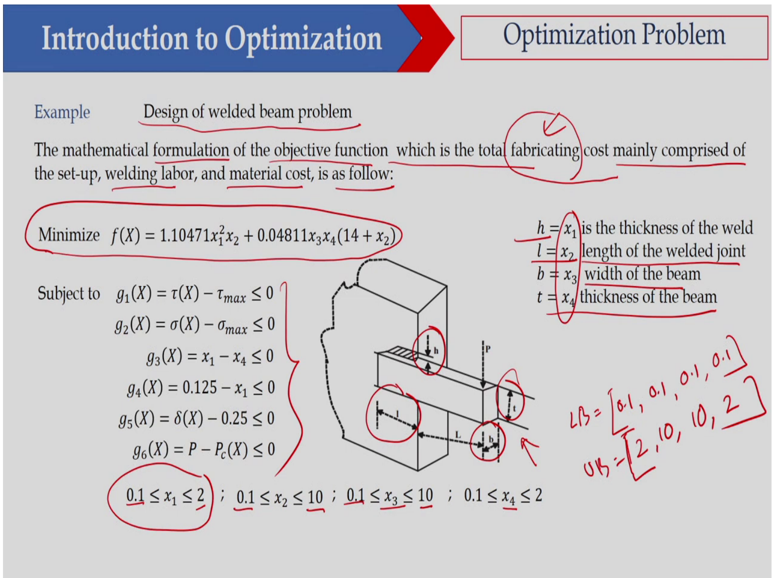
Optimization Methods for Civil Engineering
Prof. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 38
Solution of Optimization Problem: An Civil Engineering problem

Hello student. Welcome back to the course on Optimization Methods for Civil Engineering and this is the last class of this particular course and in today's class, so, I will solve a problem so using different methods. So, initially I will describe the problem here and then I will solve this problem using genetic algorithm then using classical method and then using particle swarm optimization PSO method. So, let me explain this problem and then I will go to MATLAB.

So, this problem I will solve using MATLAB, but you can also solve this problem using R software. So, only thing is that you have to write the code and then you can use the GA package available in R or classical optimization package available in R or PSO optimization package so, that you can use to solve this problem. So, let me explain the problem. So, this problem is a welding design problem of a beam.

(Refer Slide Time: 01:44)



So, this is the problem. So, this problem is design of welded beam. So, design of a welded beam. So, this is the welded beam here I have shown in this particular figure. So, this is the figure and I would like to design this welding ok. The mathematical formulation of the objective function which is the total fabrication cost ok.

So, what I would like to do? I would like to minimize the total fabrication cost ok. So, this I would like to minimize that mainly comprised of the set-up, the welding labor then material cost as follows ok. So, this is basically the total fabrication cost of this join and I would like to minimize this cost.

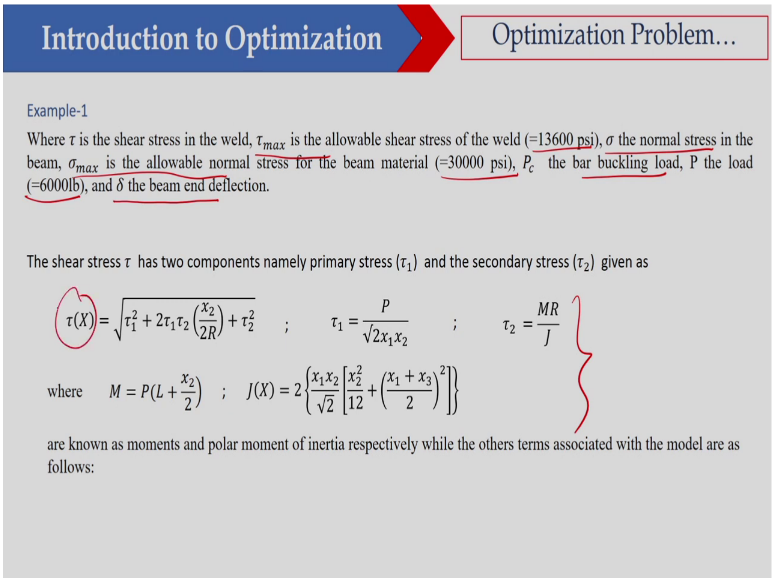
So, objective function can be written like this ok; so, objective function can be written like this. So, here there are four design variables ok x 1, x 2, x 3 and x 4. So, where x 1 is the thickness of the well ok. So, this is your h and that is your x 1 and I have considered variable x 1 then, l is the length of the welded joint. So, this is the length ok. So, this is the; this is the length of the welded joint.

So, this is your the variable 2 and the variable 3 is width of the beam ok. So, this is the width of the beam and variable 4 is thickness of the beam ok. So, for these 4 variables I have to find out the optimal value so that the fabrication cost is minimum ok. So, this is a benchmark problem and many researcher has solved this problem using various optimization method including genetic algorithm, including classical optimization technique, including particle swarm optimization.

So, here I have considered total 6 constrain. So, these constrains are taken in such a way that the shear stress develop is within permissible limit or normal stress develop is also within permissible limit. So, all together we have total 6 constrains. So, I am not discussing about those problem or those constrain here, but if you are taking all these into consideration then I am getting total 6 constrain and this constrains are defined here ok.

So, these 6 constrains are define and lower limit and upper limit of x 1 is 0.1 and 2; that means, range of x 1 is 0.1 and 2. Similarly x 2 is 0.1 and 10, x 3 is 0.1 and 10 and x 4 is 0.1 and 10. So, what I can say that lower bound; so, lower bound here it is 0.1 then 0.1 ok then all are 0.1 so, 0.1 and 0.1 ok. And upper bound here is that, upper bound here is your 2 then 10 then 10 and then 2. So, this is my lower bound and upper bound.

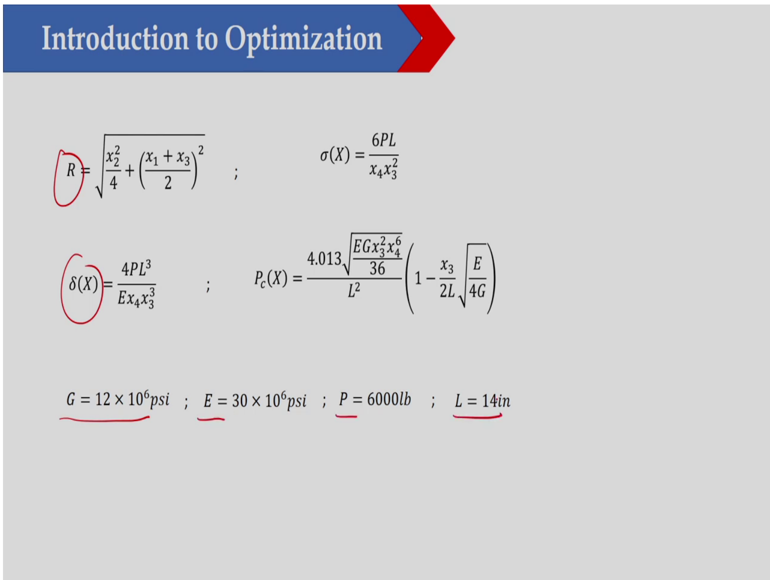
(Refer Slide Time: 05:25)



So, this problem has this particular details. So, where tau is the shear stress and tau max is the allowable shear stress and this value is given and sigma is the normal stress and sigma max is the allowable normal stress that also given and P c is the bar buckling load and the bar buckling load is given here ok.

