Computer Aided Applied Single Objective Optimization
Dr. Prakash Kotecha
Department of Chemical Engineering
Indian Institute of Science, Guwahati

Lecture – 24
Case Study: Production Planning MATLAB Implementation

Welcome back. In this session, what we will be doing is we will be first coding the Production Planning problem which we have discussed previously and then we will plug it with optimization algorithm right. So, we will be plugging it initially with TLBO. So, the first part of this session is going to focus on developing the fitness function file for the production planning problem right.

(Refer Slide Time: 00:53)

Case Study: Production Planning Industry

Sale Price (monetary unit/unit of product)	Product	Process	Capacity (units of product/yr) l_j	m_j	b_j	Production cost (monetary unit/unit of product) C_{ij}	C_{mj}	C_{bj}	Investment cost (monetary unit/unit of product) V_{ij}	V_{mj}	V_{bj}	Raw material used (per unit of product) R1	R2	R3
0.975	T1	P1	70	135	270	50.7	90.1	170.7	55	81.1	131.6	0.948	0	0
		P2	75	150	300	56.8	103.8	196.2	58	85.1	132.4	0.9432	0	0
		P3	77.5	155	310	56.9	103.7	195.7	60.2	86.8	134.1	0.949	0	0
0.975	T2	P4	70	145	290	51.7	97.6	184.8	55.1	83.1	132	0.9546	0	0
		P5	47.5	95	190	38.2	69.8	130.4	43.3	66.8	104.3	0.955	0	0
0.780	T3	P6	40	80	160	38.5	65.2	120.7	66.2	92.8	153.2	1.045	0	0
		P7	40	80	160	31.8	57.1	105.5	40	61.4	95.1	1.05	0	0
0.735	T4	P8	45	90	180	37.8	57.7	94.9	106.6	151.7	231.5	0.5103	0	0
0.1450	T5	P9	40	80	160	38.5	65.6	119.1	82.8	125.4	207	0.6289	0	0
0.1130	T6	P10	90	180	360	92.2	159.2	290.9	233.5	390.7	698.7	0.8648	0	0
		P11	90	180	360	86.7	154.1	287.7	185.8	304.5	537.1	0.9546	0	0
		P12	90	180	360	95.8	175	330.9	119	179.4	289.2	0.8265	0	0
		P13	90	180	360	87.5	157.2	294.9	212.3	362.7	657.7	0.7875	0	0
		P14	90	180	360	105.9	196.6	375.2	109.8	164.3	263.1	0.8101	0	0
		P15	90	180	360	93.1	131.1	239.4	221.7	376.1	672.7	0.8782	0	0
0.830	T7	P16	50	100	200	41.4	68.7	117.2	115.5	180.4	287.4	0.815	0	0
		P17	50	100	200	34.9	62	111.6	63.7	100.2	156.3	0.6994	0	0
0.450	T8	P18	60	120	240	36.6	62.1	120.8	23.1	33.2	50.7	0.3784	0	0

So, just to recollect the problem right so, here we will be having few processes right. So, in the slide, we have shown you only 8 products, but there are 24 such products right. So, as we

can see product 1 can be produced by three process. Process 1, process 2, process 2; 2 can be produced by 4 5, 3 can be produced by 6 7 and so on right. So, we will be having till P 54 right. So, 24 products that can be produced by 54 processes; again not all products can be produced by every process right

So, these are the production costs associated with each process. So, process 1 if we operate at low capacity level, we will have a capacity of 70 right; so, 70 units of product per year, so, you can take it as 70 tons per year. Similarly if we operate it at medium level it is 135 tons per year and if we operate it at h j, it is 270 tons per year right. So, the production cost is also given right.

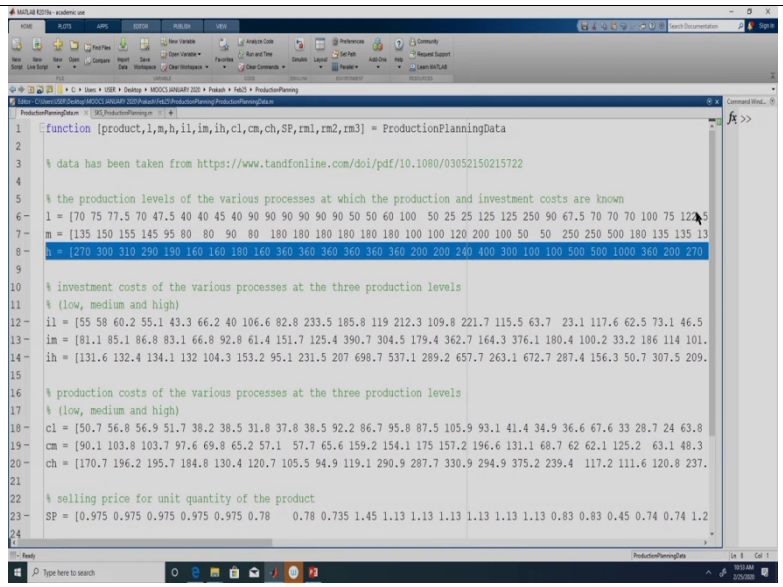
So, if we produce 70, the production cost is 50.7; if you produce 135, the production cost is 90.1. If we produce 270, the production cost is 170.9. So, in between these if we produce. So, if we produce between seventy and 35, then we will have to use the line connecting 70 comma 50.7, 135 comma 90.1 to find out the production cost at any production which is between 70 and 135 right.

So, similarly we will be having this investment cost. So, if we produce 70 units per year the amount of investment cost that will require is 55 monitory units per unit of product. If we produce 135, the investment cost is 81.1. If we produce 270, the investment cost is 131.6 right. So, this is true for all the 54 processes and then we have three types of raw material raw material 1, raw material 2, raw material 3.

So, at least this first 18 process which we have do not consume raw material 1 and raw material 3 right. So, if you remember the previous slides some of the rest of the processes consume R 2 and R 3 right. So, this is the amount of raw material 1 that is required to produce 1 unit of product right. So, if we decide to produce 70 units from process 1, then it is 0.948 into 70.

So, similarly we are also given the selling price right. So, if we sell 1 unit of product T 1, the amount that we get is 0.975. So, if we produce 100, units over here then it is 100 into 0.975. So, our first job is to get this data.

(Refer Slide Time: 03:28)



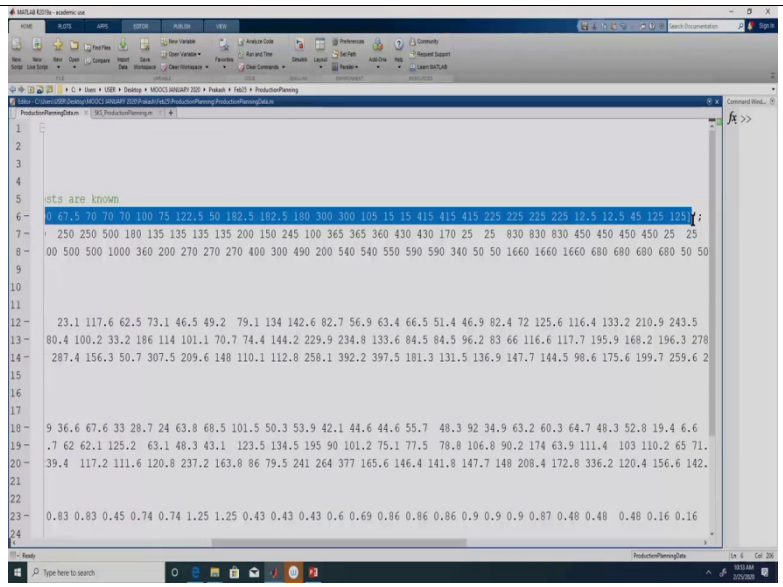
So, what we will do is we will separate the data. So, we will get all the data over here. So, what we have is a function file right. So, the name of the function file is ProductionPlanningData right. So, you can give any other name that you would want right. So, we have given ProductionPlanningData.

So, there is no input to the this function right. So, here if you see on the right hand side, we are not passing any input. This function will only give us the output. So, what we will be doing is we will be calling this function whenever we want to evaluate the fitness function right. So,

to get the data we do not need to supply any input right. So, these are the 3 production levels right.

So, whatever you see in this slide 71, 35, 270 right so, this is 70, the low level the medium level is 135, the high level is 270 right. For the second process, it is 75, 150, 300; so, 74, 150, 300 right.

(Refer Slide Time: 04:21)

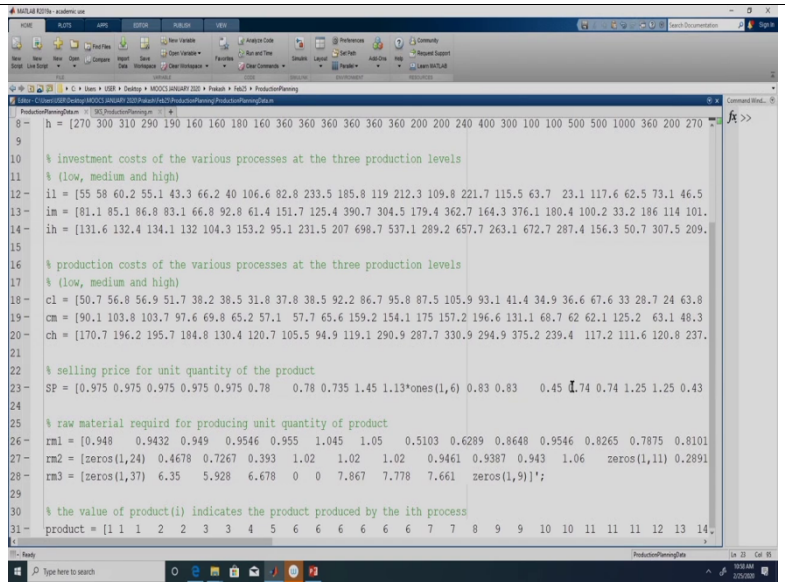


Like this we have 54 values in each of this variable right. So, l is a column vector right. So, it will have 54 rows and 1 column right. So, we have typed it as row vector. But you have use the transpose over here. Similarly m will contain 54 values and h will also contain 54 values. So, we have taken this data l m h into this file right. Similarly we need to take the investment cost right. So, il denotes the investment cost for the low level.

Here if we see the investment cost is over here. So, it is 55 81 and 131.6. So, 55, 81, 136.6 and the second one is 58 81.6 132.4; 58 81.6 132. Just like for each production capacity, we will have a investment cost right. So, if we decide to produce 70, the investment cost is going to be 55. If you decide to produce 135, the investment cost would be 81.1. If we decide to produce 270, the investment cost would be 131.6.

If we decide to produce anything between 70 and 135 the investment cost has to be calculated using this 55 and 81.1. So, that we will be doing later, right now we are only entering the data whatever we have.

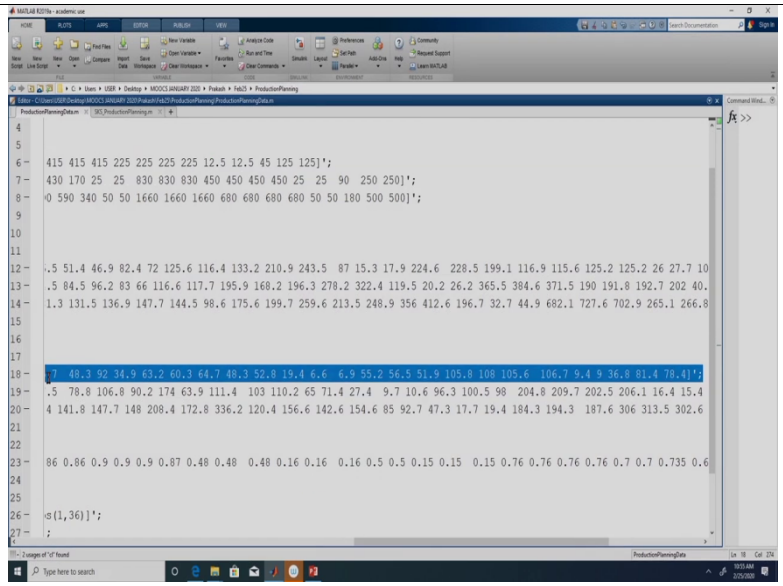
(Refer Slide Time: 05:37)



So, similarly cl, cm, ch indicate the production cost corresponding to l m and h. So, here if we see the production cost is 57.7, 90.1 and 170.7 for process one. So, cl is 57.7, 90.1, 170.7. Similarly for the second process, it should be 56.8, 103.8 and 196.2 right; so, 56.8 1, 103.8

and 196.2 right. So, this is similar to l, m, h the c l cm ch will be a column vector right. So, at the end we have put this transpose over here. So, it will be a column vector.

(Refer Slide Time: 06:13)



So, what we have done so far. We have entered the production capacity, the investment cost and the production cost right. So, selling price if we see over here so, this we can write it as 0.975, 3 times right. So, because T 1 is produced from process 1, process 2, process 3 and no matter from which process it is produced, the production cost is 0.975. So, that is why we are repeating this 0.975 3 times right; 0.975, 0.975, 0.975 and product 2 is produced from product 4 and product 5 right, but the cost is 0.975.

So, that is why we again have 0.975 twice. So, product 3 is produced from process 6 and 7 and the selling price of product 3 is 0.78 right. So, we will have 0.78 twice because it can be produced by two processes right. So, product T 4 can be produced only from process 8 and

the selling price is 0.735. So, here we have 0.735 just once. So, 1.13 is the selling price of product 6 right and it can be produced by 6 processes right.

So, here we have 1, 2, 3, 4, 5 and 6. So, we have this 6 times right. So, we could do this or we could do 1.13 into 1s of 1 row and 6 columns right. So, this will also give us 1.1 3, 6 times and then we have a product 7 which is produced from 2 processes right. Cost is 0.83. So, 0.83, 0.83 and we have product 8 which is produced only from process 18 and the selling price is 0.45.

