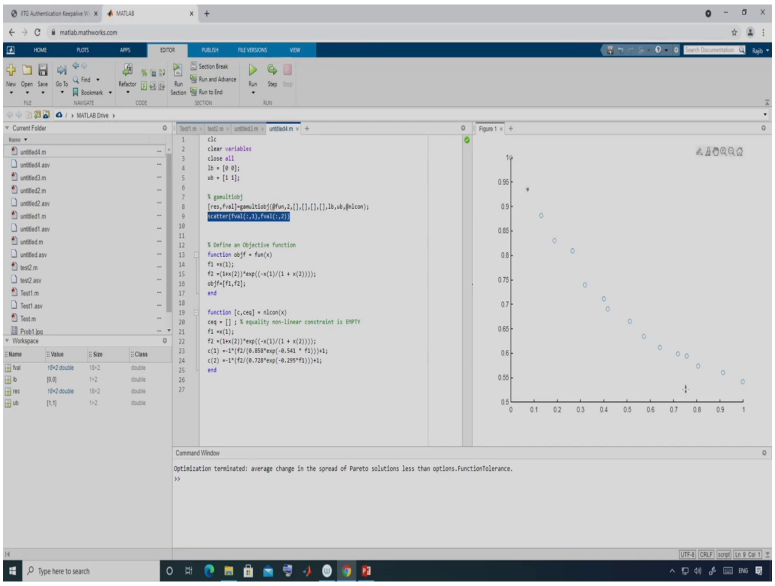
So, we do not have any equality type constrain. So, therefore, we are returning an empty vector and here we are defining f 1 and f 2 and after that we are defining c 1 and c 2. So, you please remember that original inequality type constrain was greater than equality type constrain and, but here we have converted that greater and equality type constrain to less than equality constrain by multiplying minus 1.

So, you can see that one. So, this is the minus 1 I have multiplied ok. So, now, these two constrains are inequality type constrain. So, I have defined so, now, I am returning c and c equality type. So, c equality is null and then you define lower bound you define upper bound then this is the objective function.

So, I can directly write the function here ok. So, you can directly write the function I would like to reduce the number of line ok. And non-linear constrain. So, I can also define the non-linear constrain here. This is the non-linear constrain ok. So, I am defining here.

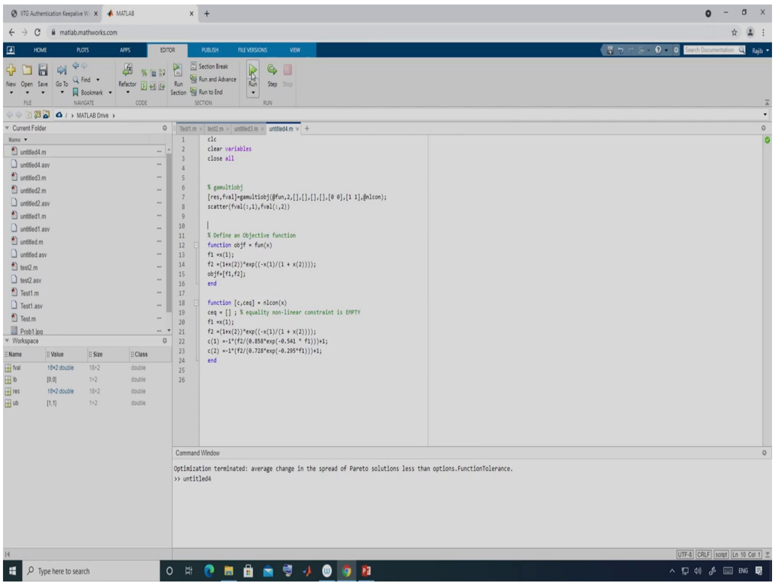
So, I can remove these two lines and we do not have any linear constrain. So, therefore, I will define it here. So, this is null, this is empty, this is also empty this is also empty ok. So, this is not required. This is the objective function and number of variable is 2 then this is not there then lower bound and upper bound I have defined and then the non-linear constrain and after that I am just plotting the Pareto optimal front.