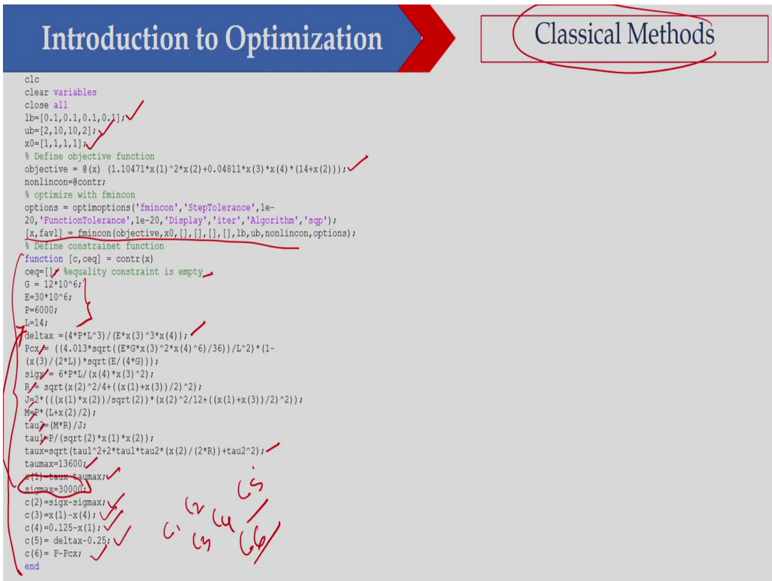
So, this is the bar buckling load and that value is also given ok. So, this is in pound and delta is the beam end deflection. And these are the relation between your shear stress ok. So, other relations are also given. So, I can find out what is tau X within using this relation.

(Refer Slide Time: 06:14)



And, similarly the other equations are also given for R then delta X I can calculate and these are the value of G E P and L. So, length is given that is your 14 inch. So, I can use this equation; I can use this equation to calculate the value of the constrain ok. So, I can calculate the value of the constrain and this constrain has to be satisfied is not it.

(Refer Slide Time: 06:47)



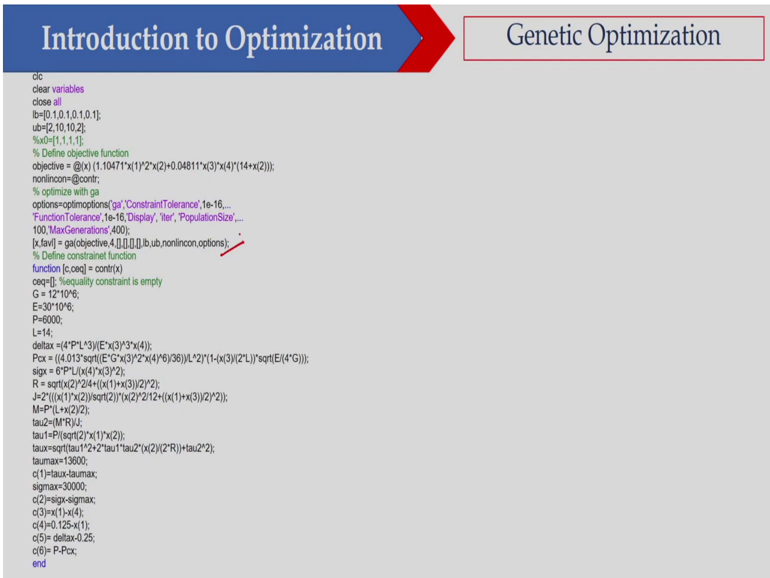
So, this is the MATLAB code using classical method. So, here I have used f min con ok. So, this is the fmincon function I have used and what I have to do I think already you have solved several problems. So, I have to define the lower bound, I have to define the upper bound, I have to define an initial point because I am using the classical optimization method. So, I have I am using fmincon algorithm.

So, therefore, I have to define this initial point and then I have to define the objective function. So, this is the objective function and here there are total 6 constrain and constrain is define in this particular function. So, you just see all these equations are calculated here. So, all these equations are calculated here.

So, we do not have any equality type constrain. So, therefore, this is your null and this is the value of G E P and L is define then you calculate deltax then Pcx you calculate then sigma x you calculate then R, then J then M and then tau2 and tau1 you have to calculate and after that I am calculating the value of taux ok.

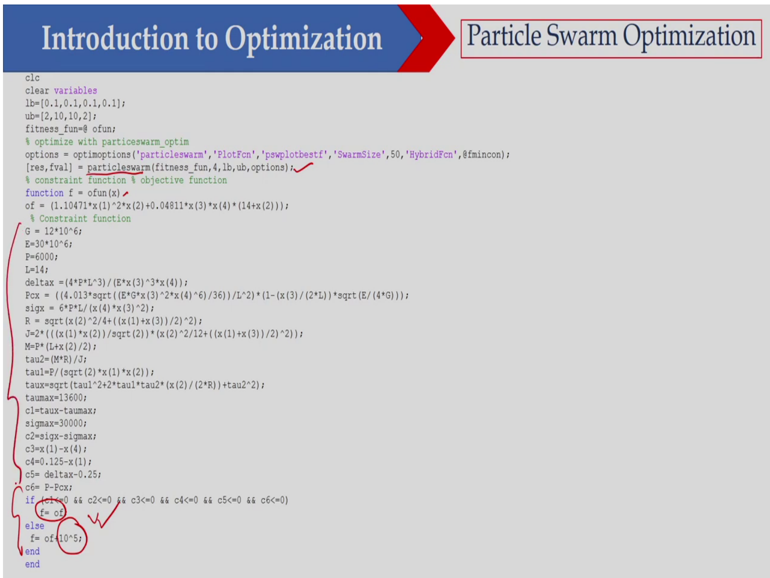
And then taumax is given. So, this is the value and then you define that c 1, c 2, c 3, c 4, c 5 and c 6. So, sigma is also defined here. So, I can define sigma here also ok. So, there is no issue. So, once you are calculating this one. So, this function will give you the value of c 1 then c 2, c 3, c 4 and c 5 and c 6 ok c 6 ok. So, you will get the so, this function will return the value of c 1, c 2, c 3, c 4, c 5 and c 6 ok.

(Refer Slide Time: 08:56)



So, this is in using fmincon and then I have solved this particular problem using ga ok. So, this is the ga function.

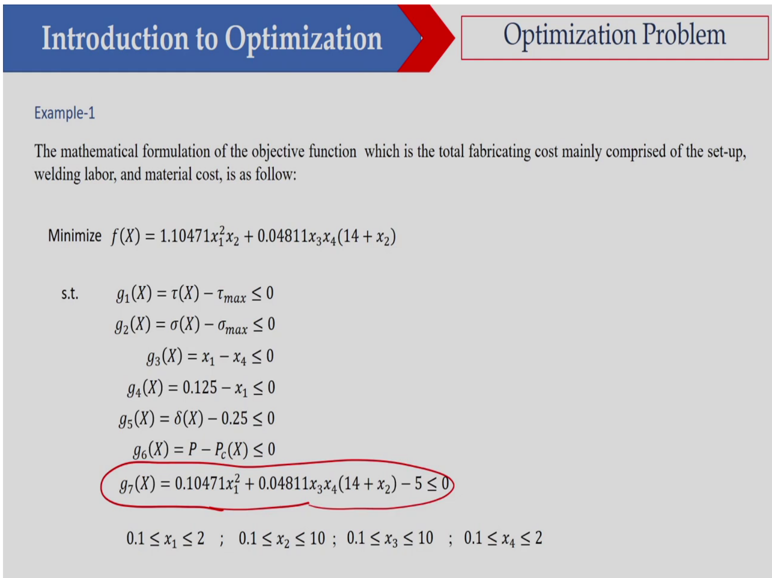
(Refer Slide Time: 09:05)



And then this is solve using PSO ok. So, particle swarm optimization, particleswarm and this is again. So, this part is same ok. So, this part is your same only thing is that I have define this value somewhere here. So, in case of PSO. So, I have to write the objective function code. So, we cannot define the constrain and objective function separately. So, what I am doing here I am writing the objective function and then the constrain.