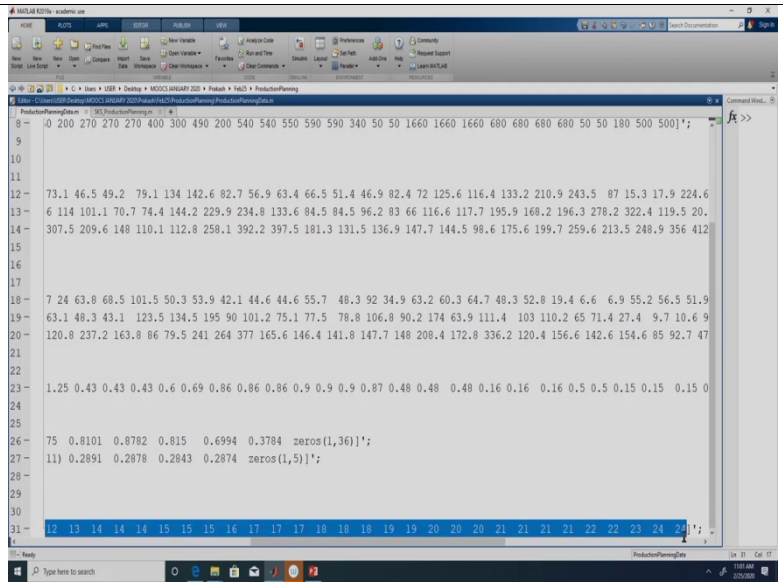
So, we have this 0.44 over here these are the values for the remaining processes. So, you can go back and look at the slides right and then we have these 3 raw materials; raw material 1, raw material 2, raw material 3 right. So, for example, 0.948 0.9432, 0.949 is the amount of raw material required to produce 1 unit of product right. So, that is why we have 0.948, 0.9432 and 0.949 right.

So, similarly we have entered the values for the rest of the processes right. If you look at the entire table, you will see that the first 24 processes do not require raw material two. So, here at least for the 18 you can see. So, for raw material 3, the first 37 processes do not require raw material 3. So, we have put 0 and then the values given are 6.35, 5.92, 6.678 and this last 9 processes also do not require raw material 3. So, we have put that also as 0s.

So, right now what we have entered is the production capacities at which we know the investment cost and the production cost, the selling price of each product that is coming from each process right; the amount of raw material that is required for each of the 54 process. So, that has been entered right. So, the next thing is to code this information that which of the processes are producing which product right.

So, what we have done is we have defined the variable product right. So, for example, this is the first value of product, this is the second value of product, this is the third value of product right.

(Refer Slide Time: 09:29)

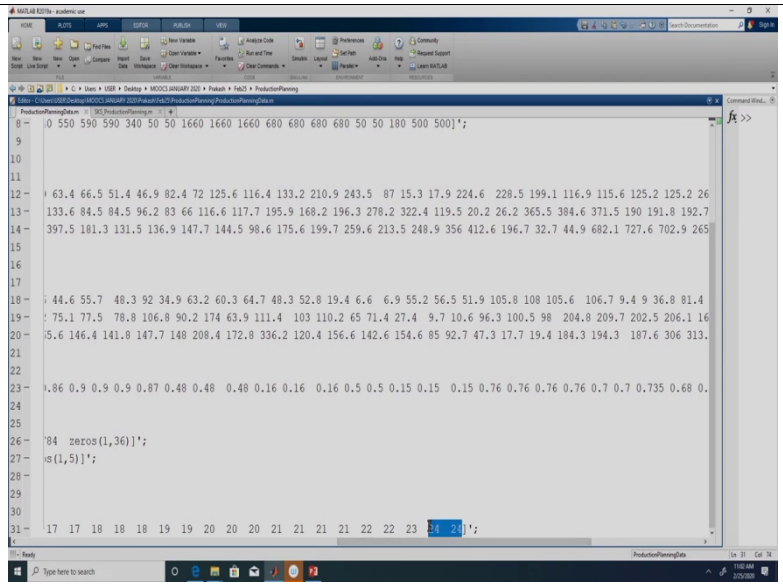


So, there are 54 such values right. So, it indicates what is the product which is produced by that particular process right. So, this is at the first location. So, this indicates about process 1. So, process 1 produces product 1, process 2 produce product 1, process 3 also produces product 1 right; process 4, this is process 4 because this is the fourth value in this vector right. First value, second value, third value and the fourth value so, the fourth and fifth value are 2. So, that means, the fourth process and the fifth process produce product 2 right; this is 6 7 right.

So, the process 6 and process 7 produce product 3. This is 8 right. So, process 8 produces product 4, process 9 produces product 5 right. So, this is 10, 10, 11, 12, 13, 14 and 15. So, till 15th process products 6 is produced that is why we have repeated product 6 6 times right. So, similarly this would be the 16th value and this would be the 17th value right. So, the 16th and

17th value are 7. So, that is why. So, process 16 and process 17 produce product 7. So, that is why 7 is written. So, similarly we have coded for all the processes right.

(Refer Slide Time: 10:44)

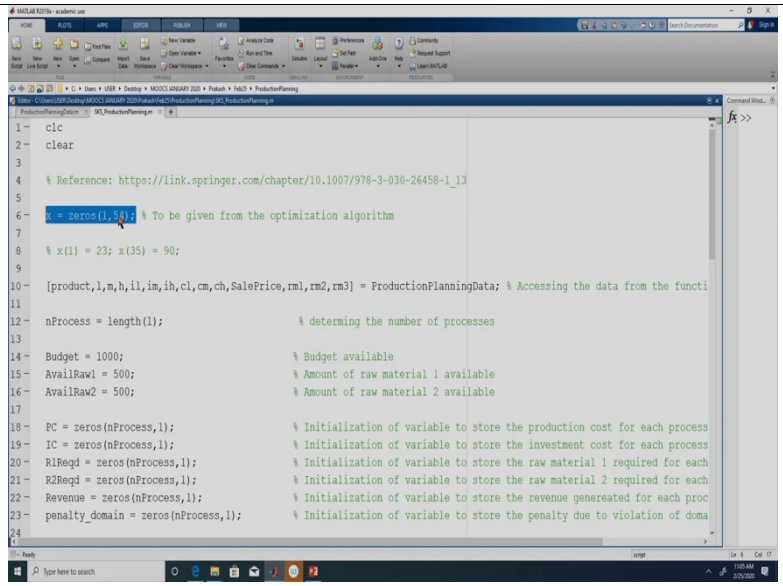


So, we have product 24 which is being produced by process 54 and 53. So, now, that we have this information, we will be passing all this information whenever this function is called right. So, if we call this production planning data, we can get this as output right.

So, product will tell us which process is producing which product l m h are the production capacities, at which the investment cost and the production cost are known; il im ih are the investment cost at l m and h; c l, c m, c h are the production cost at l m and h selling price is the sales price by selling the product from a particular process.

Since we have 54 processes, this we have 54 values; rm 1, rm 2, rm 3 will indicate the raw material 1, 2 and 3 respectively required for each process right. So, this completes the data definition. So, whatever data we have for this problem, we have taken it over here right. So, what we will do is we will not write any piece of code over here. We will just restrict it to the data right. What we will do is we will write another file right which will access this file write and given a x, it will give us fitness function value right.

(Refer Slide Time: 12:02)

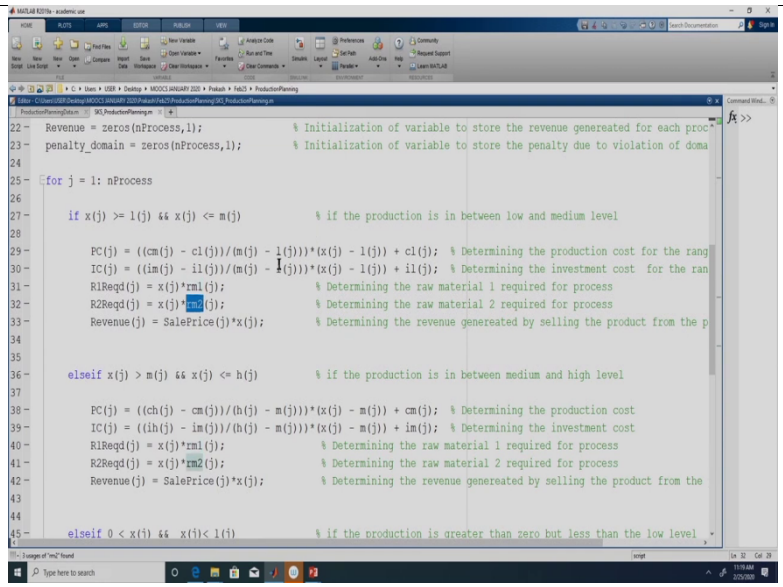


So, here initially we are developing a script file. So, remember that first we need to develop the fitness function file right only after we have developed the fitness function file can we solve the optimization problem right. So, we are independently developing the fitness function file without worrying about the optimization. So, clc clear you know right.

So, as we discussed previously, the algorithm will tell us how much quantity of product has to be produced from each process and there are 54 process. When we execute this problem right the algorithm is supposed to give us the decision variables right. So, right now we are not clubbing it into an with an optimization algorithm. So, we will define the x ourselves right.

So, initially we define it as 0s of 1 comma 54; 1 rows and 54 column. So, whatever the decision variables we are going to get from the algorithm is going to be in this form. It will have 1 row and 54 columns right.

(Refer Slide Time: 12:54)



So, then what we are doing is we are accessing that function which we have written previously right; so, production planning data. So, this we are accessing over here right. So, we will get all necessary information over here and we do not need to copy paste this data over here right. So, we can just access this file. So, that way we can keep the calculation part separate from

the data part right. So, once we have this, what we will first do is calculate the number of processes involved right. So, that can be determined from the length of any of this vectors right.

So, all these are vectors which will have uniform number of rows at least in this case where in we know the data for 54 processes right. So, we can measure the length of any of this variable and determine the number of process right. So, let us assume that the amount of budget that is available to us is 1000, the amount of raw material 1 which is available to us as 500 and the amount of raw material 2 that is available to us is 500.

So, whatever production plan we come up with right it should not exceed these values right. For any production plan if the investment cost is going to be greater than 1000, then it is not going to be a feasible plan right. So, we will incur penalty. Similarly for raw material 1 and raw material 2, it has to be less than 500 right.

So, here line 18 to 2, 3 we are defining 6 variables right; 6 variables. This variable PC will contain the production cost for each process. Remember this PC is different from this cl, cl also contains production cost, but this is the production cost of the production indicated by l right. But when we are solving an optimization problem, it is not necessary that we will be either producing l m or h right.

So, we maybe even be producing between l and m or between m and h or we may even choose not to produce it right. So, for whatever amount we are deciding to produce we need to calculate the production cost. So, that is why we are defining this variable pc right. So, for each process, we will have a production cost. Initially we are defining it with 0s; 0s of number of rowses n process which is 54 in this case and 1 column right.

So, similarly for each process, we are finding out what would be the investment cost. So, il, im, ih indicates the investment cost at l, m and h right, but we might have a production which is not necessarily lmh it may be 0 or between l and h right. So, for that we need to calculate

the investment cost accordingly. So, that will be the investment cost over here; raw material 1 required and raw material 2 required.

So, these two variables will store the amount of raw material one that is required for each process, amount of raw material 2 that is required for each process. So, we will be using this variable revenue to store the amount of revenue, we get from each process right and then will sum it up to find out the total revenue, but what is the revenue obtained from each process will be calculated and stored in this vector revenue right. So, again it has 54 values for this data.

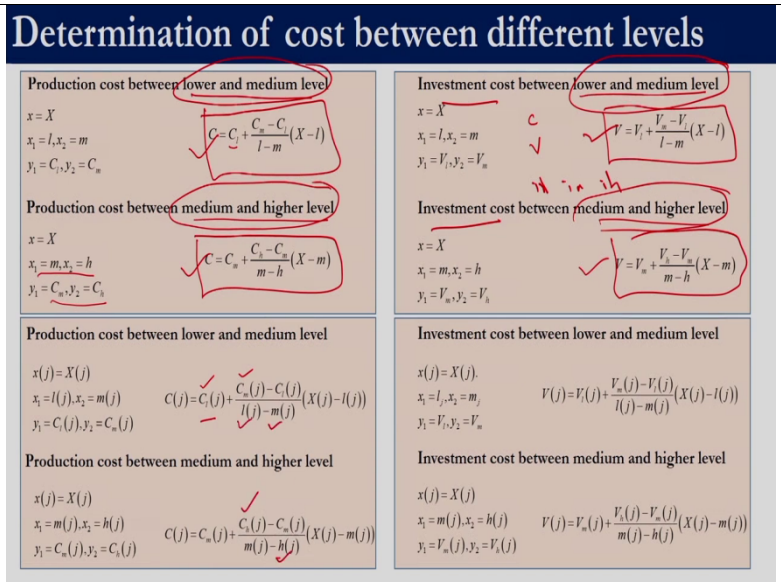
So, then if you remember, we also have a domain whole constraint right. So, any value has to be greater than or equal to l or it has to be less than or equal to h or it can be 0 right. So, if it is greater than 0, but less than l right then we will have to assign a penalty. So, as discussed earlier what we will be doing is if something is greater than 0, but less than l; we will assign a penalty of 10 power 5 right.

So, but we are not going to assign it for every variable we are only going to assign it for variable which violates. So, what we are doing is, we are defining a variable penalty underscore domain right. Initially it is having a values of 0. So, whichever variable is going to violate that constraint the domain whole constraint, we will be assigning a value of 10 power 5 for that particular process. So, these are just initializations; this fixed variables are just initializations right.

So, now that we have defined the necessary variables. We can calculate the required values right. So, what we are doing is we are running this loop right for j is equal to 1 to n process because this vector x is going to contain 54 values right.

So, for each of the value, we need to see whether it is within the permissible domain or not if it is not within the permissible domain, we are going to assign a penalty. If it is going to be in the permissible domain, we are going to calculate the cost accordingly. This for loop ensures that we calculate everything for each of the process.

(Refer Slide Time: 17:25)



So, this equations we have derived previously right. So, this is nothing, but if the production cost is between lower and medium level medium level, we will have to use this equation. If the production cost between medium and high level, we need to use this equation. If the investment cost is between lower and medium level, we need to use this equation right. This is for investment cost these two equations are for investment cost.

This is valid between low and medium level, this is valid between medium and high level. Similarly production cost is this is valid between lower medium level and this one is valid between medium and high level right. This is for one right. So, this is for one particular process. So, for the jth process right, it will be the same thing instead of this cl we are going to have c l j. So, for every process this values are different cl cm ch values are going to be different; l m h are going to be different for each process right.

So, need to quote these four equations for as we have discussed the production cost between l and m is given by this expression right. So, how to get this expression that we have seen a little while earlier right. This is the investment cost for a process which produces between l and m right greater than or equal to l and less than or equal to m. This 2 lines will help us to calculate the production and investment cost right that is because the production and investment cost in this region right varies linearly. So, that is why we are using these two equations right.