(Refer Slide Time: 29:16)



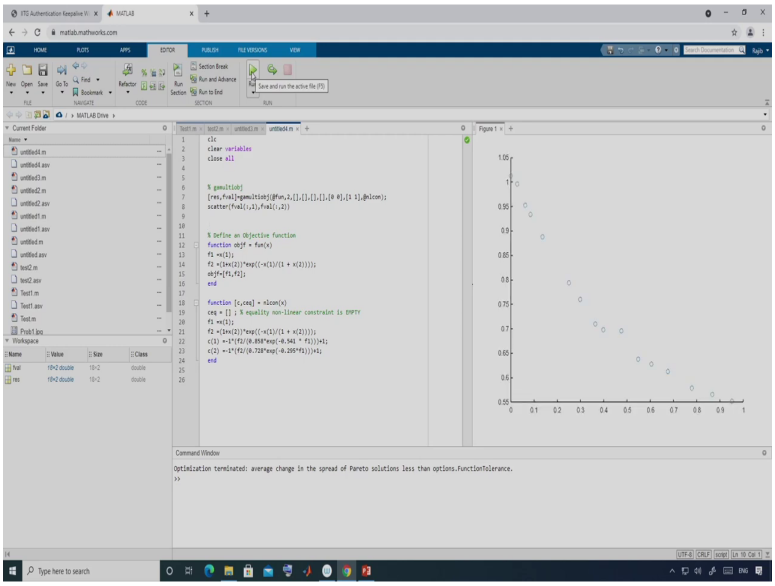

So, let me run this particular m file. So, I am getting this Pareto optimal front.

(Refer Slide Time: 29:29)



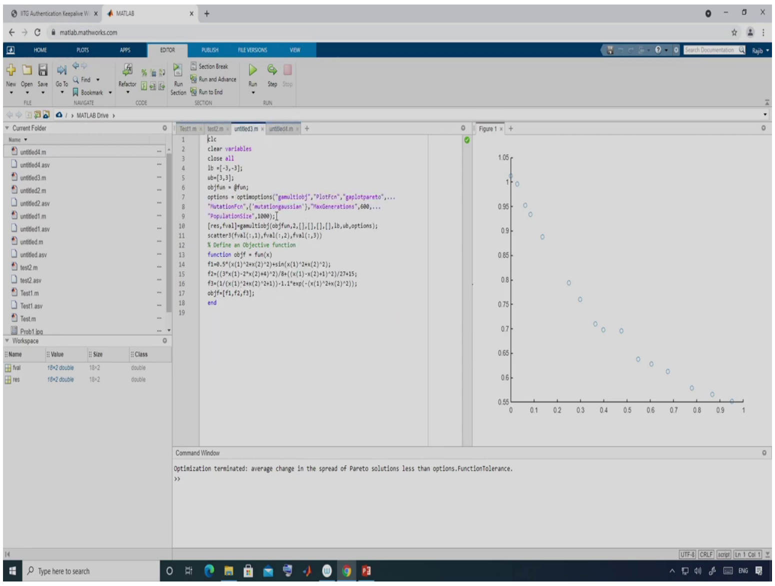

So, what also you can do that this lower bound I can define here also ok. So, I can define lower bound this is 0 and 0 and upper bound also I can define here this is 1 and 1 ok. So, I can define here. So, I can remove this two lines you just see this is quite simple. So, here itself you can define everything.

(Refer Slide Time: 29:59)

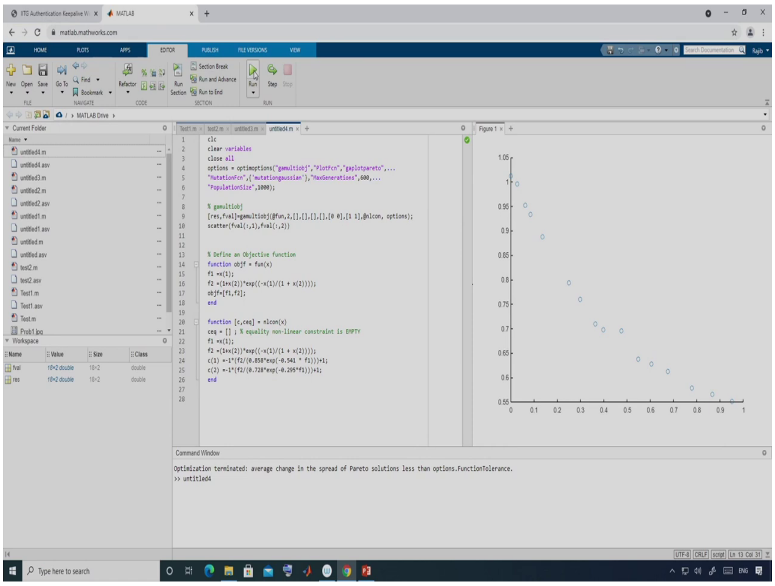



Now, let me see let me execute this particular line. So, yeah so, this is working you just see.