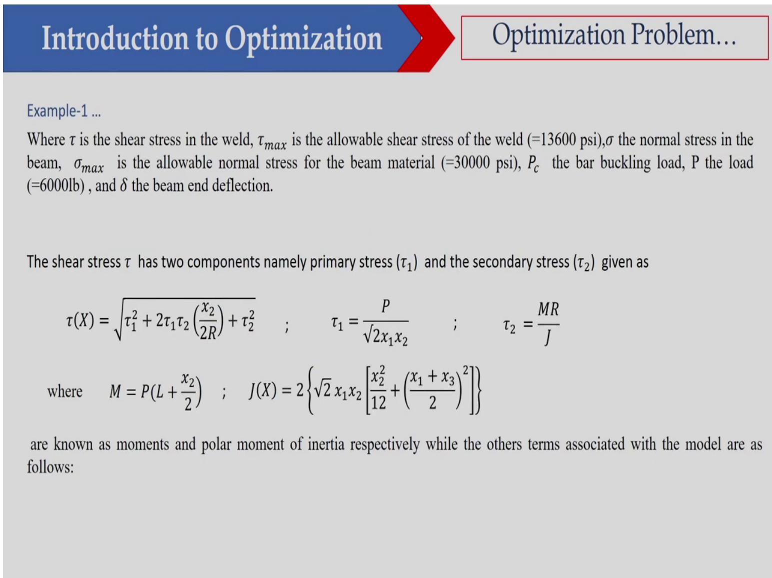
So, if there is no violation then it will return the objective function value; if there is a violation it will return a very large value so, that this particular solution can be avoided ok. So, that we are doing. So, for that I am using if else statement. So, if there is a violation then I am returning this value and if there is no violation then I am returning this value. So, this can be written and this part is similar to what we have already define.

(Refer Slide Time: 10:19)



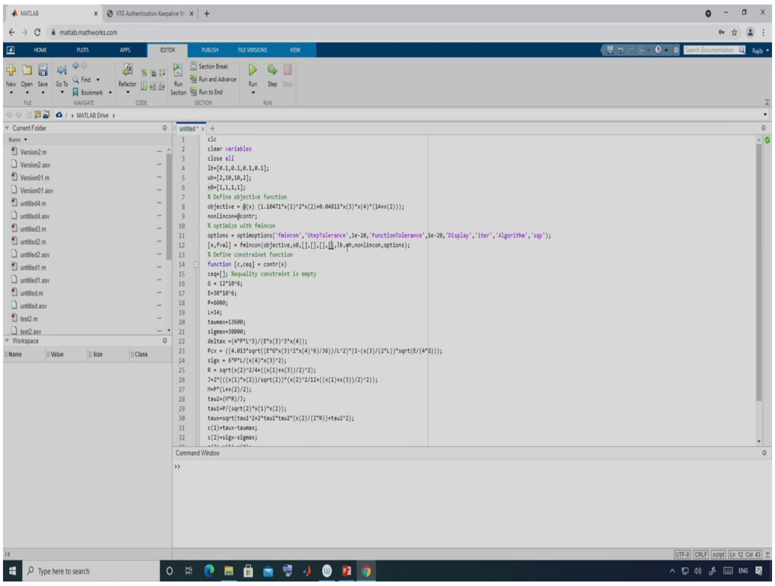
So, there are another version of this problem. So, there is another constrain that is c 7. So, you can see. So, this is another constrain of this particular problem and so I have also; I have also solved this problem using the constrain 7 ok.

(Refer Slide Time: 10:42)



So, constrain 7 is defined here and I can calculate the value of constrain 7 ok. So, this is the code using classical method; using classical method. So, I have used fmincon here and the second one is your ga ok and third one is particleswarm optimization. So, third one is particleswarm optimization. So, let us solve this problem or let us solve let us run this m file in MATLAB.

(Refer Slide Time: 11:18)



So, let us go to MATLAB. So, this is the MATLAB and let me solve the first m file and this is your fmincon classical method ok. So, I can copy this. So, here you can see that I have define the lower bound, I have define the upper bound and this is the initial point 1 1 1. So, I have consider 1 1 1, but let us see if we can if we change it whether you are getting the solution or not then you define the objective function and then you write the constrain function ok.

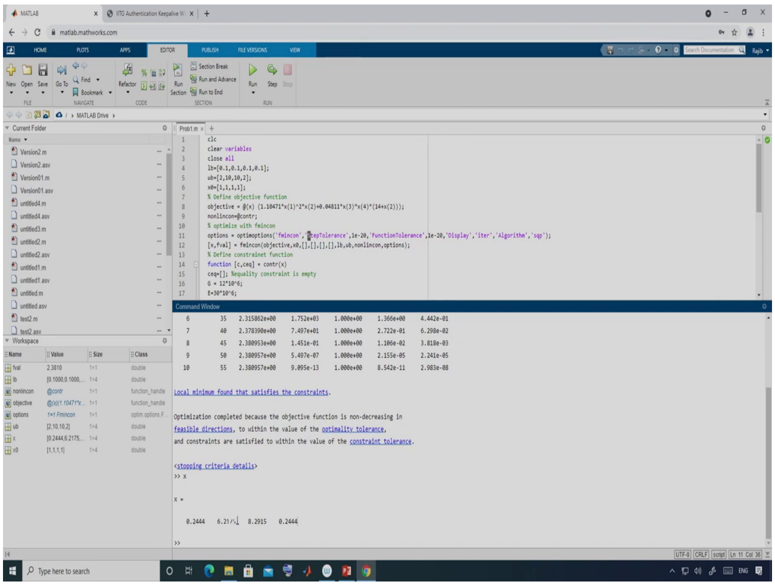
So, constrain function. So, here as I said that ok. So, let me define what is the value of G, what is the value of E, then what is the value of P, then L. So, I can define this taumax here itself. So, I can define here. So, all the parameters I can define here called the constant value. So, I am define I am defining all these ok.

So, this is so, I have define ok then I have calculated deltax then Pcx then sigma x R, J, M then tau 2 then tau 1 and finally, I am calculating taux. So, once you are getting that 1 then you can define the constrain 1, 2, 3, 4, 5 and 6 all together 6 constrain and basically and there is no equally type constrain. So, I am putting empty here ok.

So, after that I can simply run this particular line and this is your fmincon ok. So, this is your fmincon and here this is the objective function then I am defining x naught then a b a equality is not there then lower bound I have define, upper bound I have define then non linear constrain I have define and I have also define the option. So, in option I have set the tolerance limit ok since the tolerance limit to e to the power minus 10 then function tolerance this then display iteration.