So, the raw material 1 that is required is the amount that is produced right multiplied by the amount that is required to produce 1 unit right. So, this is given per unit right. So, if we produce 1 unit of product 1, the amount that we require is 0.94 unit for process 1. If we decide to produce 1 unit of product right, the amount that is required is 0.9432 if we decide to produce.

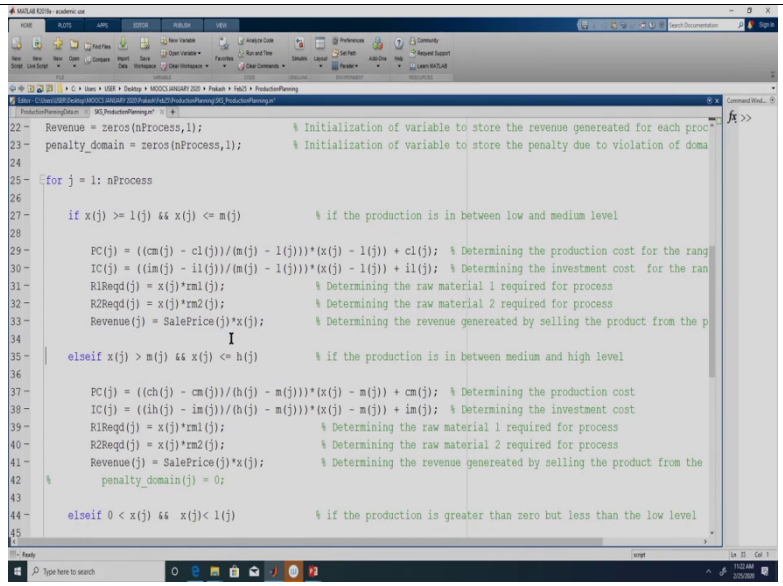
Let us say 80 units over here for product 2, then the amount total amount of raw material 1 that is required 0.94 2 into 80 right. So, that is what we are doing over here. So, the amount of production multiplied by how much of raw material 1 is required for producing 1 unit right.

So, similarly we calculate for rm 2 right. So, these two equations are similar except for here it is rm 1, the amount of raw material 1 that is required and here the amount of raw material 2 that is required. So, if this value is 0 so, as may number for many processes raw material 2 is not required. So, this value will be 0, this line 32 would give us 0 right. Similarly here we are calculating the revenue. So, we know the sales price. So, for each product, we have been given the selling price. What we are doing is the amount that is produced which is x j, we are multiplying it with selling price for 1 unit.

So, this will give us the revenue for process j right. This equations are valid for production greater than or equal to l and less than or equal to m right. If it is greater than m right and less than equal to h, we will calculate accordingly right. So, these 3 equation are the same, it does not matter what the range of x is as long as it is between l and h right.

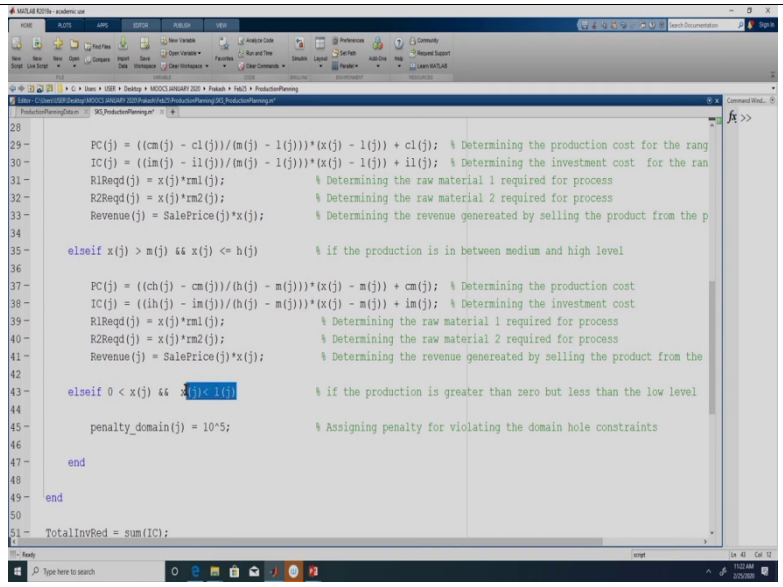
So, these two equations change. Here we had the cm minus cl in the numerator. Here we have ch minus cm in the denominator here we had m minus l here we have hj minus m and j. So, again this is the same set of equation that we have derived previously. So, right now what we have done is given any production which is greater than or equal to l and less than or equal to h, we can calculate the production cost investment cost raw material 1, raw material 2 that is required the revenue right.

(Refer Slide Time: 21:10)



And if the production falls between l and h, then the domain whole is violated right. So, since this penalty domain is already 0, we are not writing here penalty domain of j equal to 0. If it is in the range right between l and m or if it is in the range m and h the penalty is 0, but we do not need to write this line because the penalty is anyway initially assigned as 0. So, only in those cases were it is violating will we go and assign a penalty.

(Refer Slide Time: 21:40)



So, the third condition is this one right. So, this is between l and m, this is between m and h and this is x is greater than 0 right, but it is less than l right. So, in that what we need to do is we cannot calculate the production cost. So, for example, consider this process 9. So, the production cost from 40 to 80, we know how to calculate and if it is 0 right, then there is no problem the production cost is 0, the investment cost is 0, the raw material 1 that is required is 0, raw material 2 that is required is 0.

And no matter what the selling price, the revenue earned is 0 because the production itself is 0 quantity right. But if the production is let us say, 10 as far as the algorithm concerned, the lower bound is 0, the upper bound is 160. So, it can give any value between 0 and 160; let us assume it gives a value 10.

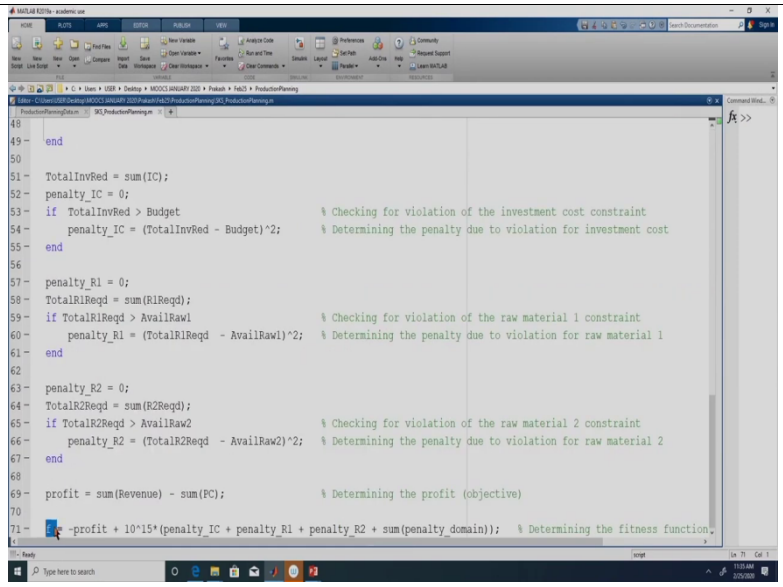
So, if it gives a value of 10, it violates the domain whole constraints, it is greater than 0, but less than 40 right. So, we cannot calculate the production cost investment cost, the revenue or anything. Since this production cost and investment cost is anyway assigned as 0, we do not do anything to the production cost and investment cost.

Similarly, we do not calculate raw material 1, raw material 2 because it is 0. So, similarly revenue is also 0. If it falls in this range right so, what we will do is so, but in this case we need to assign the penalty. What we are doing over here is the variable penalty domain, the j'th value in that right. So, this is going to be a vector penalty underscore domain, it is going to have 54 values for this example right so, the j'th process. So, whichever process is violating, we are assigning a value of 10 power 5 to that right.

So, this section so, line 25 to 47 will help us to calculate the production cost, the investment cost, the raw material 1, the raw material 2, the revenue and it also assigns penalty wherever the domain whole constraints is violated right. So, now, that we have calculated for all the 54 process, we need to check for the additional 3 constraints. Remember we had four types of constraint; one is domain whole constraint which is taken care over here right. So, assign penalty whenever it is violated, but there are 3 additional contains right.

So, we need to see whether those constraints are satisfied or not if they are satisfied, we do not assign penalty. But if they are not satisfied, we will have to assign a penalty right.

(Refer Slide Time: 24:07)



So, in line 51 what we are doing is we are calculating the total investment cost that is required. So, this variable total invRed determines the total production cost. So, we know the production cost of each process right that is in the vector IC. So, here what we are doing is we are merely summing that vector so, that will tell us what is the total investment cost that is required right.

In line 52, we are initializing the variable penalty underscore IC. So, this variable will contain the penalty assigned for violation in the budget constraints right. So, initially we are assigning a value of 0 to it right. So, then we are making a check over here. So, what is the total investment that is required is given by this and what is the budget that is available is given over here right. So, we have defined budget as 1000 in this case. So, if this condition is satisfied, we

are actually violating a constraints that the total investment which is required is greater than budget right.

So, we are violating that constraints. So, if you are violating the constraints, we need to assign penalty right. So, here we are calculating penalty underscore IC incurred because of the violation in the budget constraint right. So, penalty underscore IC is whatever investment is required minus the budget right. So, let us say the investment required is 1200 right.

So, this will be 1200 minus budget. So, budget in this case is 1000. So, it will be 1200 minus 1000 the whole square. So, that is the penalty which will be assigned for violating the investment cost constraint. So, on the contrary if the value of total inverse Red is 800 let us say and the budget is 1000 right. So, what we are saying is amount of budget that is available is 1000 how much we are utilizing is only 800 right. So, it does not satisfy this condition.

So, if it does not satisfy this condition means the solution is not violating. If it is not violating, then no penalty is to be assigned. In that case penalty underscore IC will remain as 0. Similarly right in this section, we calculate the penalty which is incurred if the solution violates the constraints related to raw material 1 right. So, the amount of raw material 1 which is available is AvailRaw1 what is required is summation of the individual required.

So, remember we have calculated for each process over here for each of the j process as to how much of raw material 1 is required right. So, if you sum all of those values that will tell us how much of total raw material 1 is required and then we are checking for the constraint over here right. So, if it is less or equal, then we do not need to assign a penalty; if it is greater we need to assign a penalty over here, this line 57 to 61 is for assigning the penalty for violating the constraint of the raw material 1.

So, if this is similar to line 57 to 61. Here we are checking for the violation in raw material 2. So, now, what we have done is we have calculated the production cost for each process, we have calculated the revenue earned from each process and we have also checked for the 4 types of constraint right. So, the constraint on raw material 1, the constraint on raw material 2,

the constraint on investment cost as well as the domain whole constraint right. So, now, the objective function was revenue minus production cost right.

So, revenue is a vector which contains the revenue earned from each of the process right. So, if we sum this, it will tell us the total revenue which is earned right. So, PC contains the production cost for each process. If we sum it up, it tells us the total production cost. So, this is the total revenue which we earned minus the total production cost right. So, that is our profit right. Since our problem is a maximization problem right and all the algorithms which we wrote are minimization algorithm. So, what we are doing is we are converting our maximization problem into minimization problem by writing minus profit over here right.

So, as we discussed earlier a maximization problem can be converted to minimization problem by multiplying the objective function with the negative sign right. So, this minus profit is because of that right and then we need to add the total penalty; penalty underscore IC is the violation due to the investment cost, penalty underscore R 1 is the penalty incurred for violating the raw material 1 constraints, penalty underscore R 2 is the amount of penalty incurred for violating the constraint on raw material 2 right and penalty underscore domain is a vector right. So, for all those processes where in we have violated, we need to sum the penalty.

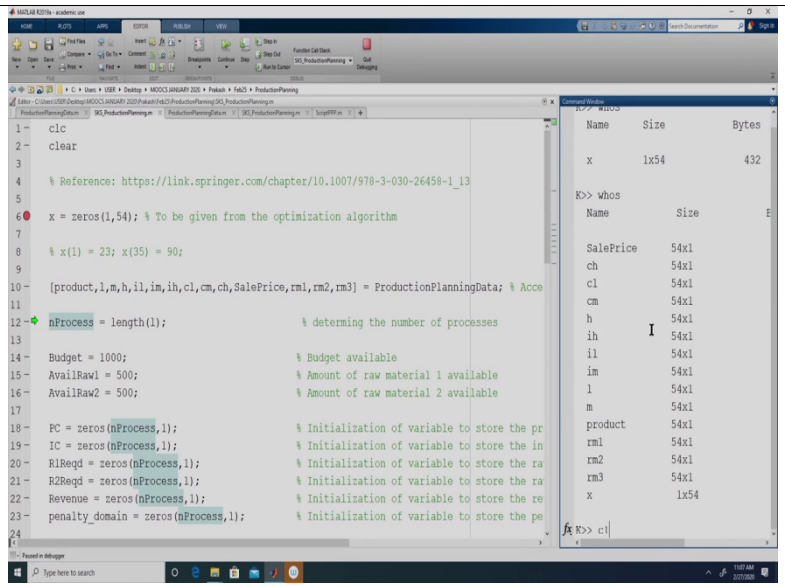
So, the total penalty is sum of penalty underscore domain. So, this will be a scalar right penalty underscore R will be scalar, penalty underscore IC will be a scalar right where as this penalty underscore domain is a vector right. So, that is why we are summing it up right.

So, this part right is the total penalty right and we are multiplying it by a very large value right the penalty factor so, as to get the fitness function value right. So, this completes the determination of fitness function right. So, if you are given a solution so, this file can help us to find out the profit and the violation in all the constraint if any right.

So, if there are no violation; if a solution satisfies all the constraint all of this would be 0 right and the fitness e function would be nothing, but the objective function right. But if some of the variables violate the constraints right, then an appropriate penalty is calculated and added to

the objective function. Now that we have coded the objective function file or the fitness function file in this case because we have constraints right. Let us actually run it in debug mode right so that we can make sure that it is working fine before plugging it with any optimization algorithm right.