(Refer Slide Time: 30:10)

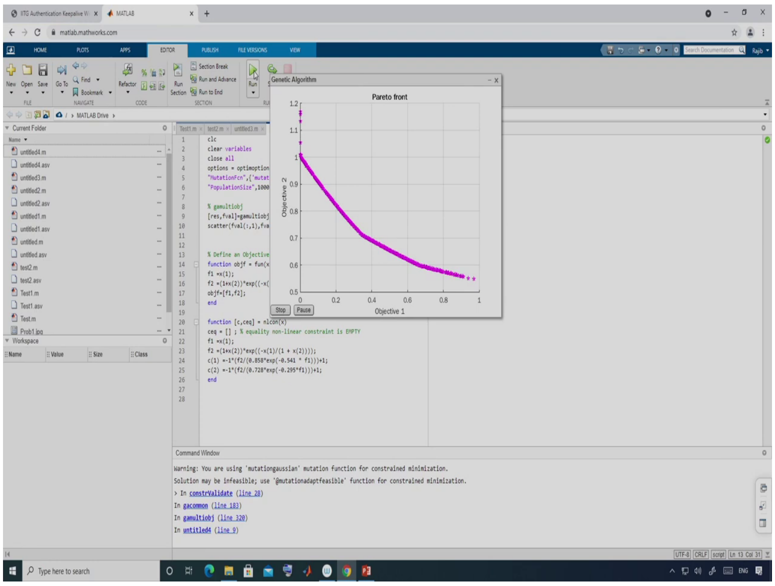


(Refer Slide Time: 30:19)



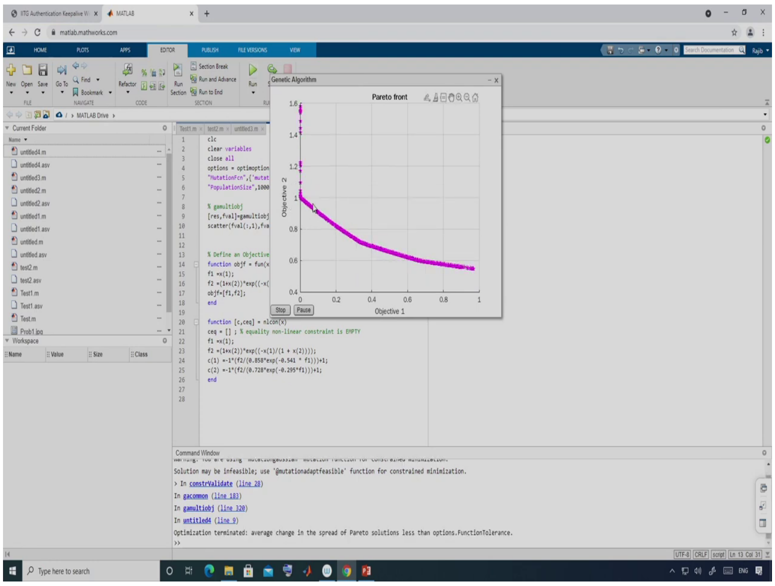

Now let me use the option; so, what I can do? I have suppose I would like to copy this option I would like to increase the population size. So, option I will define here and here I will write option here you are I will write option ok. So, now, let me run this one yeah.

(Refer Slide Time: 30:34)

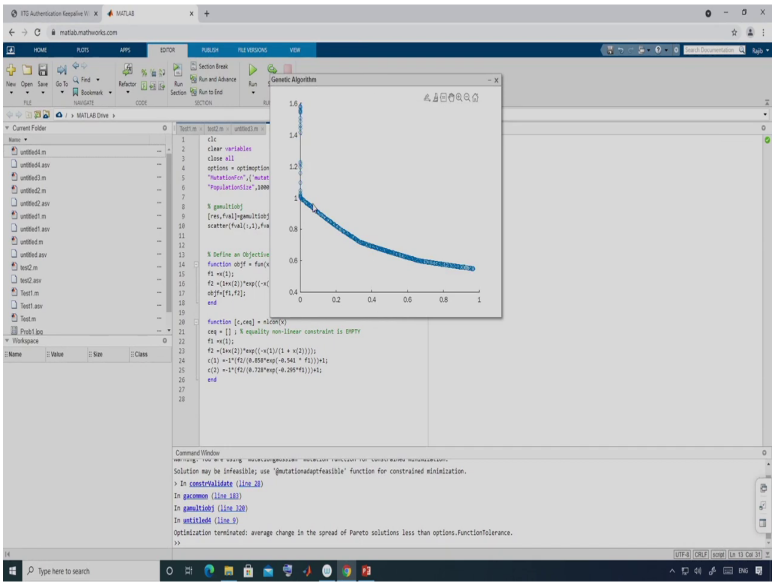


So, I am getting this plot at different generation ok.