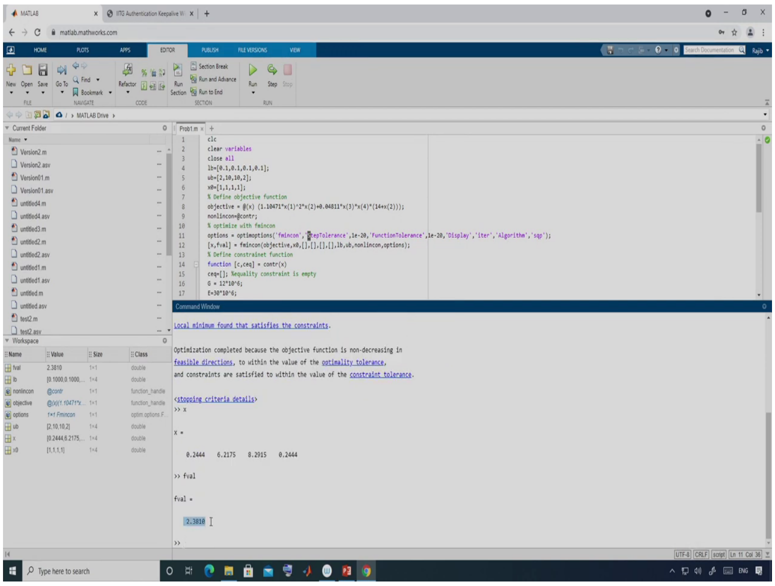
So, I would like to see the iteration value and algorithm also I can define this is sequential quadratic programming as qp ok. So, you can also define the other algorithm so, but here I am using this one. So, let me execute this one. So, before that, I have to solve these problems. So, I have I before that I have to save this file. So, this is problem 1 ok. So, this is n dot m, dot m yeah.

(Refer Slide Time: 14:46)



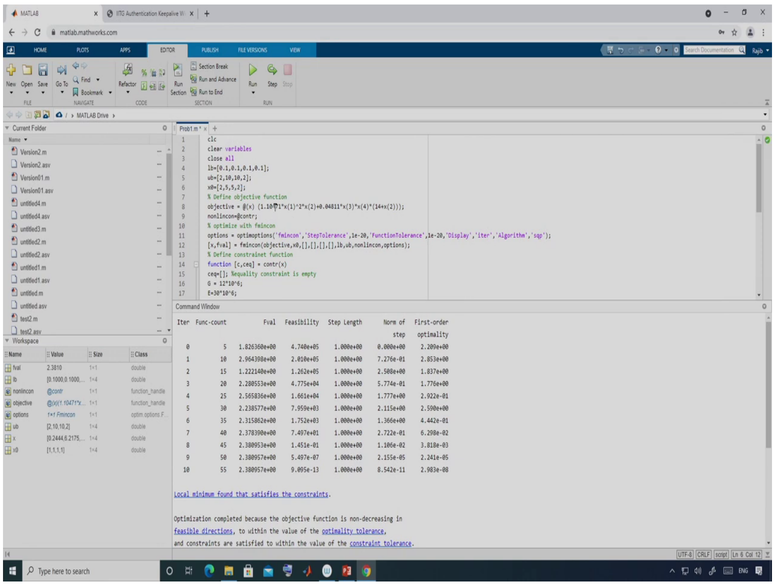
So, now I got the solution ok. So, I got the solution and you can see these are the iteration also and finally, I am getting the solution ok the solution value is that value is 2.38 ok. So, I can check what is the solution that x value so this is the solution that x 1 is 0.244, then x 2 is 6.21 and this is 8.29 and this is 0.244.

(Refer Slide Time: 15:21)



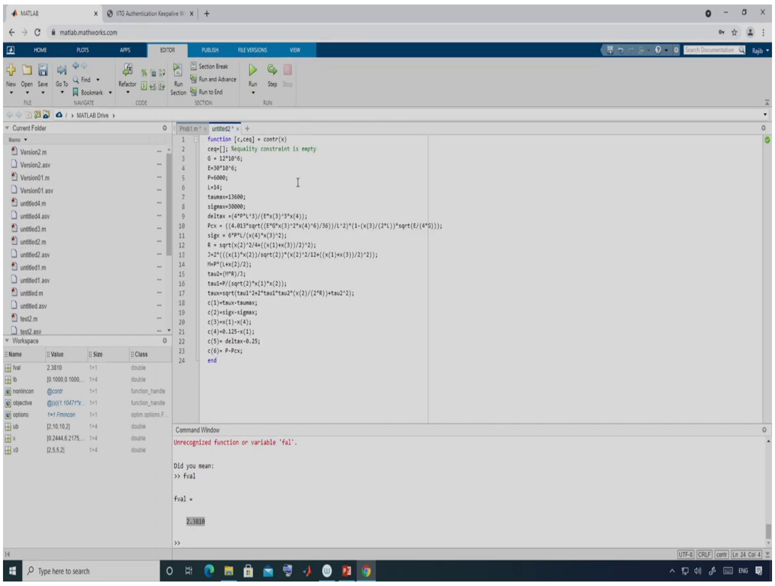
And the function value you can see and this is the function value 2.38 ok and as I said the other researcher also solve this problem and they also got similar solution. Now, let me see if I change the initial solution whether I am getting this solution or not ok.

(Refer Slide Time: 15:41)



So, let me put it suppose if I put 2 and if I ok no this is basically I this is 2. So, let me and this is I am putting 5 ok. So, this is also I am putting 5 and this is 2 ok.

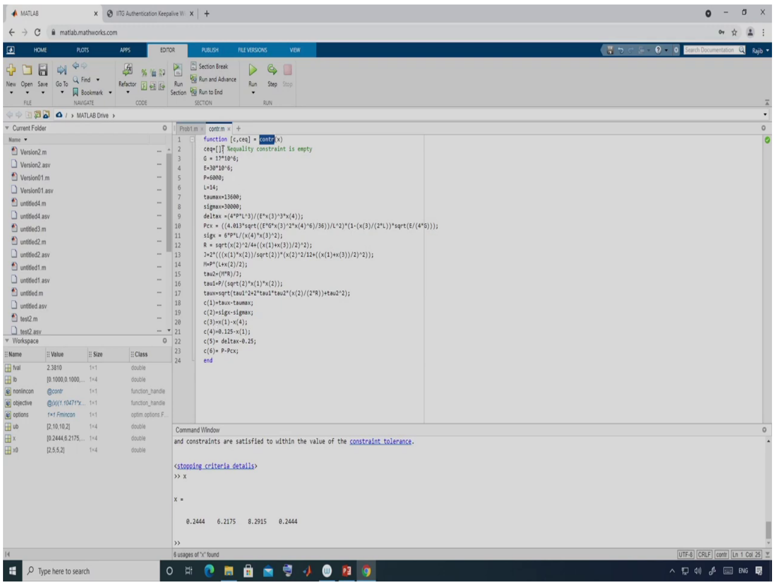
(Refer Slide Time: 16:11)



So, let me run it yeah I am getting the solution this time also. So, this is not that sensitive to initial solution ok so, ok. So, I am getting this is 2.38. Now, I would like to explain one thing. So, here I have define the function here ok. So, this is the function I have define. So, I can also write this function somewhere else ok. So, I can save this file as an m file ok. So, this is a function and let me save it. So, I will save it as contr ok.