(Refer Slide Time: 30:02)

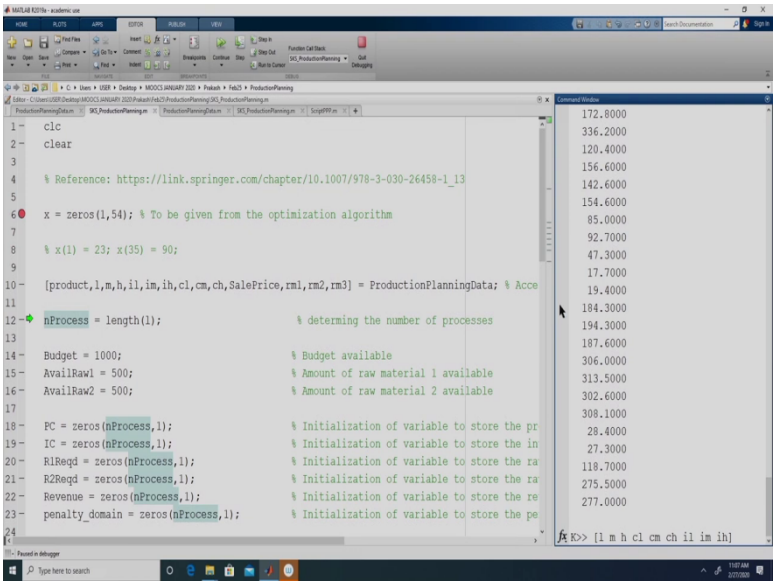


So, let me put up breakpoint over here. Let me execute this all right. So, right now remember we are not running the optimization problem, we are merely checking whether this file is working fine or not. So, this x is supposed to come from the algorithm right, but since we are not plugging it with any optimization algorithm as of now, we are initializing it with 0s right. So, for 0s if you think about it, then all the decision variables are 0. So, what we are saying is there is no production from any of the process. So, this solution will not violate the domain constraint right because is permissible.

Similarly the consumption of raw materials will be 0 for this solution. Similarly the investment cost for the solution will also be 0 and the production cost will also be 0 and the revenue will also turn out to be 0. So, basically what the fitness function what we are expecting is 0 right. So, let us see if that is we are getting that properly or not; if you get that properly then at least for that solution, we are not having problem, then we will check for couple of other x values and then we will plug it with an optimization algorithm.

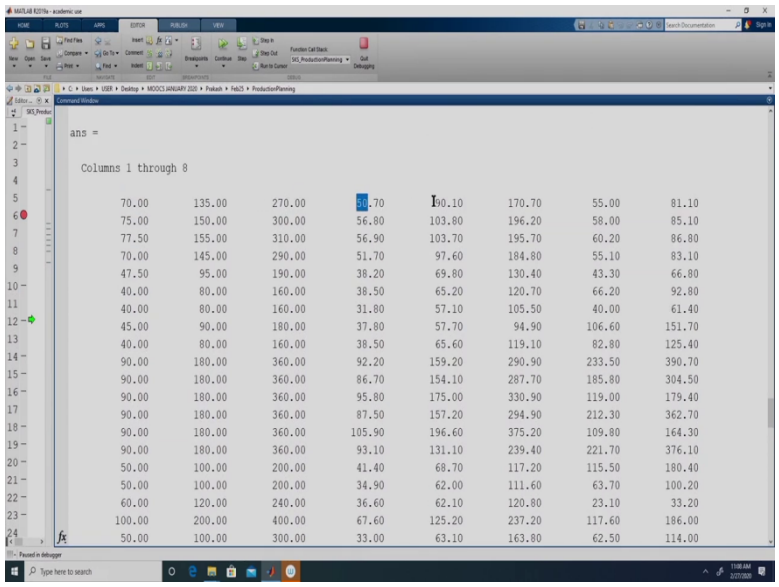
Here we are just calling this function production planning data. So, it will get us all the datas. So, here if we see whos, there is just x right. So, now if we look at this whos, then we have got all the data right.

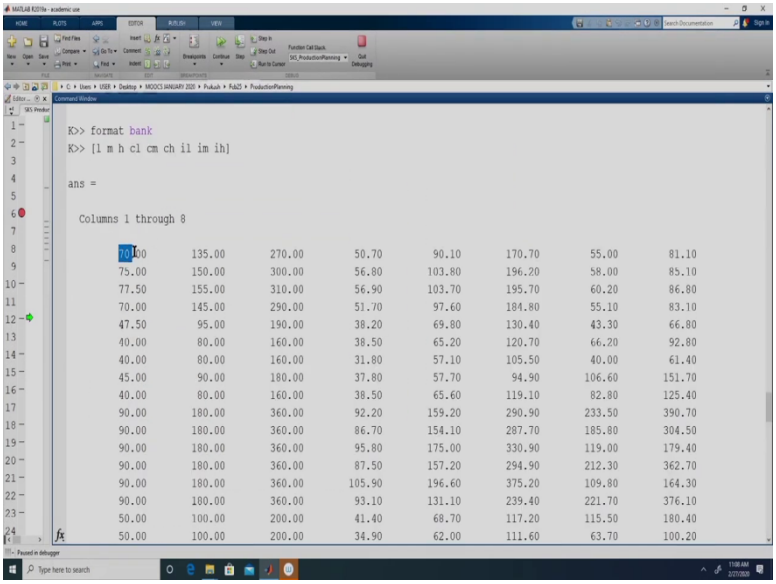
(Refer Slide Time: 31:24)



So, cl; so, this is cl, cm, ch right. So, if you want, we can just do c l, c m, c h l m h right i l, i m i h.

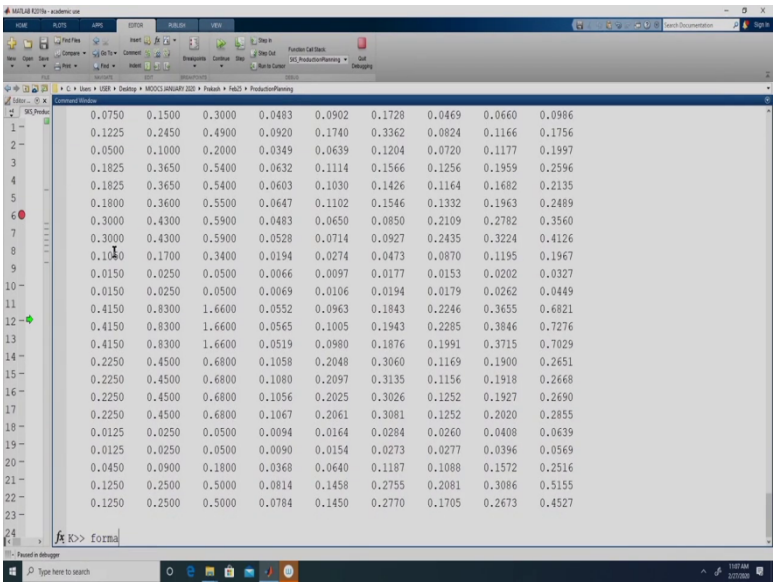
(Refer Slide Time: 31:42)





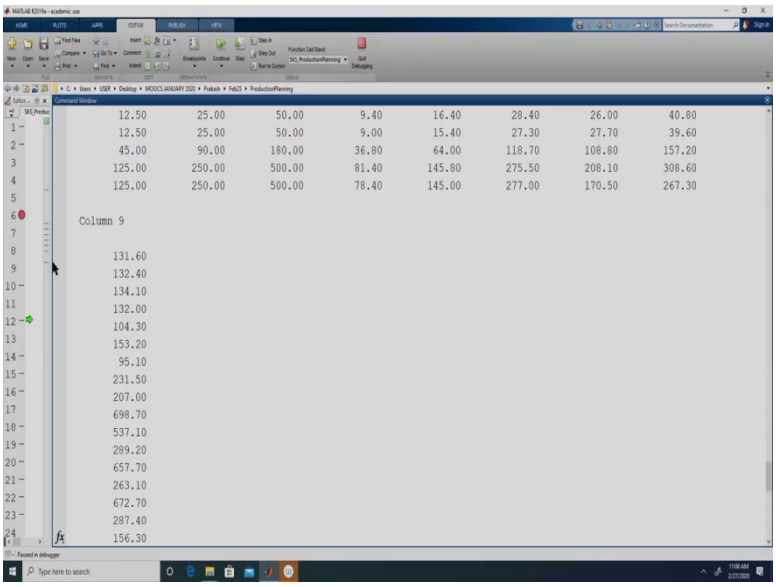
If you look at this data so, this is the table which we have been seeing right.

(Refer Slide Time: 31:48)



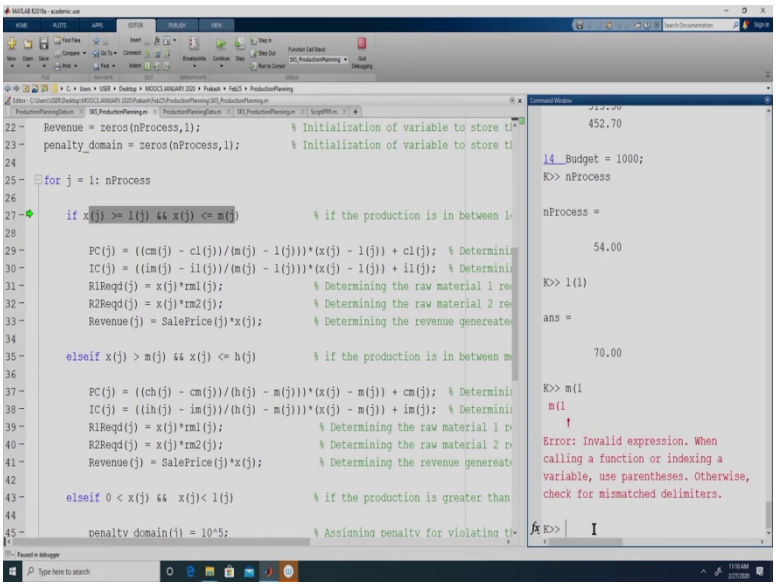
So, let me just change the format right. So, this is the l values right for the 54 processes, this is m value, h value, this is cl, cm, ch, cl, cm, ch right. This is il, im, and ih right.

(Refer Slide Time: 32:05)



Similarly, you can look at the other values the raw material 1, raw material 2; it is number of processes would be 54.

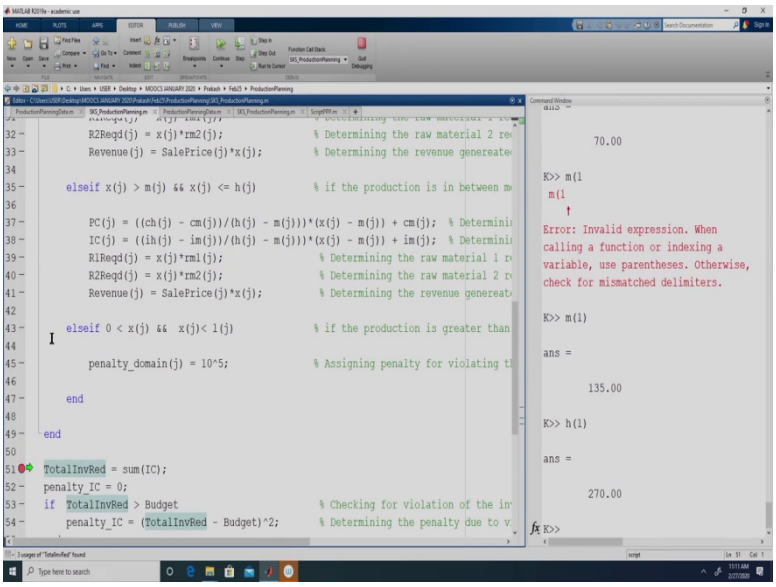
(Refer Slide Time: 32:11)



So, in this case number of processes is 54 right. The value of budget that we are taking is 1000. So, the production plan that we design should have an investment cost which is either thousand or less than 1000. It should not be greater than 1000. Similarly the raw material quantity which is available is 500, 500 right. So, this is just assignment. Similarly from line 18 to line 23, we are just assigning variables right.

So, PC would be used to store the production cost, IC would be used to store the investment cost of each process, R 1 required will be used to store the raw material 1 required, R 2 required is for storing the amount of raw material 2 which is required for each process. Revenue is used to store the revenue earned by selling the products from each of the process and this penalty domain will be assigned the value of 10 power 5 for those process which violet the domain constraint right.

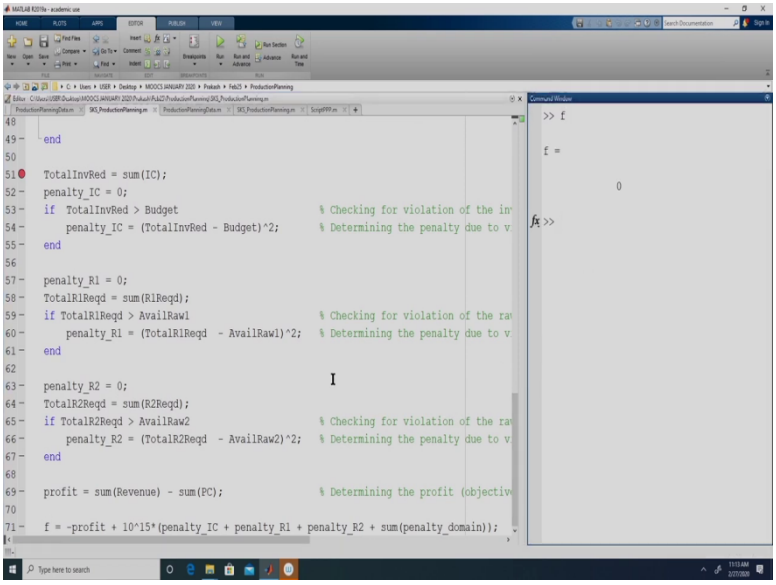
(Refer Slide Time: 33:05)



Till line 23 we are just initializing the variables right. So, now, we are running this loop for each value in x right. Since all the values are 0, it will not go into this loop right. So, l of 1 is l of 1 is 70; m of 1 is 135 and h of 1 is 270 right. So, what we currently have is 0. So, 0 will not fall in this range it will not fall, it in this range and it will not even fall in this range right. So, what we are doing is if it is 0, we are not calculating the production cost investment cost or revenue or the raw materials. So, this is for the first process. So, for the second process again it will not go in any of this condition right so, because it is 0. So, even for third process, it will not go into any of this conditions right.

So, let me just put a breakpoint over here and continue. So, what it has done now it has calculated the production cost, the investment cost, the raw materials required and the revenue for all the 54 processes; let me just clear this right.