(Refer Slide Time: 30:40)

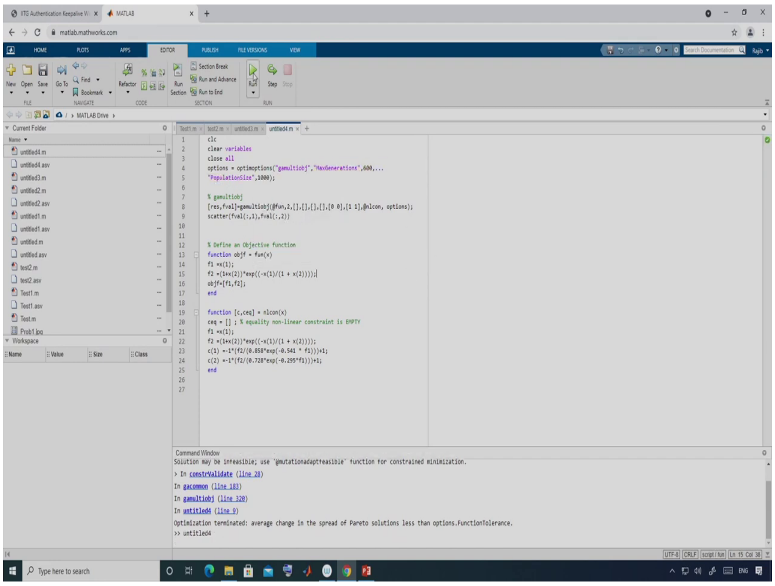


(Refer Slide Time: 30:43)



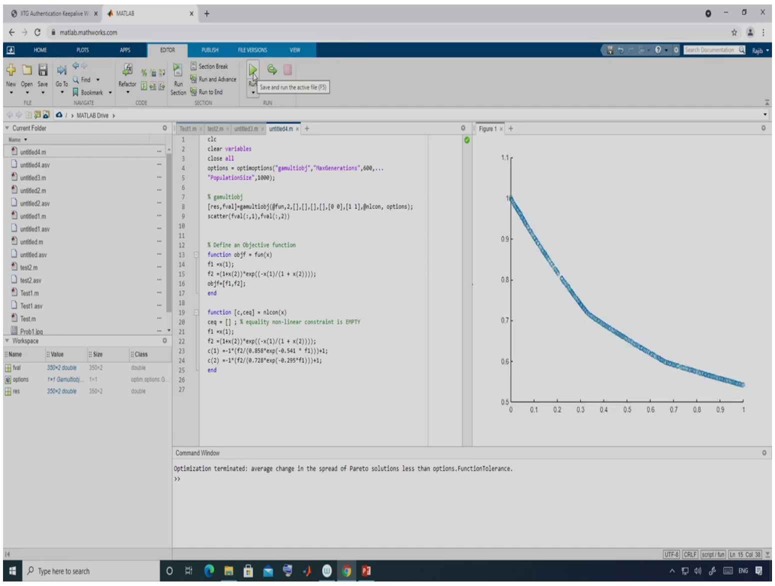

So, you are getting this particular plot.

(Refer Slide Time: 30:54)



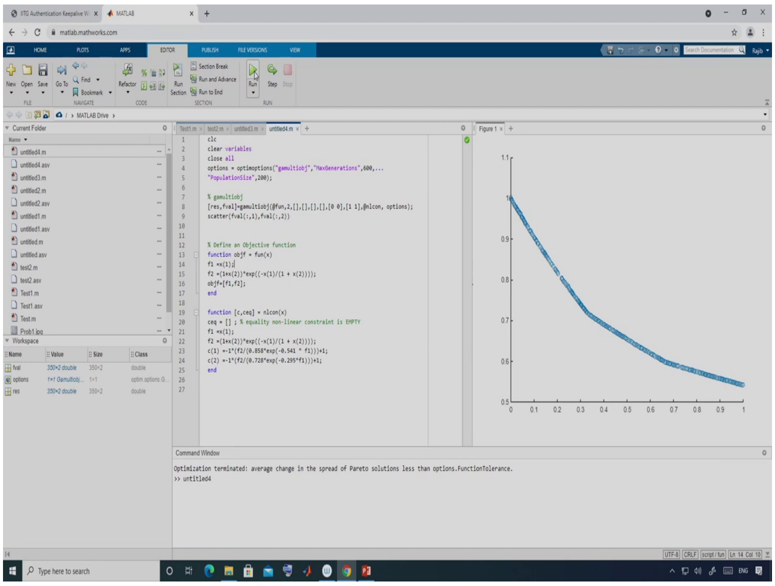

Now let me see I will just I would to like to increase this one ok.