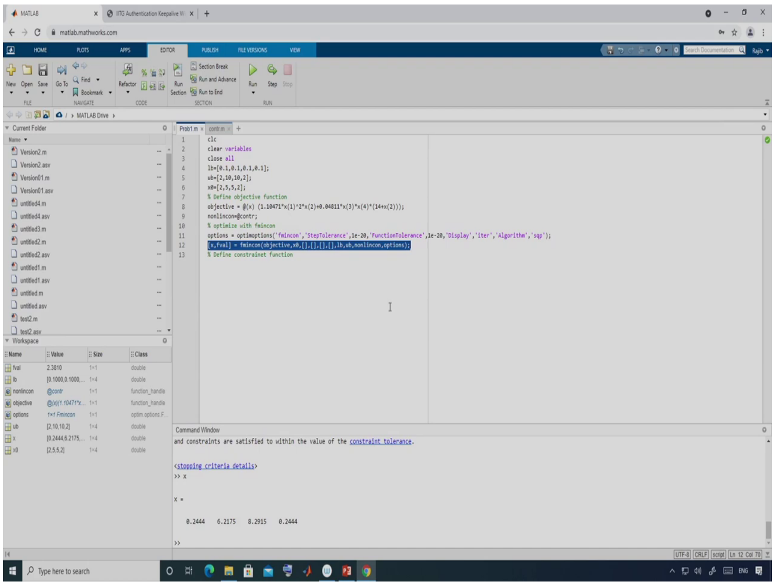
So, I am saving in a separate file. So, let me see whether it is working or not. So, it should work. So, if I execute this particular m file ok. So, it is working ok. So, I am getting the solution. So, therefore, what you can do for objective function or constrain function you can create a different m file and that you can call and it should be in the same directory ok. So, if it is on the same directory. So, you can save as a different file also.

(Refer Slide Time: 17:59)



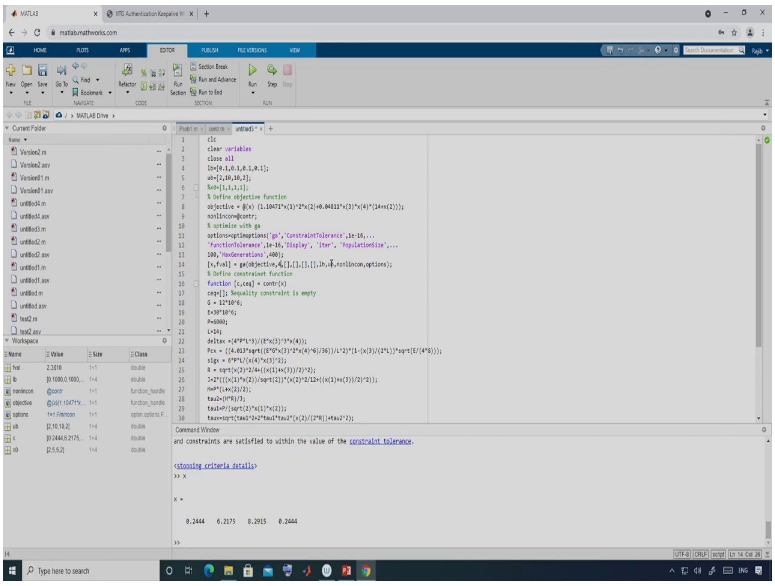
So, here what I am doing, I am defining this is an m file. So, this is a function and the function name is contr ok and the file name should be contr. So, this should be same that function name and file name and then and this particular function will return c and c equality.

(Refer Slide Time: 18:17)



And here I am using this one ok. So, here I am using this one and then basically I am running this particular function ok, to solve this problem using sequential quadratic problem algorithm using sqp is using sqp algorithm. Now, let us solve this problem using genetic algorithm or n PSO. So, let us solve it by genetic algorithm.

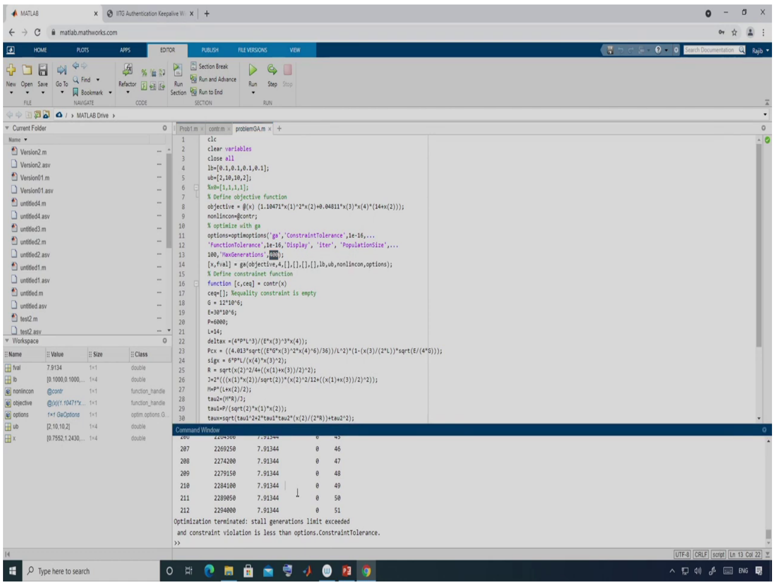
(Refer Slide Time: 18:54)



So, I am just copying this one. So, this is the ga ok. So, this is the ga part and here everything is similar. So, what we have run during the classical method. So, m code is similar. So, only thing is that, that we are using genetic algorithm function ok. So, this is the function. So, this is the function this is the function. The function is ga and I have defined some option here.

So, here the constrain violation I have change then function tolerance, I have change and I am also displaying the iteration and population size I make it 100 and maximum generation I make it 400 ok. And then this is the objective function this is the number of variable then we do not have a, b, a equality, b equality, lower bound, upper bound we have define and then non linear constrain this is the non-linear constrain and we have option.

(Refer Slide Time: 20:11)

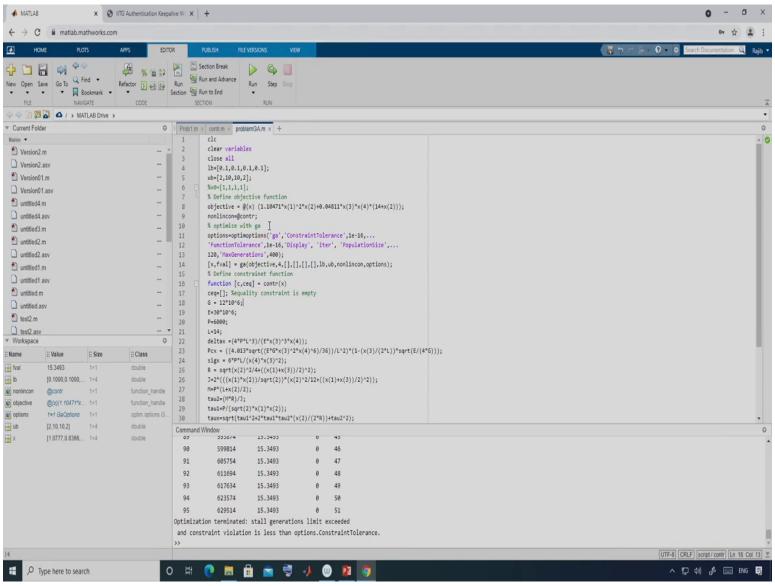


So, this function is same as what we have used in case of classical method ok. So, let me run this thing. So, before that I have solve it. So, this is problem ga, problemGA ok. So, let me yeah. So, you can see that function value is 67 now it is 15. So, it should go up to 2.36 ok. So, it is running. So, it will go up to 400 iteration ok. So, up to 400 iteration.