(Refer Slide Time: 34:15)



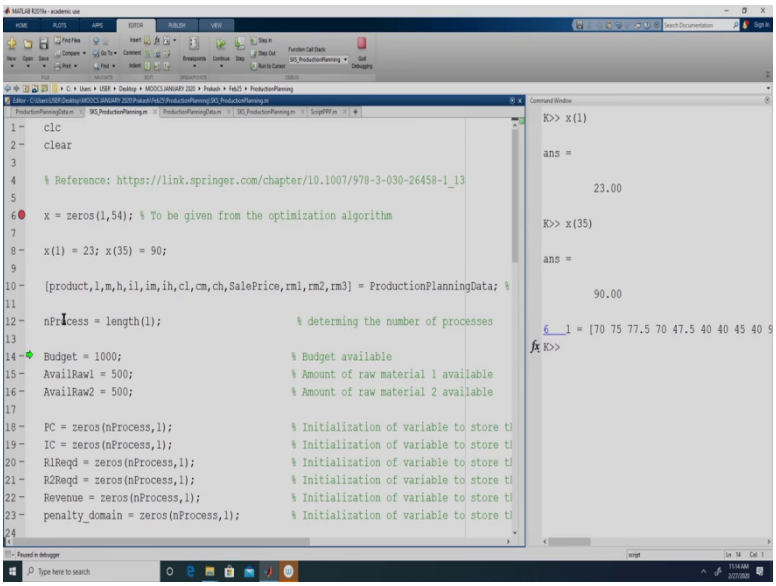
So, now we are calculating the total investment that is required right. So, IC is currently it will have a value of 0s because we initialized with 0s over here right and we did not assign any value to it right. So, this is going to be 0 right. So, penalty we are initializing it to be 0.

So, there are budget which is available to assess 1000 and what we require is only 0. So, this condition 0 is not greater than 1000. So, it will not go into this if condition right. Similarly penalty underscore R 1, we are assigning it to be 0 initially right. The raw material that we require is again 0 right and the available amount of raw material is 500. So, it again does not go into this condition. So, penalty underscore R 2 is the same thing right since we have all the variables is 0; it is not getting into this condition. So, profit would be again 0 right.

So, revenue is 0 for all the processes. So, some of that will be also 0. Similarly production cost we did not calculate the production cost at anywhere right. We initialized it to 0, we did not

calculate because it did not fall in any of this ranges right. So, this profit will be 0 right and the fitness function value would also be 0. So, f is 0 right. So, at least with heroes, it is working fine right.

(Refer Slide Time: 35:46)

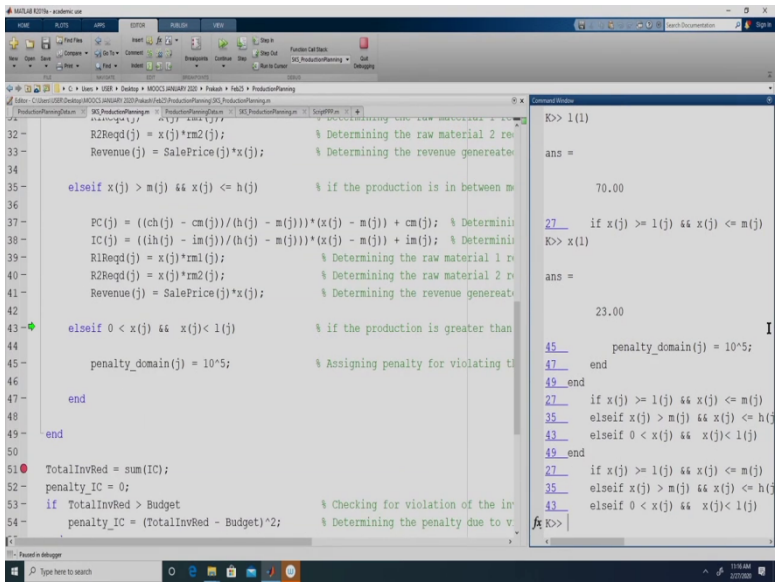


So, now what we will do it we will again do it right, but this time what we are doing is we are not giving an vector which is full of 0s. We are giving x of 1 is equal to 23 and x of 34 is equal to 90.

So, the first element is 23 and then 2 to 34, we have 0s; 35 we have 90 and then after 90, we again have set of 0s right. So, here in this case it will at least go into one of those if conditions. Let us see what happens in this case right. So, initially x is 0s right. Now we are over writing x of 1 and x of 35 right. So, if we say x of 1 is 23 x of 34 is 90 right. So, now, we are calling the data right. So, this we can just step out right.

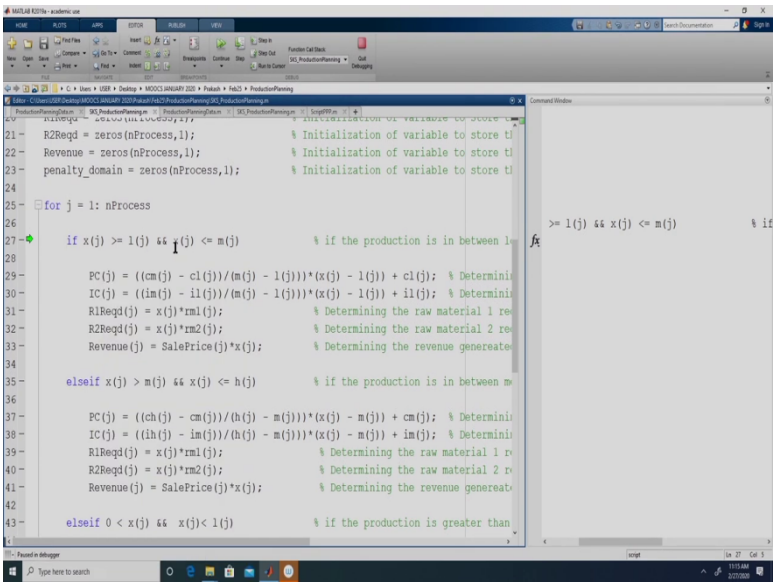
So, all the data has been brought into the workspace right. So, the number of process is 54. These 3 lines we discussed right and again this is just initialization right.

(Refer Slide Time: 36:42)



So, before going to this loop just let us look at what is l of 1. So, l of 1 is 70 right and what we have is 23 right. So, if you think about it x of 1 is not 0 right. So, it is supposed to be greater than 70, but it is not greater than 70 right.

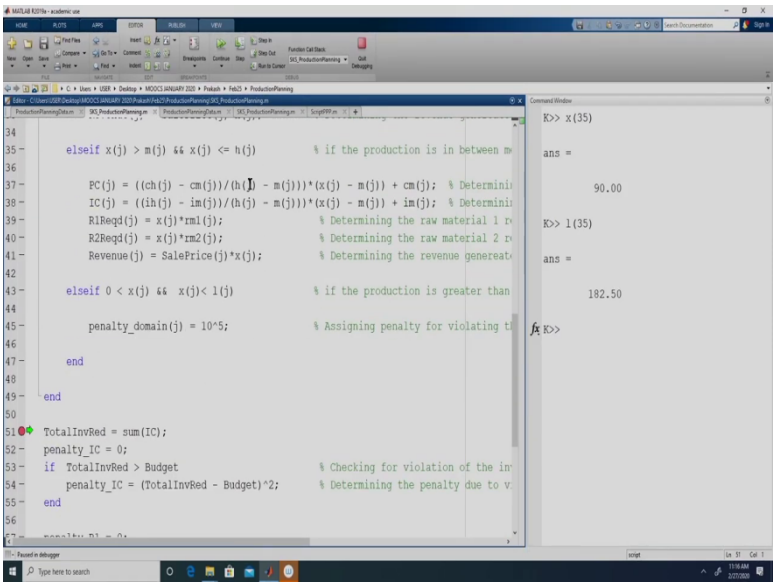
(Refer Slide Time: 36:56)



So, this condition is not satisfied right. So, l of j the first value is 70. So, x of 1 is not greater than 70. So, it will not get into this part right again, it will not get into this part. So, we get into this right x of 1 is violating the domain constraint right. So, that is why it goes into this. So, this condition will be satisfied. So, we are supposed to assign a penalty for this variable right that is what we are doing over here. So, penalty domain of the first variable is 10 power 5.

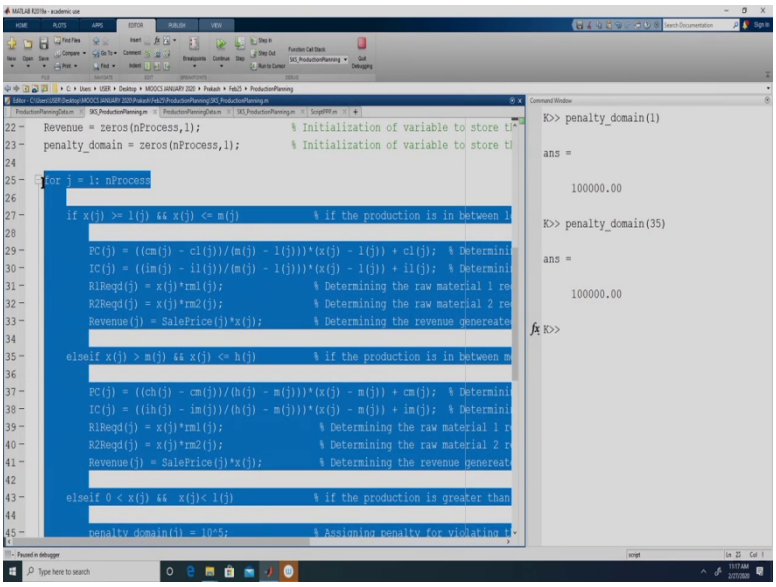
So, similarly we need to calculate for all the variables right. So, for 2 again it will not go into any of these conditions right, for 3 it will not go into any of this condition right.

(Refer Slide Time: 37:37)



So, let say x of 35 is 90, let us see what is l of 35. So, that is 182.5 right. So, here also it incurs a domain constraint penalty right. So, what we will do is right we will just continue this right. So, we will say continue right.

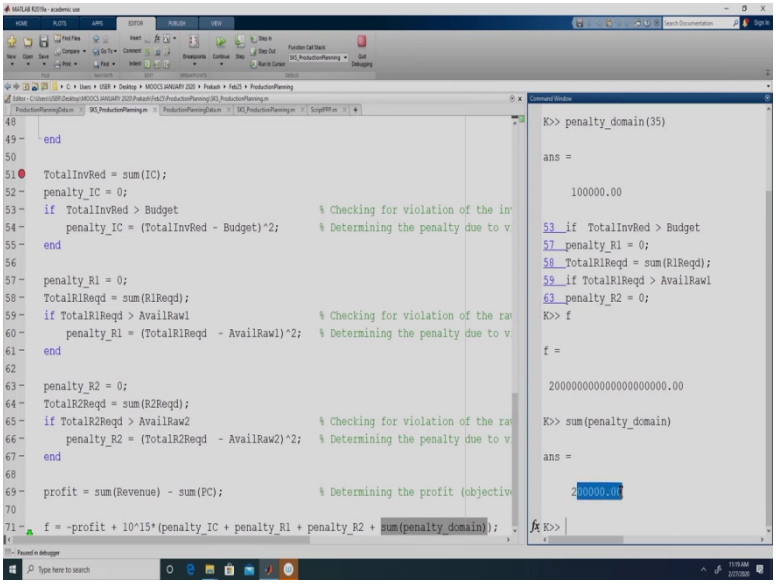
(Refer Slide Time: 37:57)



So, now let us see is what is the penalty underscore domain of the first variable that is 10 power 5 and the 35th variable that is also 10 power 5 right. So, it violated 2 domain constraints.

So, 2 of the processes are active right, but both of them are violating the domain constraint right. So, we need to assign penalty for that; that is what this piece of code has helped us to do right. So, we did not calculate production cost, investment cost, revenue raw material 1, raw material 2 because the solution itself is an infeasible solution right. We cannot produce 23 from process 1 and we cannot produce 90 from process 35.

(Refer Slide Time: 38:36)



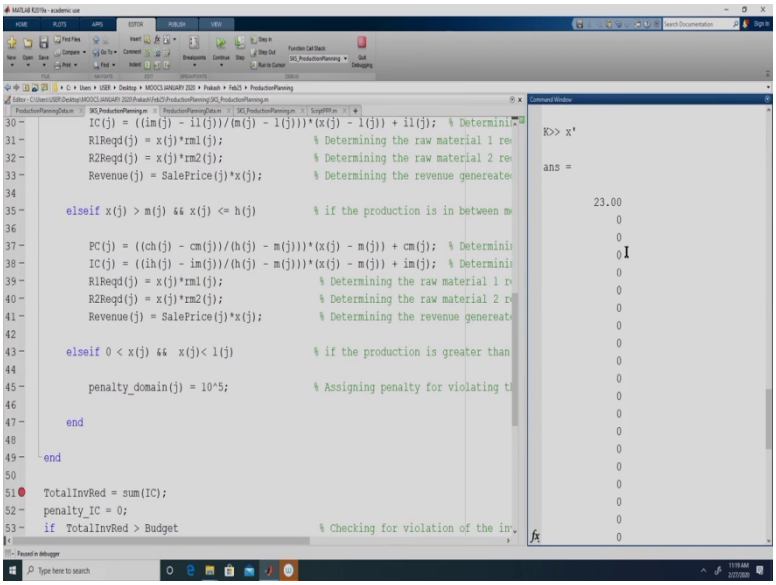
So, all of that would be 0 right. So, if you remember the discussion which we previously had investment cost is not calculated, production cost is not calculated, raw materials are not calculated, revenue is not calculated and profit from those processes would be 0 and since both the variables are violating in this case, the total profit is 0 right. So, that is what we are expecting. So, it did not go into this because total investment required is 0 raw material required is also 0 raw material 2 required will also be 0, the profit will also be 0 right because the variable themselves are not in the domain right.