(Refer Slide Time: 31:02)



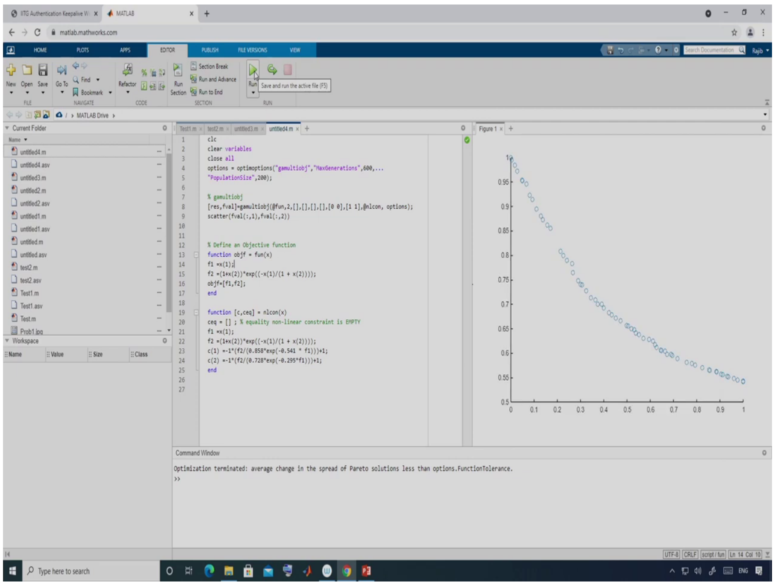

So, I am getting this Pareto optimal front and population size I have used 1000.

(Refer Slide Time: 31:11)



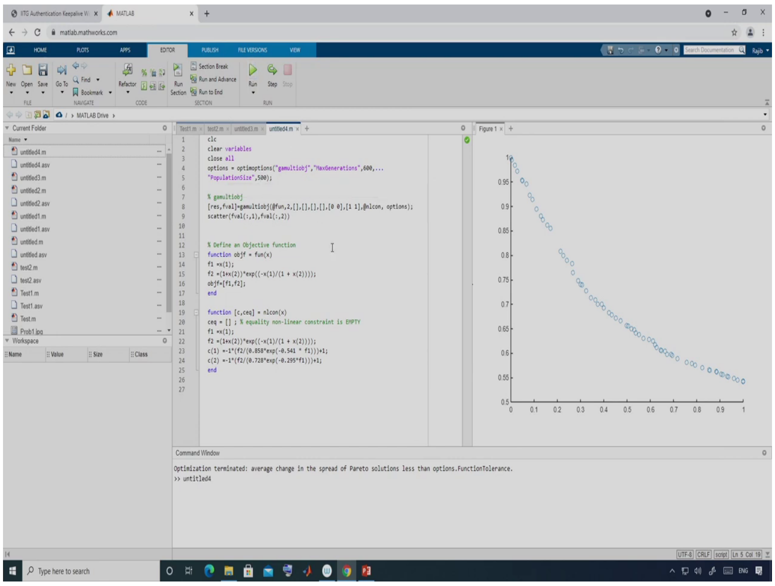

But I can also use suppose what will happen 200 and generation is 600 you keep. So, let me see yeah.

(Refer Slide Time: 31:21)



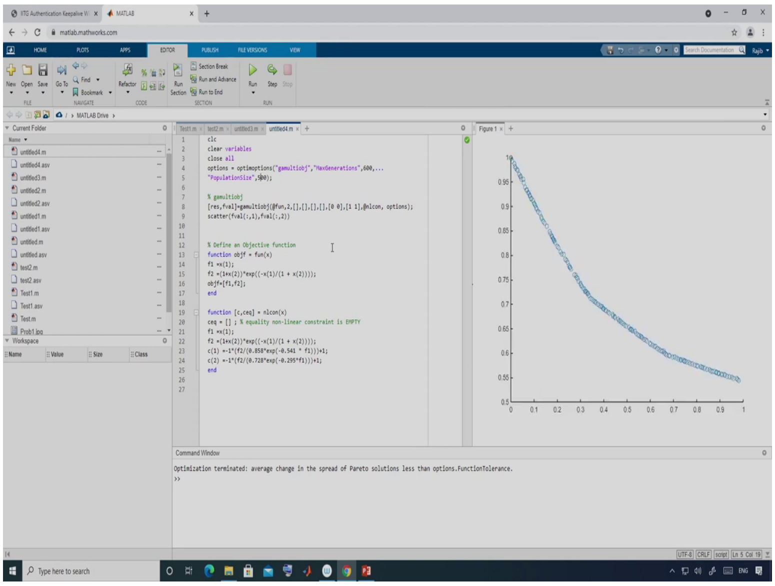

So, I am getting this. So, I am not getting continuous one.