So, as you have seen. So, when I have applied the classical method. So, you just see within 10 iteration of your classical method. So, I got the solution, but here it is running still. So, this is up to 35 iteration 36; at 36 iteration I am getting 7.91. So, I have to go up to 2.36. So, final value is 2.36 around 2.36 yeah.

You just see it is not reducing. So, the function value is not reducing. So, I am getting the same function value. So, in that case it may happen that we are not getting the solution here ok. So, we are not getting the solution. So, let me run this thing again. So, now, it has reduce up to 4.2. So, in this case also it is I am not getting the solution and I got 4.22 ok. So, let me see again.

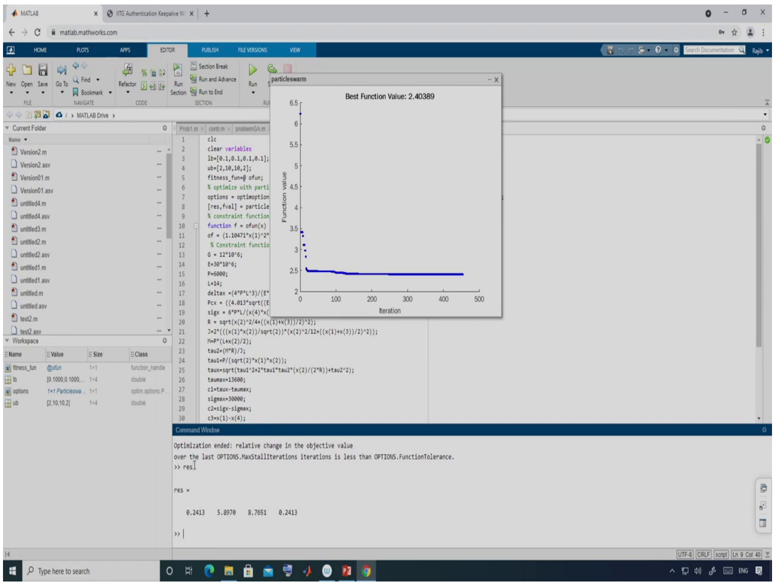
(Refer Slide Time: 24:59)



So, I did not get the solution. So, what I will do I will increase the population size. So, make it 200 just see whether I am getting the solution or not, make it 120 ok. So, I am not getting the exact solution using genetic algorithm. So, and you just see it is taking lot of time to converge it.

So, finally, I am not getting the solution, but if you are running it. So, you may get solution sometime, but therefore, it is better to solve this problem using classical method ok. So, let me see whether I am getting the solution using particle swarm optimization method.

(Refer Slide Time: 26:34)



So, this is the code for particle swarm optimization, let us save it. So, this is using PSO. So, here I have defined this objective function and then I have use the particle swarm optimization ok. So, let me run this thing yeah you just see I am getting the solution. So, here you can see what is the solution, the solution is yeah.

So, I am getting. So, this is not the exact solution, but it is a near optimal solution, this is 0.2372 and so, solution and this is 5.46. So, it should be 6.7 and this is 9.48. So, I am not getting the exact solution, but it is a near optimal solution. So, this is 2.44 ok. So, let me run it again.

So, let me check yeah I am getting little bit better solution, but this is not the solution. So, let me run it again yeah 2.46 again ok yeah 2.39. So, this is what I got when I have applied the classical method yeah this is 2.4. So, swarm size I can use let me see actually. So, if I use 100 then what will happen? Yeah 2.39. So, I should get around 2.36, 2.59 yeah 2.38.

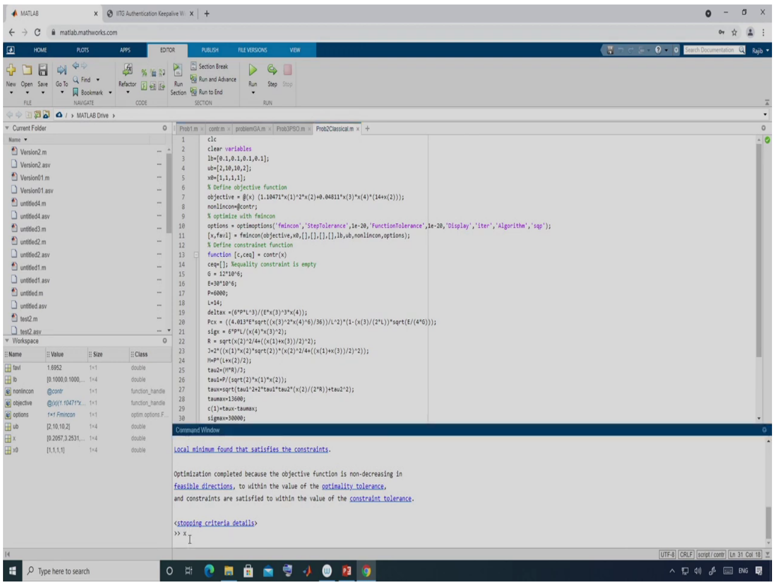
So, you just see when I have applied the particle swarm optimization. So, I am not getting the same solution every time because what is happening; what is happening here that you are starting with some different set of initial solutions and therefore, you are not getting the same solution.

And apart from that so we are not taking the gradient information here and as a result. So, you may not get the exact optimal solution, but you may get a near optimal solution of the problem. So, if you compare the genetic algorithm with PSO. So, you can see that the PSO is much faster than genetic algorithm ok.

Even in this particular case so, I am not getting the solution using genetic algorithm so, but I may get it. So, if I am running it for maybe 10 times or 15 times. So, sometime I may get the solution, but every time I am not getting the solution. So, that means, this algorithm is not robust enough to get the solution in every iteration or something like that ok, but when your problem has more than one optimal solution then, you can use genetic algorithm.

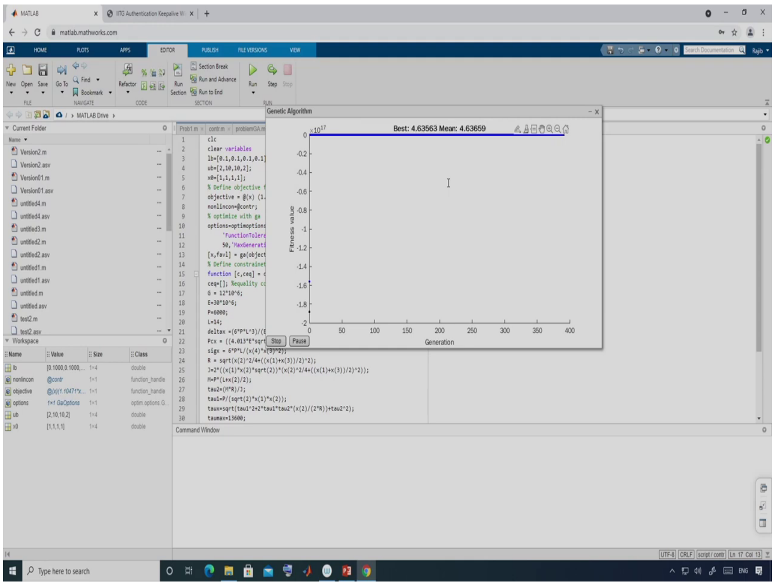
Because at that time it will be difficult to solve the problem using classical method and once you are getting a solution near the optimal solution then you can apply the classical optimization methods ok.