So, there is no point of calculating profits which does not satisfy the domain constraint right. So, fitness here if we see, this profit part would be 0 other penalties are 0 right; penalty underscore IC underscore R 1 underscore R 2. These are 0 right. So, this is 10 power 5 2

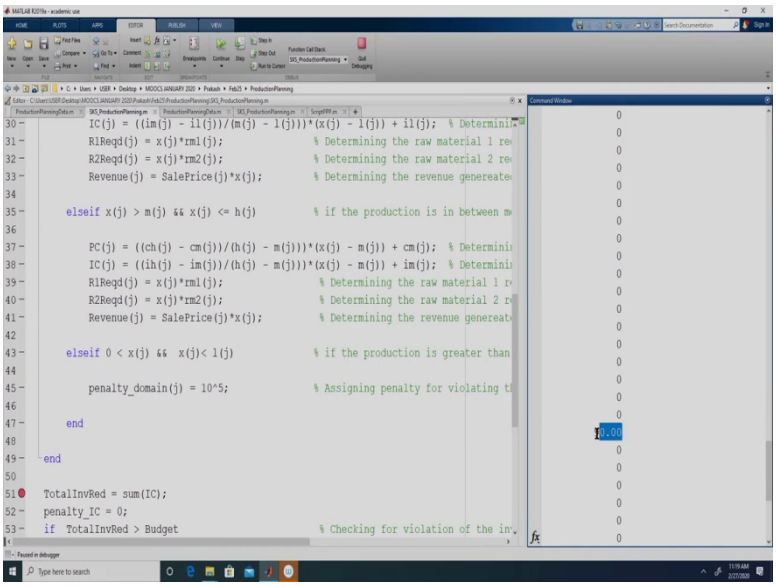
times. So, when we sum it up, it will be 2 into 10 power 4 multiply that with 10 power 15. So, that would be the fitness

So, here if we see the fitness so, this is the fitness value right. This fitness value came because we have this sum underscore penalty is right. So, here we have 2 into 10 power 5 right multiply that with 10 power 15 right. So, this is the fitness function of this solution right wherein the production from all the processes is 0 except for process 1 and process 35 right.

(Refer Slide Time: 40:01)

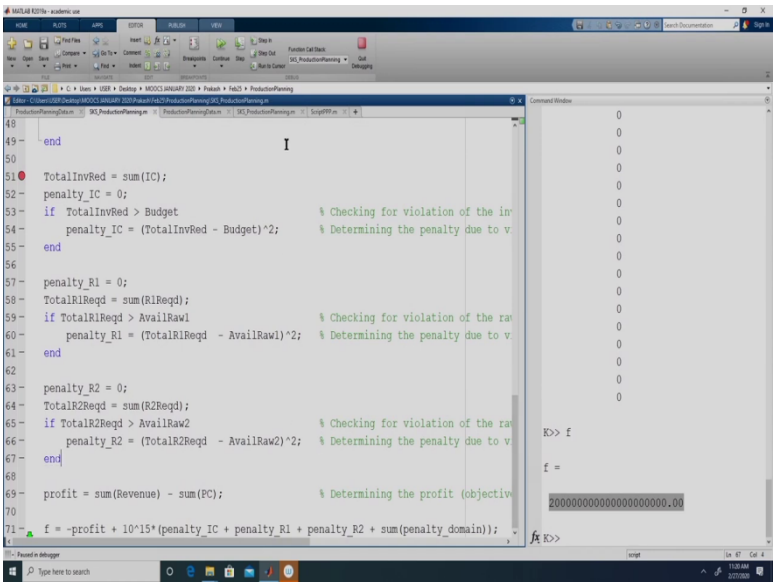


(Refer Slide Time: 40:03)



But the production which we have taken here as 23 and 90 right is infeasible.

(Refer Slide Time: 40:14)

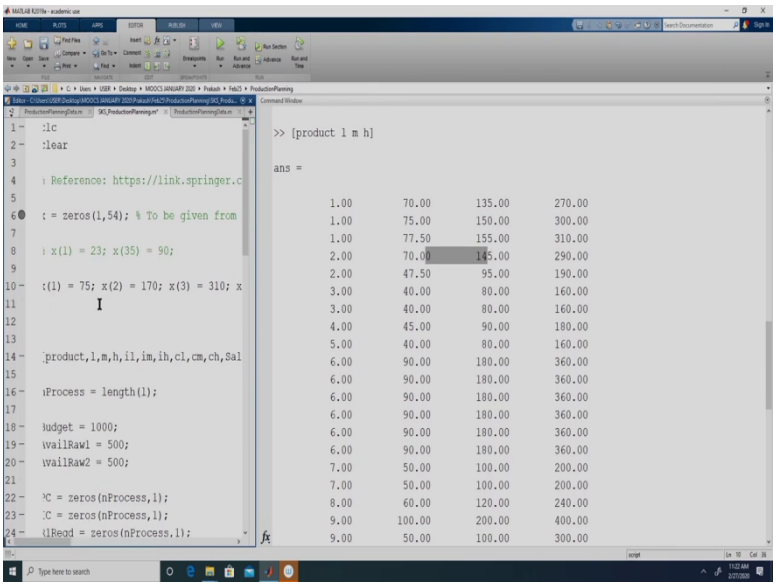


Since the solution is infeasible, we get a very high value of fitness function right. When we say a very high value of fitness function, it is a poor solution because we are talking from the perspective of minimization right; we want this f to be as low as possible.

So, when we started with we work with the vector full of 0s right, the first case when x was completely 0s that was at least a feasible solution which says we are not producing anything right. So, the fitness function for that turned out to be 0 whereas, for this solution which is an infeasible solution, the fitness function turns out to be very high value as you can see it turns out to be a very high value right.

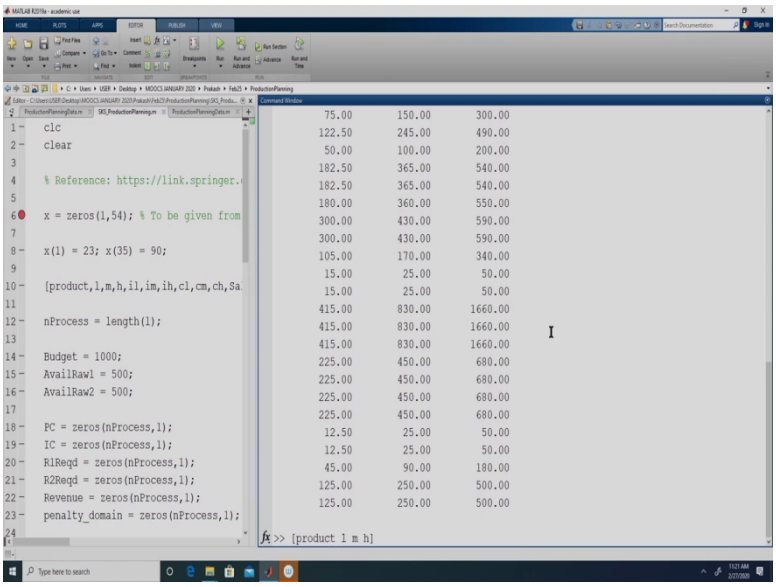
So, let us complete this right. Previously we saw trivial solution which is full of 0s, then we looked at an infeasible solution right.

(Refer Slide Time: 41:00)



So, now let us see for a feasible solution, how does it calculate. So, to give a feasible solution right let us have a look at the l m h values right. So, what we shall do is we will take process 1, 2, 3 and 4 to be active right.

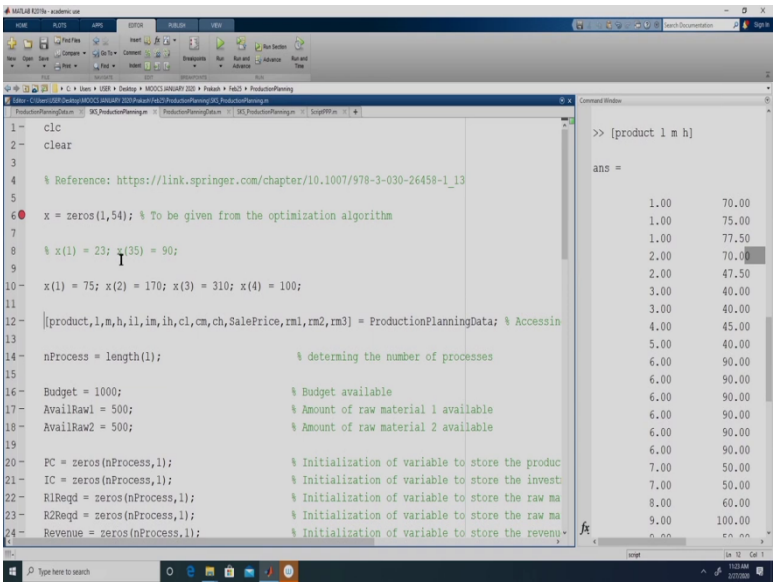
(Refer Slide Time: 41:21)



So, if you remember the product vector right; so, this product vector will tell us what is the product that is being produce. So, the first 3 processes are producing product 1 and the next 2 processes are producing product 2. Let us give a feasible solution which captures all the dynamics. So, x of 1 let us say it is 75 right because it will fall in the range l m right. Let us say x of 2 is in between m and h right. So, for process 2 the m value is 150 and the h value is 300. Let us take it as 170 right. So, x of 3 let it be in this range between 150 to 310.

So, let us say it is 310 right and let us also look into one more product right. So, let us also give some value for x of 4. So, let us say x of 4 is in this range 70 to 145; 70 to 145, let us say it is 100.

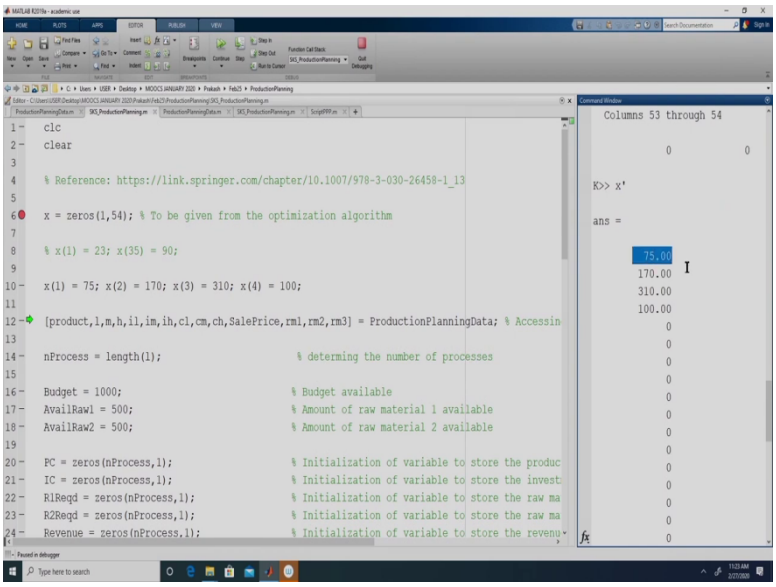
(Refer Slide Time: 42:20)



So, you need to remember that we are not still solving the optimization problem, we are just making sure that this function which we are going to club with an meta heuristic optimization algorithm is actually calculating the values appropriately right. Once we club with meta heuristic optimization technique, we will be relying on the solution whatever we get right.

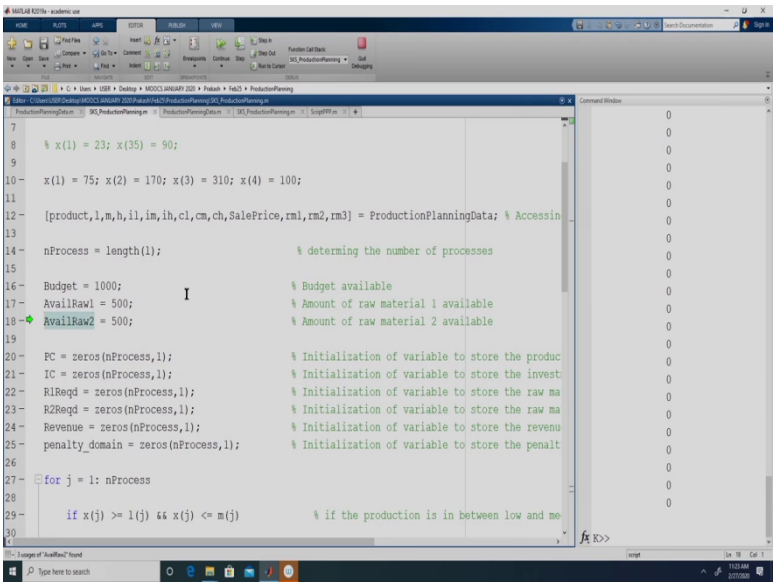
So, we are just making sure that whatever function we have written is actually calculating the values properly because the optimization algorithm is going to rely on these values. Let us execute this now right.

(Refer Slide Time: 42:48)



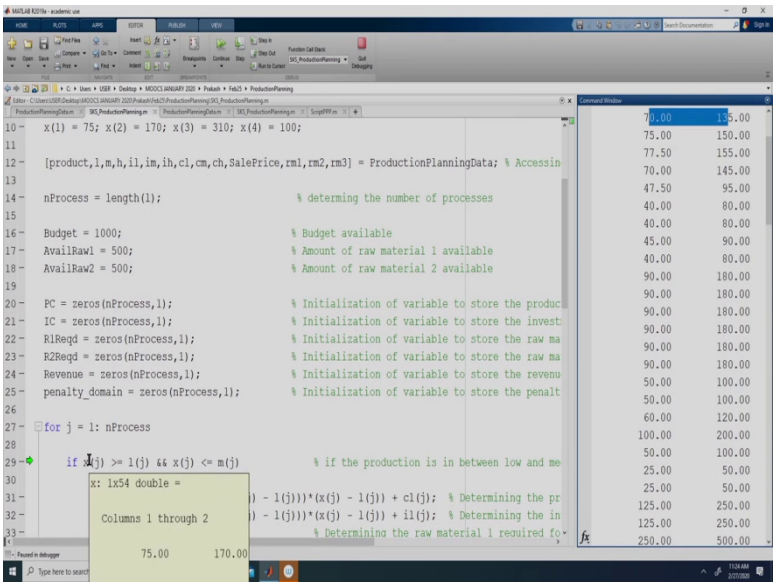
So, 0s; so, x vector is defined x transpose. So, the first 4 values are defined as we wanted right, here we are just calling the data right. So, I can just step out right.

(Refer Slide Time: 43:03)



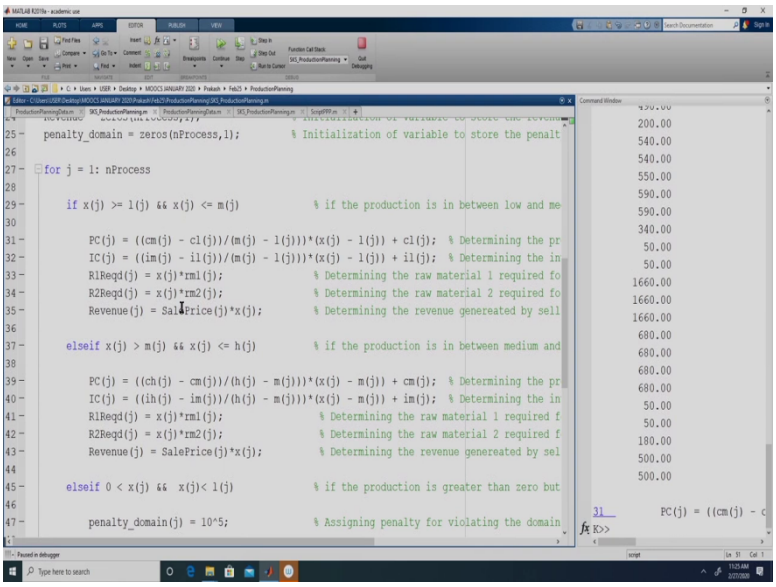
This is number of processes we are defining, we are seeing how much budget how much raw material 1, how much raw material 2 is required. This we are initializing

(Refer Slide Time: 43:12)



So, let us say what is l m h. So, x of 1 is 75. So, that is between l and m right. So, we expect it to go into this section.