(Refer Slide Time: 31:33)



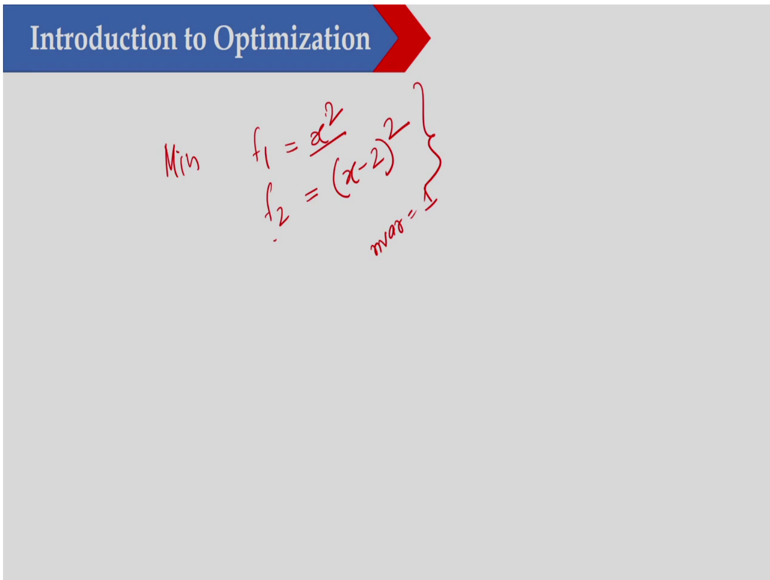
So, but if you are increasing population size suppose 500 I am using. So, in that case you are getting a continuous line.

(Refer Slide Time: 31:38)




So, you are getting a continuous line. So, you can change the population size, you can change the iteration and you can also change the parameters using the option. So, now, let me solve another simple problem without this thing. So, I would like to write the code here itself. So, let me open this part another m file.

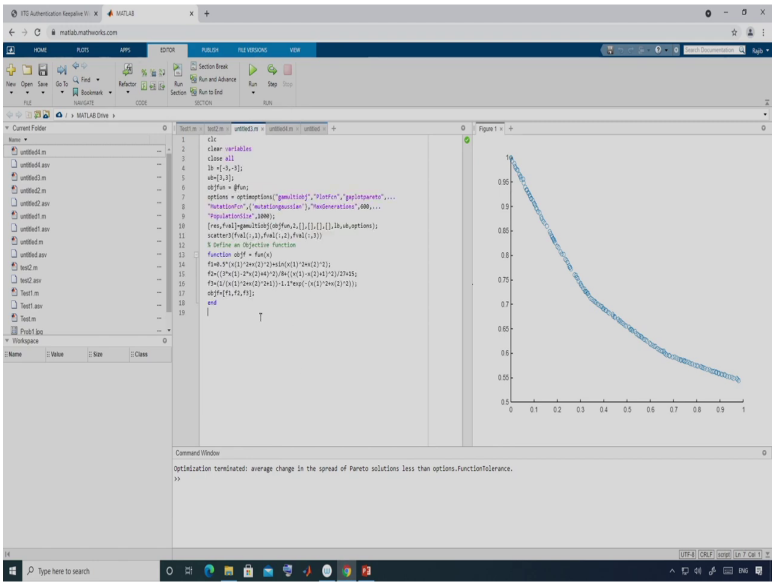
(Refer Slide Time: 32:10)




But before that the problem here is that. So, I would like to solve a problem. So, this is the minimization type problem and we have 2 objective function that f 1 equal to x square. So, this is f 1 equal to x square f 1 equal to x square and f 2 equal to x minus 2 whole square.