(Refer Slide Time: 31:12)



So, let me go to the second version of this problem. So, this is the second version of this problem and this is the MATLAB code. So, let me copy this thing ok. So, here I have use the classical method ok. So, this is the second version. So, we have another constrain 7 ok. So, let me execute this one.

(Refer Slide Time: 31:59)



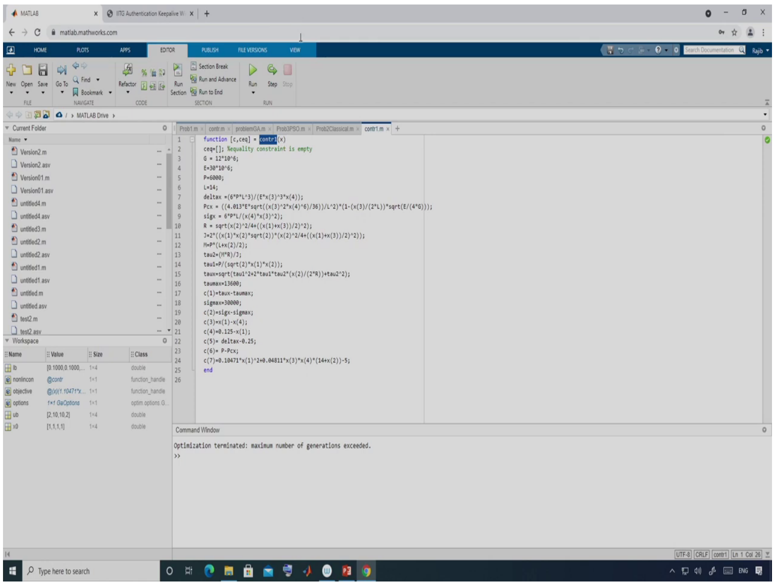
So, let me see what is the solution and this is the solution and objective function value is. So, objective function value is 1.69 ok. So, yeah so, I am getting this is the solution of this particular problem 1.69 ok. Now, let me solve this problem using ga let us see whether I am getting the solution for this one. So, I am applying ga here. So, let me see. So, here I am plotting this fitness function versus generation curve and you can see it is somewhere here.

So, the base fit solution is 5.6. So, now, it is 4.63. So, you just see the function value is not reducing and here the base solution and mean solution is almost same; that means, the populations are saturated with the best solution and we may not get any improved solution because crossover will not create any new solution.

So, therefore, so, we may not get any solution and this loop may be terminated. So, now, you just see the best and min solution is same so, that means, we are not getting any new solution.

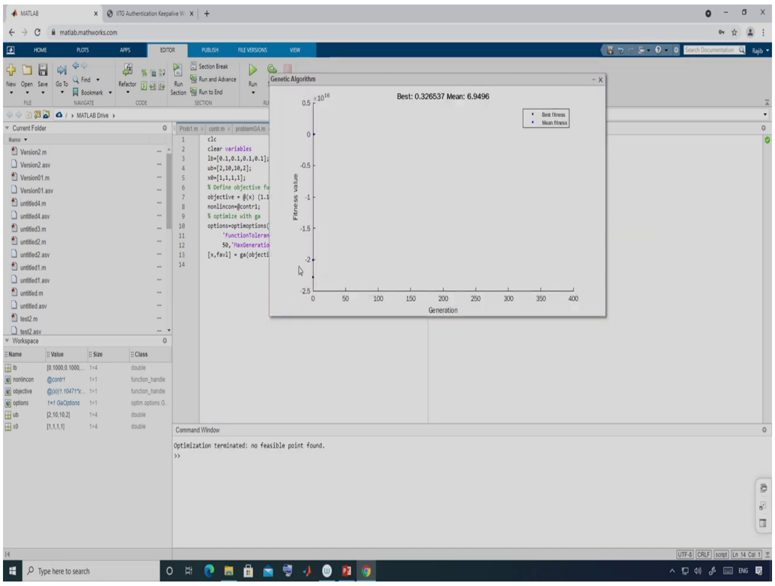
So, it will go up to 400 iteration, but the function value is not reducing ok. So, I did not get the solution.

(Refer Slide Time: 37:36)



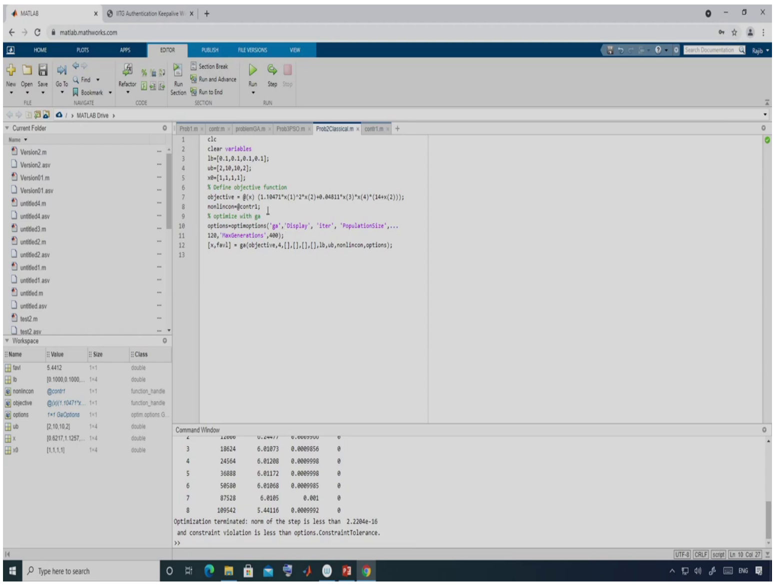
Let me try one thing. So, let me delete this part because I have already defined this particular problem. So, so, let me delete this part ok. I am defining a different function yeah. So, this function I am defining in a different file. This is 1 now let me save it contr1, ok.

(Refer Slide Time: 38:51)



So, now I will be using this function here ok. So, this is 1 ok. So, let me try. So, I am not getting any solution.

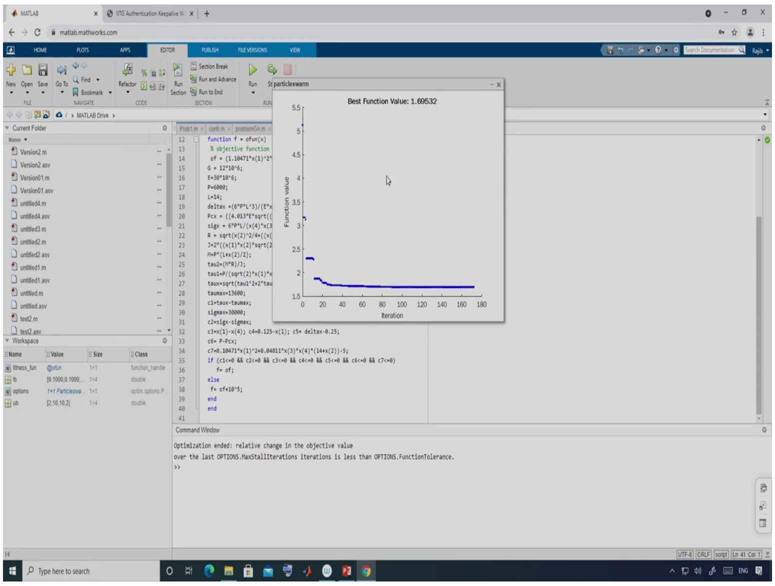
(Refer Slide Time: 40:30)



So, just see. So, let me check. So, now, it is reducing 5.65. So, let us see whether we are getting the solution here or not. It is changing, but it is changing very slowly. So, in this case also we may not get the solution when you have applied genetic algorithm. So, it will go up to 400 iteration. So, there is a small change, but actually it is not converging at optimal solution.