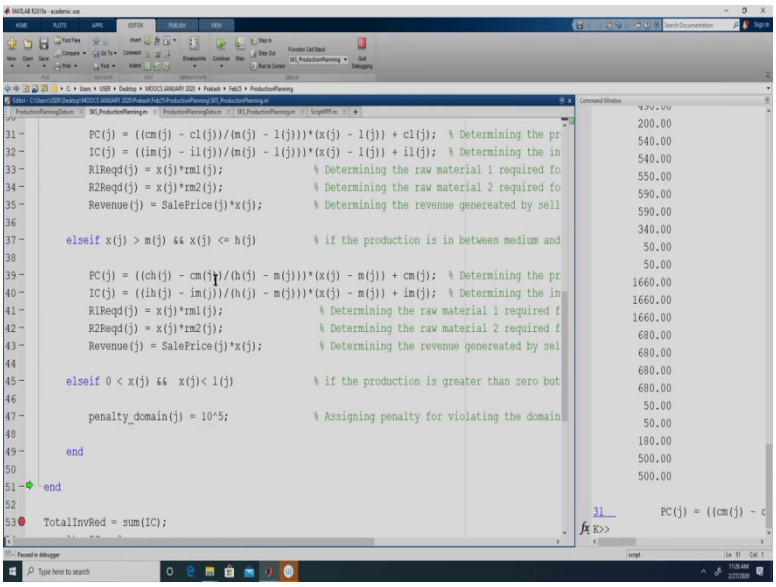
(Refer Slide Time: 43:23)



As expected it goes into this section because x of 1 is 75 and l of 1 is 70 right. So, it calculates the production cost 53.73 is the production cost. This is the equation which we had shown you previously right. So, it calculates the production cost according to this, it calculates the investment cost 57.01, the raw material 1 that is required is 71.10 right.

So, rm 1 is 0.95 and x of 1 is 75. So, 75 into 0.95, it gives us 71.1. Similarly we are calculating what is the amount of raw material 2 that is required right. So, the raw material 2 that is required for this production is 0 because rm2 of 1 is 0. So, the revenue is 73.3 right. So, the sale price of 1 is 0.97 and we are producing 75. So, 75 into 0.97 gives us 73.13 right. So, that is for process one right.

(Refer Slide Time: 44:19)

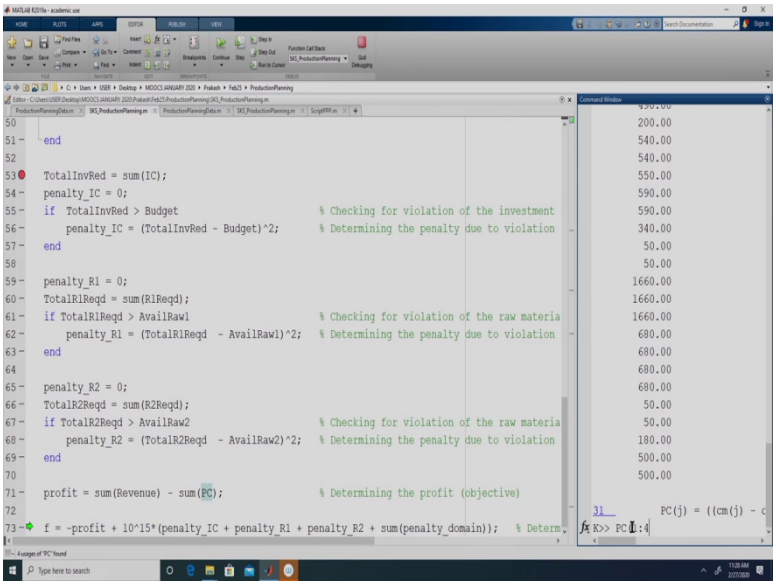


So, process 2 if you remember, we had taken a value such that it goes into this section right. So, here again we are calculating the production cost right 116.1 2 investment cost is 91.41. So, raw material 1 that is required is 160.34, raw material 2 required is 0 the revenue earned from selling the products which come out of process 2 is 164.74 right and this we can continue for the other processes right. So, for x 3 also it goes between m and h right. So, for x 4, it was 100 right. So, it goes into this range right l of 4 is 70 right and m of 4 is 145. So, 100 is in between that.

So, production cost is 70.06, investment cost is 66.3. Similarly we calculate raw material 1 that is required. So, raw material 2 required is 0 right and the revenue from the process 4 is 97.5 right. So, since we had only 4 active processes right for the rest of the processes, it is 0.

So, it will not go into any of this conditions right. So, the production cost and the investment cost will be 0 for rest of the processes. Let us continue right.

(Refer Slide Time: 45:35)

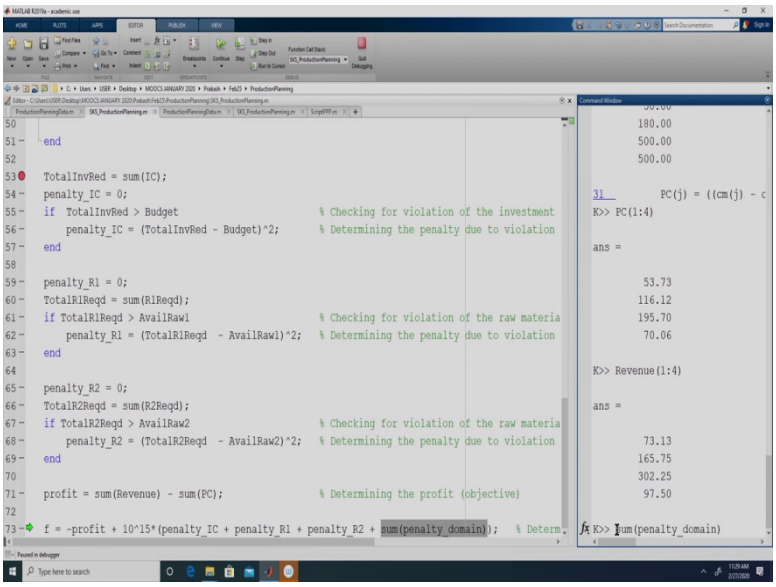


So, now we have calculated. So, now we have calculated for all the processes. So, the total investment cost is 348 right which is nothing, but the summation of these 4 values. So, penalty we are initializing as 0 right. So, budget what we had was 1000 right. So, what we require is 348.81 right. So, 348.81 is not greater than thousand so, no penalty right. So, that is why I did not go into this line. Similarly raw material 1 that is required is 621.09 and what we have is 500 right.

So, this production right which does not violate any of the domain constraint actually violates the raw material constraint 1 right. So, raw material 1 that is required 621 and what we have is only 400 right. So, the penalty we calculate by 621.09 minus 500 the whole square. So, that is

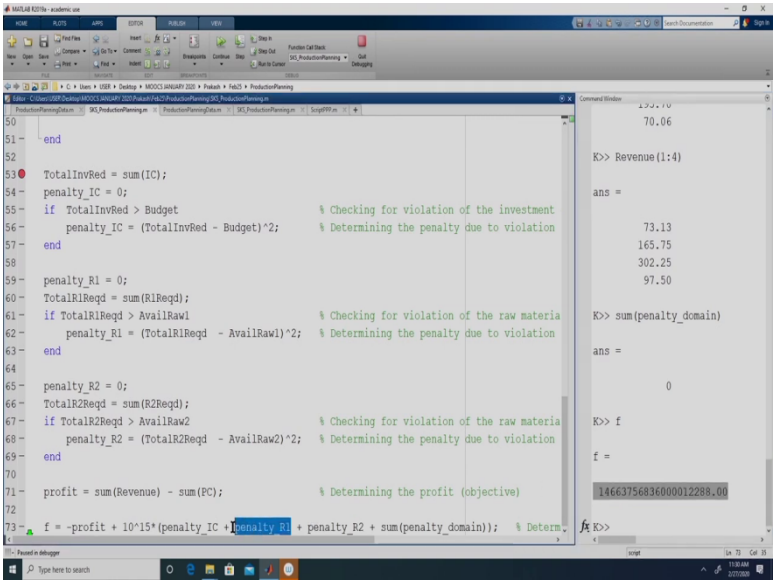
the penalty with respect to raw material 1 right. So, raw material 2, there will not be any penalty for this production plan because none of the 4 processes use raw material 2 right. So, the profit for this is the sum of revenue minus the sum of production cost right.

(Refer Slide Time: 46:51)



So, production cost for the first 4 processes are this one and the revenue earned from these 4 processes is this one right. So, the difference between them is the profit right. So, we get a profit of 203.01 right. Though this production plan has a profit associated with it right, but we cannot implement it because it violates the raw material constraint right. So, let us see what will be the value of fitness. So, this will be 203.01, this is penalty underscore IC is 0, penalty underscore R 2 is 0 and the sum of penalty domain will also be 0 because all the 4 active processes do not violate the domain constraint right.

(Refer Slide Time: 47:31)



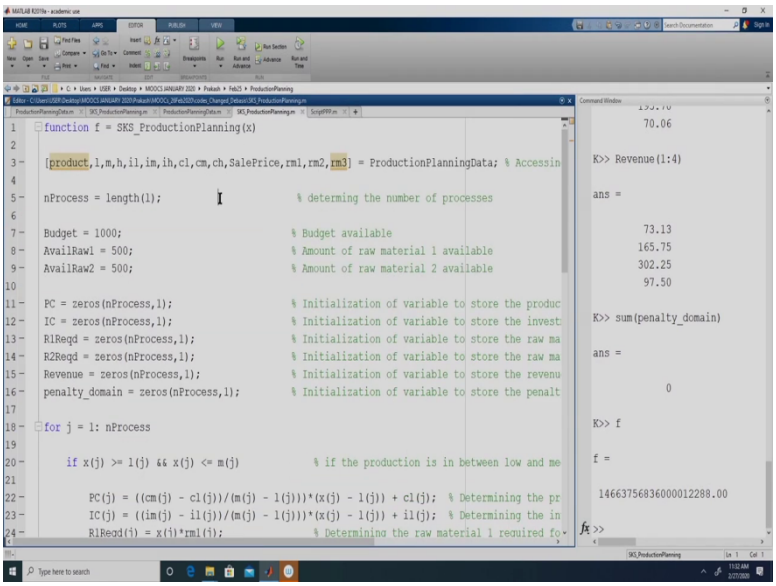
So, this is also 0 right; only penalty we have is the penalty with respect to the raw material 1. So, that we are multiplying by a very high factor 10 power 15 and we are calculating the fitness for it right. So, f is this high value right. So, it is a high value because it is a infeasible solution. The infeasibility is due to the violation of the constraint related to raw material. So, that is how this fitness function file is going to work right, given a solution it is now helping us to calculate the fitness of that solution. If you carefully analyze, you will be able to understand; if it is a feasible solution, then the penalty value is going to be 0 right.

So, 10 power 15 into 0 is going to be 0 and profit would be a positive value right, but fitness is equal to minus of profit. If the fitness value which is f in this case if f is negative, we can be confident that it is a feasible solution because that will happen only when the penalty is 0 and the penalty will be 0 for a feasible solution and whatever the value of fitness function is comes from only the profit. So, for this case profit is a positive quantity and we are taking negative of

that right. So, for this case study at least right so, whenever we get a negative value we can be confident that it is a feasible solution right.

So, now that we have looked into this, now let us plug it with an optimization algorithm. To plug it with an optimization algorithm we need to convert into a function file because remember that we will be calling this function multiple times. So, if you are executing let say teaching learning based optimization, we will be calling it in during the initialization on of population, will be calling it multiple times in teacher face, we will be calling it in learner face right. So, this file which is a script file as of now needs to become a function file.

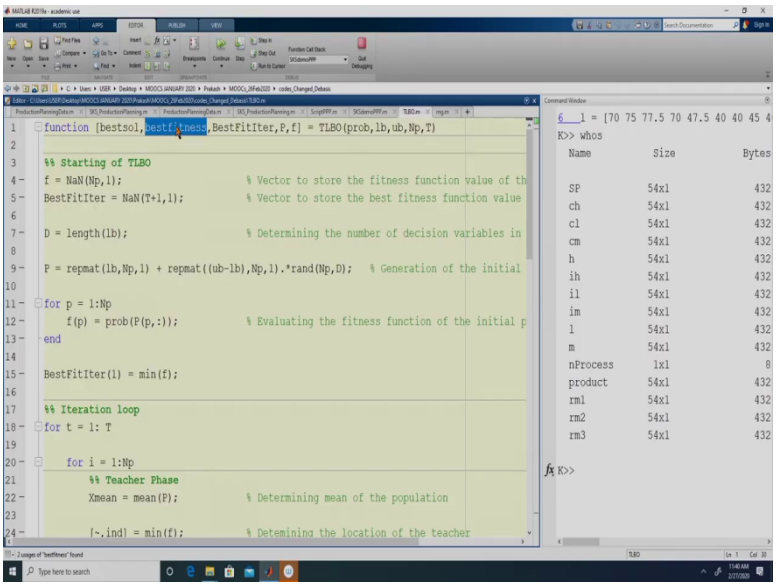
(Refer Slide Time: 49:18)



So, here we have that function file right. So, what we have done is we have removed that x values right because the x values are now going to come from an algorithm and this function is merely going to return the fitness function value f right. So, what we will now do is we will

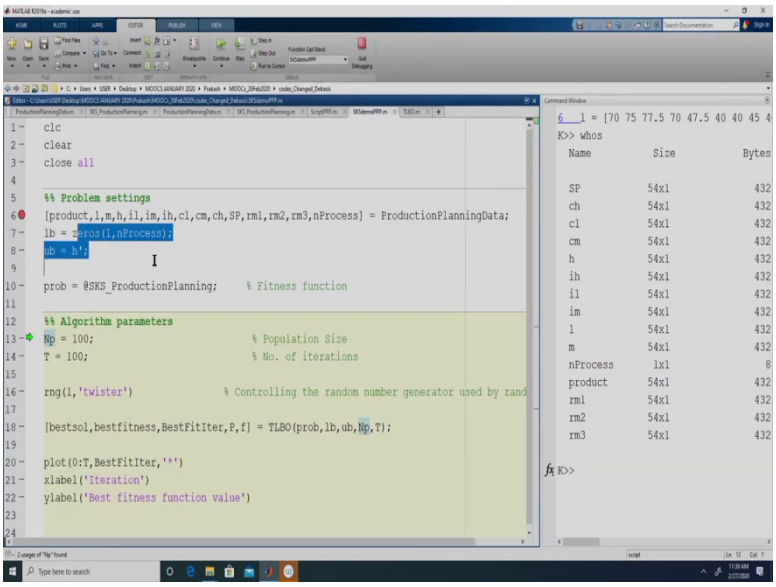
plug it with teaching learning based optimization right. So, this is the teaching learning based optimization algorithm which we had developed right. So, we are not going to modify anything in this algorithm.