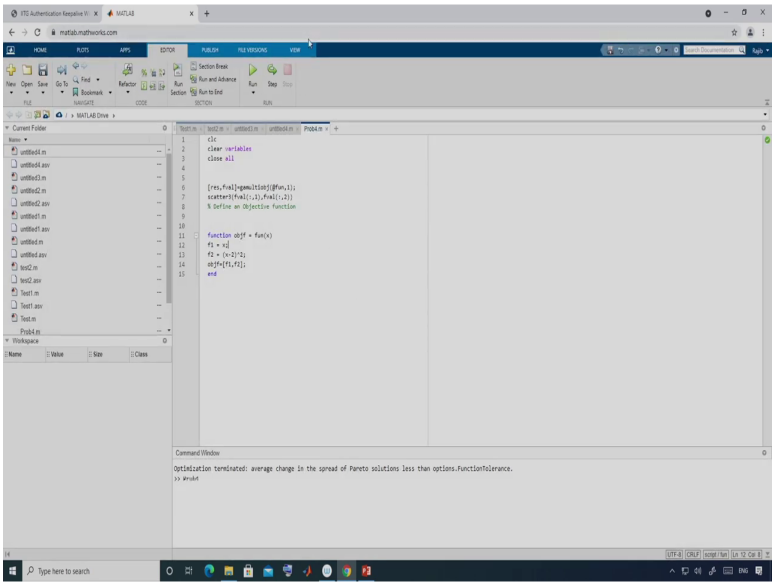
So, this is a very simple problem and the objective function are minimization types. So, minimize f 1 which is equal to x square and f 2 equal to x minus 2 whole square. So, number of variable here nvar is equal to 1 and we do not have any constrain ok. So, there is no constrain. So, this is a very simple problem. So, let us solve this particular problem.

(Refer Slide Time: 33:04)




So, what I will do? I will just copy this one and then I will write ok.

(Refer Slide Time: 33:10)

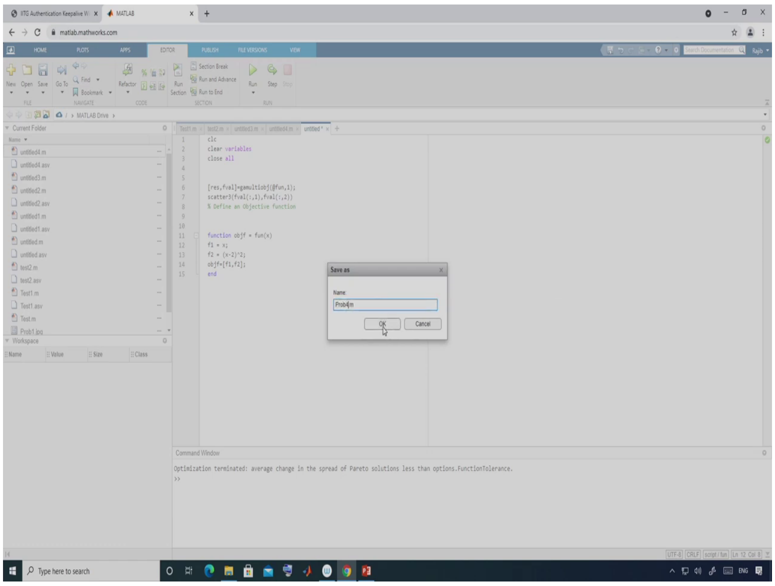



First what I have to do? I have to define this is not required ok. So, I have to define f 1 and f 2. So, here I have to define that f 1 equal to x basically ok and f 2 equal to x minus 2 whole square. So, this is quite this is very simple. So, we have two objective functions that is f 1 and f 2. So, we do not have third one.

So, this is my objective function code. So, you can give any name here. So, I am giving the name of the this particular function is fun f u n ok. So, I have defined what is first objective, I have defined second objective and then I am returning this value f 1 and f 2. So, here now what I will do? I will use this particular function that is A multi objective. So, here the objective function is this is at the rate f u n ok.

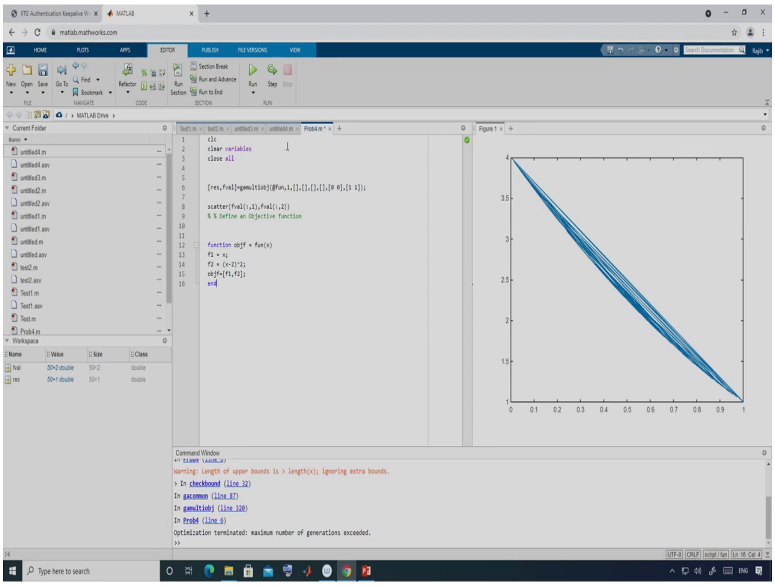
And number of variables is 2 and I do not have anything. So, let me delete this part and just see whether I can solve this particular problem ok. So, then these are just to clean the environment ok. Now, I hope I have defined and this is I have only objective function and number of variable is 1 here ok and then I am just plotting. So, we have two objective function that is first one and second one. So, let me delete this one and run this.