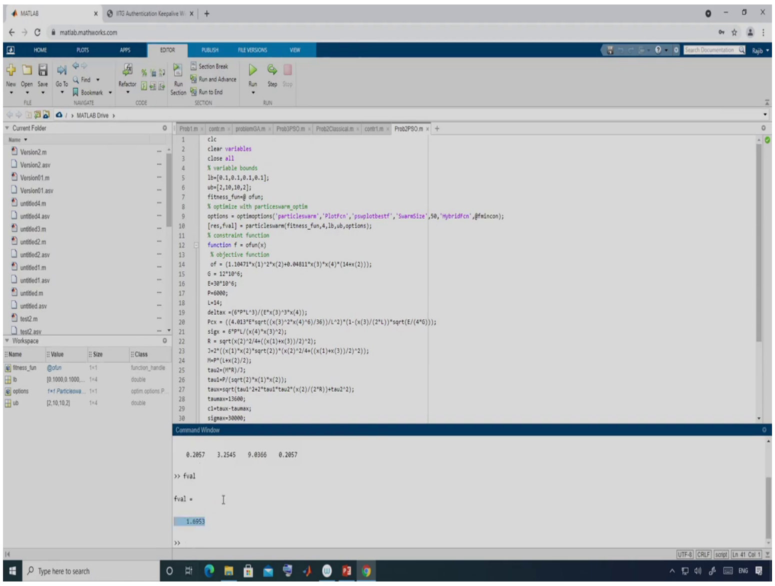
So, it will stop when we will get stall generation of 50 ok. So, we are getting 50 that is the stall generation then after that it is the algorithm has terminated. So, this has been terminated ok. So, we are not getting the solution we can run it again once just no. So, we are not getting solution using ga.

(Refer Slide Time: 43:39)



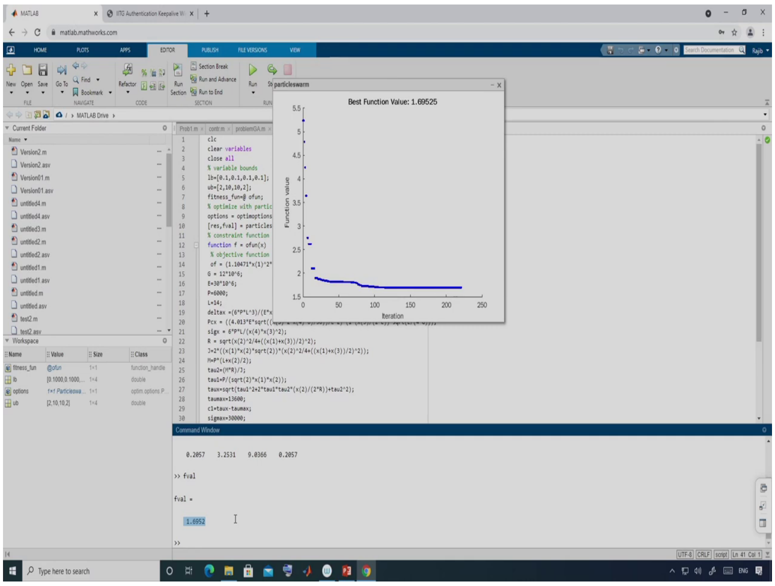
So, let me try with the PSO method ok. So, this is the PSO problem 2 PSO ok. So, let me run it, yeah I am getting the solution you can see. So, here this is the fitness function versus iteration. So, solution is around 1.7 so, or 1.69.

(Refer Slide Time: 44:33)



So, let me checked it this one. So, the solution is these ok and yeah 1.69. So, I am getting the same I am getting the same solution what I got using the classical method.

(Refer Slide Time: 44:55)



So, you can try again I think second iteration also I will get the same solution yeah. So, you can see this is the solution ok and objective function value is 1.69 ok. So, when you are applying this algorithm. So, we have learned lot of algorithms here the both classical methods, as well as non classical methods. So, it is not that the all the algorithms will be equally good for your problem.

So, you have to find out that what algorithm you can you should apply for your problem. So, depending upon what type of problem you have. Suppose as I said that if your problem is a convex problem. And you just apply the gradient base method; so easily you will get the solution of your problem, but if your problem is a non convex problem. So, in that case initially you can apply the meta-heuristic optimization method.

And all the meta-heuristic optimization method may not be suitable for your problem suppose in this case, we have observed that when we applied ga. So, we did not get the solution, but when I did some test run myself. So, I got the solution of this problem, but during this live demonstration. So, I did not get the solution.

Probably, if I am running this particular ga maybe for 10 times or 15 times so, I may get the solution maybe one time or two times I may get the solution, but most of the time I am not getting the solution of this problem so; that means, for this particular problem ga is not a robust algorithm.

So, what we have observed, that when we have applied ga. So, we I did not get the solution so that means, for this particular problem ga is too slow. So, it is taking lot of time. So, lot of iteration and after that also I am not getting the solution of this particular problem.

But, when I have applied the classical method as well as particle swarm optimization method so, I got the solution of this problem so; that means, the PSO method as well as the classical optimization methods are quite your good for this particular problem. So, therefore, the practical application is quite difficult because you do not know what is the optimal solution of your problem?

Suppose you have a problem of 100 variables. So, you will not be able to plot this problem. So, and you do not know what is the exact optimal solution of this problem. So, how many optimal solution you have that also you are not aware. So, in that case this is quite difficult.

So, what you have to do you just apply the algorithm and you can just report or you may report that the known base solution of this problem is these, because you are not aware there may be some global optimal solution that you have not actually you have you did not get that solution. So, there may be a global optimal solution and probably you are not aware of that particular solution. So, you can report that the known base solution of this problem is this.

So, these are all benchmark problems. So, I know the solution. So, this problem can be used for testing your algorithm and basically you can see. So, when I have tested the algorithm like classical method and particle swarm optimization. So, these two algorithms are performing well, but when I have applied genetic algorithm. So, for this particular problem I did not get the solution or I did not get the get a better solution ok.

So, this has to be consider when you are applying optimization for a real world problem. So, this is all about for today's class and as I said this is the last class of this particular course and I discuss different algorithms please go through this algorithm and try to solve your problem applying this algorithm.

So, you can write your own code. So, if you are willing to do that you can develop your own code for implementing particle swarm optimization, genetic algorithm then classical optimization, but if you do not want to do that and this is this will not be easy also. So, you have to write a very nice code to implement this algorithm. But if you want to do that you can do it otherwise the algorithms available in MATLAB or in R.

So, that you can use and for smaller problem ok. So, you can also use Excel Solver if you do not want to write any code ok, or do not want to use either R or MATLAB. So, in that case you can solve your problem using Excel Solver ok. So, thank you very much. So, you can contact me. So, if you have any question, you can write back to me. So, I will try to reply in future also.

Thank you.