(Refer Slide Time: 35:10)

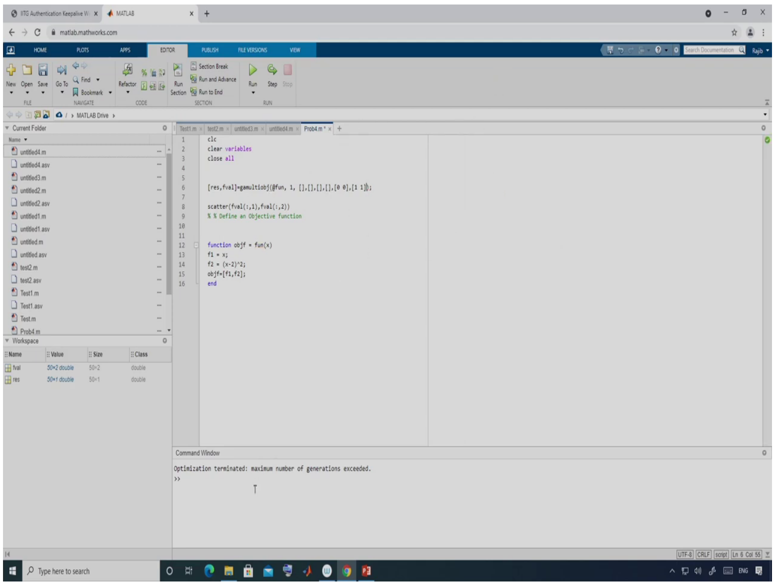



So, I have to give a name. So, this is problem 4.

(Refer Slide Time: 35:24)

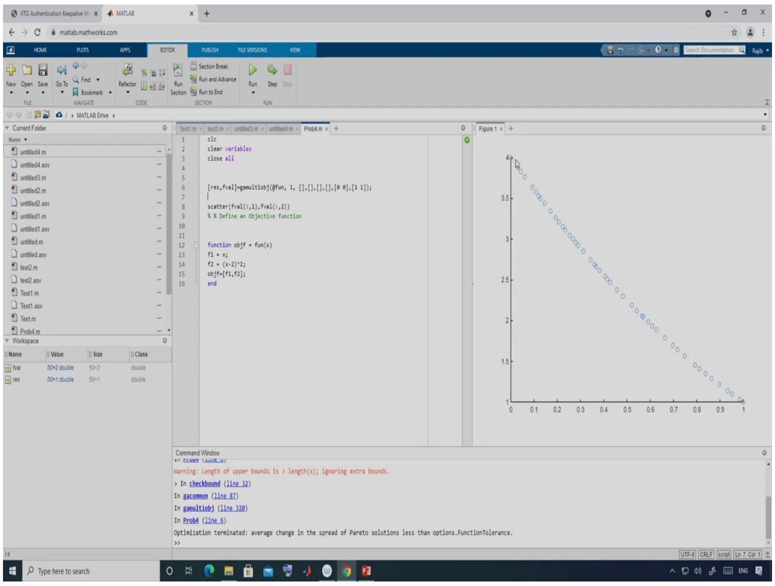


(Refer Slide Time: 36:36)




So, here now I have defined the objective function. So, objective function is fun. So, we have only one variable and then A equality A b A equality b equality is not there and then I have defined lower bound and upper bound here. And then I am just plotting this one. So, variable 1 I am just plotting and this is objective function 1 and objective function 2. So, let me execute this particular MATLAB code. So, I should get the Pareto optimal front. So, I am getting the Pareto optimal front here ok.

(Refer Slide Time: 37:22)




So, I am getting the Pareto optimal front. So, this is all about today's class. So, in today's class we have solved few multi objective optimization problem using MATLAB. So, we have used ga multi objective function for solving the multi objective optimization problem without constrain with linear constrain and with non-linear constrain both equality type and inequality type.

Thank you.