(Refer Slide Time: 49:33)



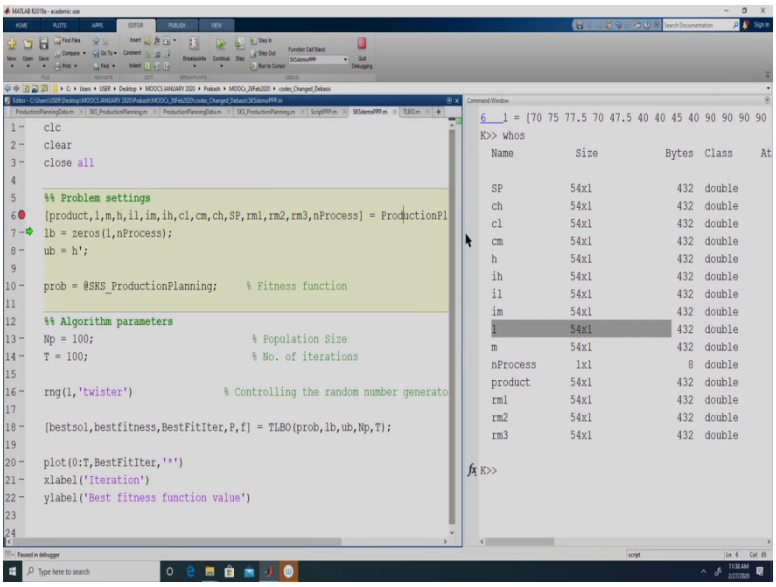
So, we will now solve it with teaching learning based optimization right. So, this is the code that we had developed previously for TLBO right.

(Refer Slide Time: 49:54)



So, what we will do now is we will a develop a script file right. So, we have this script file right. So, for this function right, we need to provide the lower bound upper bound; we need to tell the name of the file where the fitness function is calculated. We need to give the population size N p and we need to give the number of iterations that is to be performed right. So, that is why we are defining over here. So, the upper bound is the h value right. So, instead of directly giving the h value we are calling it from the file production planning data right.

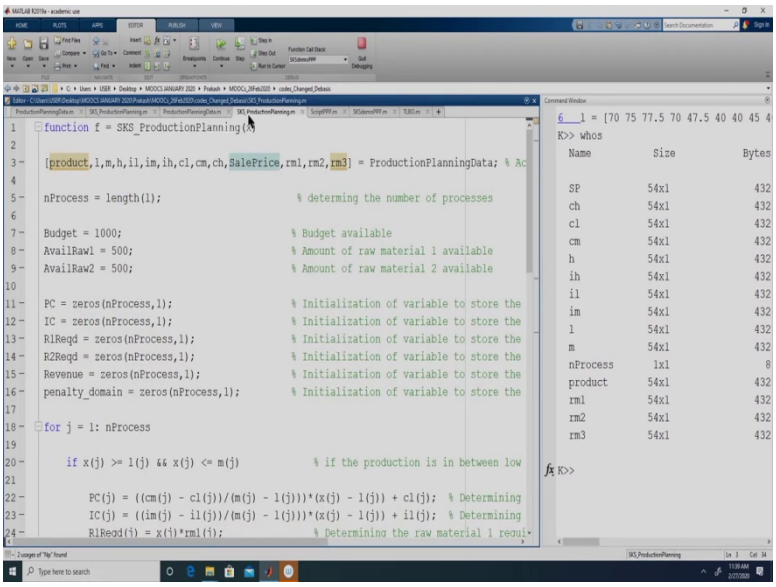
(Refer Slide Time: 50:20)



What this line 6 will do is it will help us fetch all the data right ah. So, let me just step out of this function right. So, now, all the data which we want is over here right. So, for example, selling price production cost of low level, medium level, high level right; the investment cost related to the 3 levels l values, m values and rest of all the values are available in this script file because this was your function file. Remember this function file only contains the data right. So, we did not supply any input to this function file, we merely called this function file and it helped us with all the details related to the problem right.

So, lower bound as we discussed it is not l, but it is 0s; 0s of 1 comma n process upper bound is the h value right. So, since we needed the h value over here that is why in line 6 we had to call this function file right. So, this upper bound is available now right.

(Refer Slide Time: 51:16)



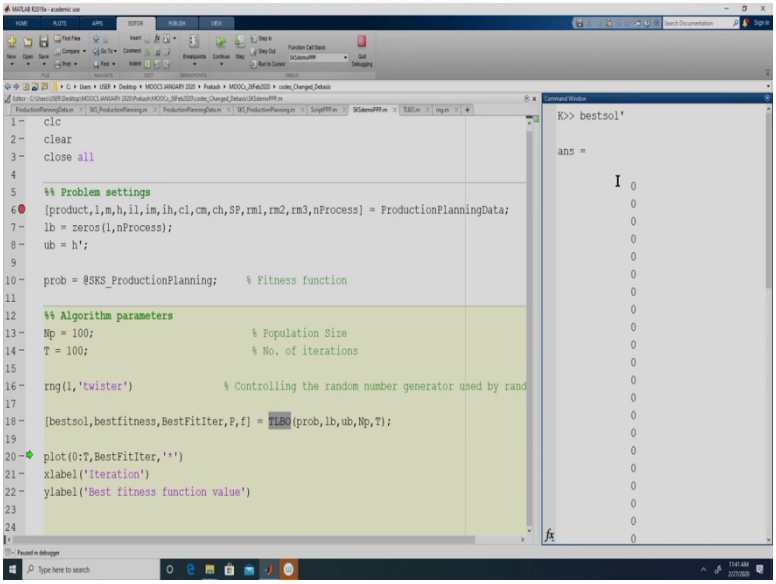
So, the problem is in this file right. So, SKS underscore ProductionPlanning that is what we have given over here. So, the lower bound is given, the upper bound is given, the file to determine the fitness function value is also given right. So, with respect to problem, we will required 3 things which have been defined right this N p is the population size, T is the number of iterations right.

So, this rng we have discussed previously, it helps to control the random numbers right. So, that if required we can regenerate the results right. We are fixing the algorithm to be twister algorithm and we are fixing the seed to be 1. So, now, what we are doing is we are providing all the required details to the TLB algorithm; only now that we are solving it as an optimization problem right. So, what we expect from the TLBO algorithm is the best solution right, the solution as in like what are the decision variables.

So, from which process how much is to be produced that is what will get in best sol and for that solution what is the fitness function value that is what we will get with this variable best fitness. The variable best fit iter will tell us the best function value right in every iteration, P is the population that we have in the last iteration and f is the corresponding fitness function value.

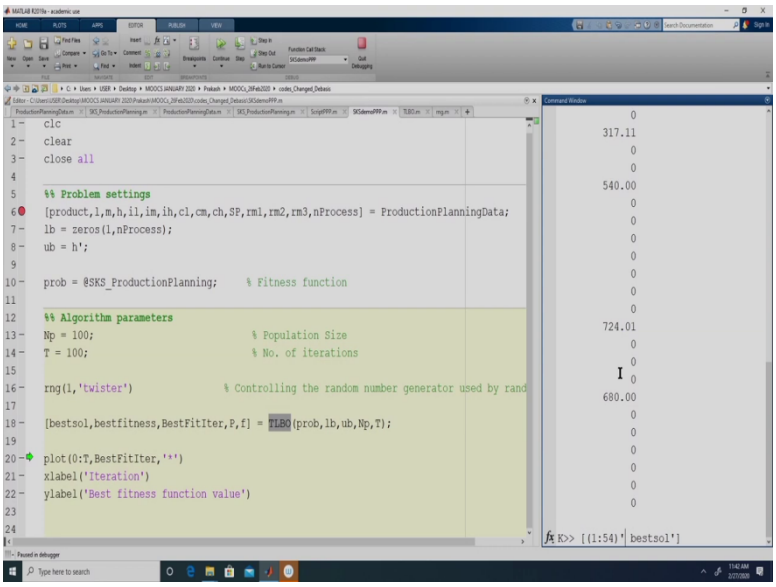
So, now, it executed right. So, we did not execute line by line because by now you would know what this algorithm is doing right. So, we did not execute it line by line, it has completed everything right.

(Refer Slide Time: 52:41)



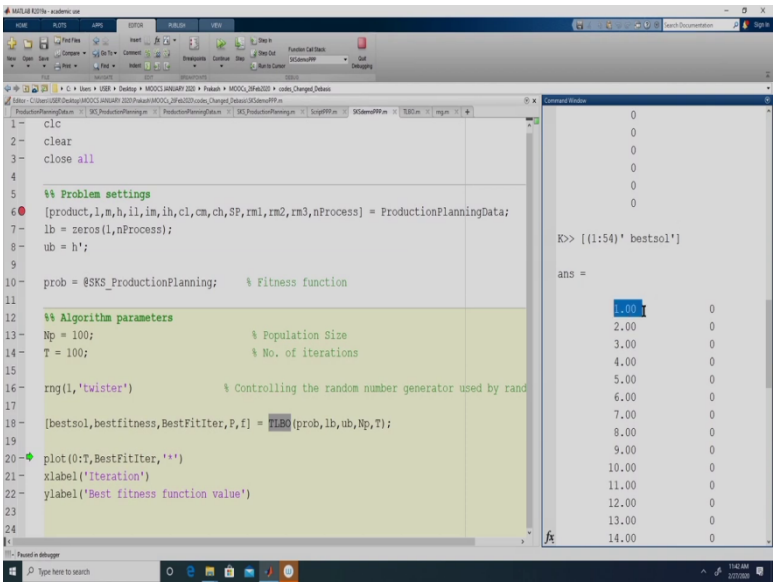
So, now if you look at this best sol bestsol; so, this is the solution it has right.

(Refer Slide Time: 52:50)



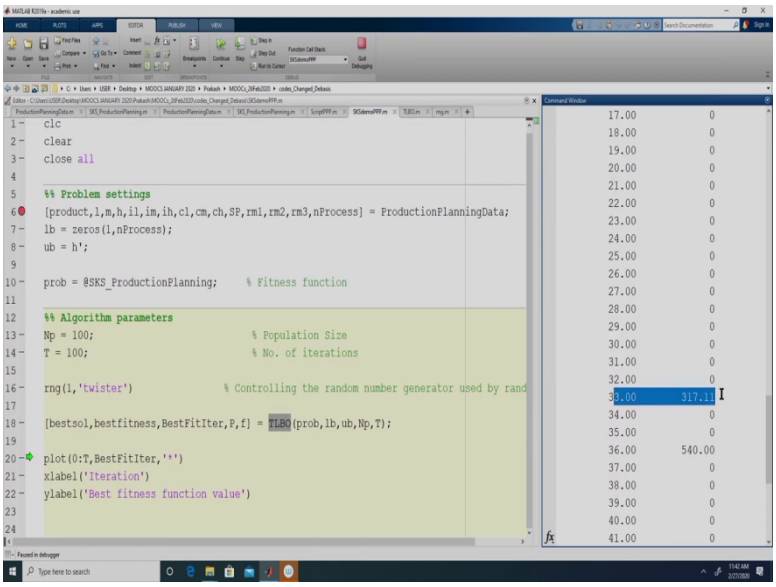
So, all this the production from all this process is 0, from this process it is 317.11 00 540 and again 0s, 724, 680 right. So, these are the decision variables right. If you want to know from which process we are getting what is the value, what we can do is the way we have arranged the decision variable is the first variable will indicate what is the production from process 1, the second variable will indicate what is the production from process 2 right.

(Refer Slide Time: 53:16)



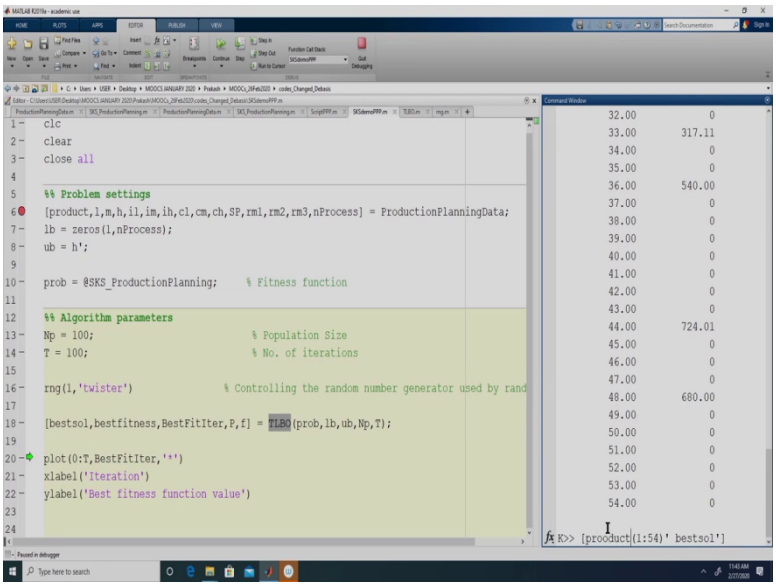
So, here what we did was from 1 to 54 we generated a vector right, it will generate a row vector and since we want the column vector we use this transpose symbol right. Similarly bestsol what we got was a row vector. So, we are transposing it to get a column vector. Now we have appended this column. So, that we know from which process how much is to be produced right.

(Refer Slide Time: 53:35)



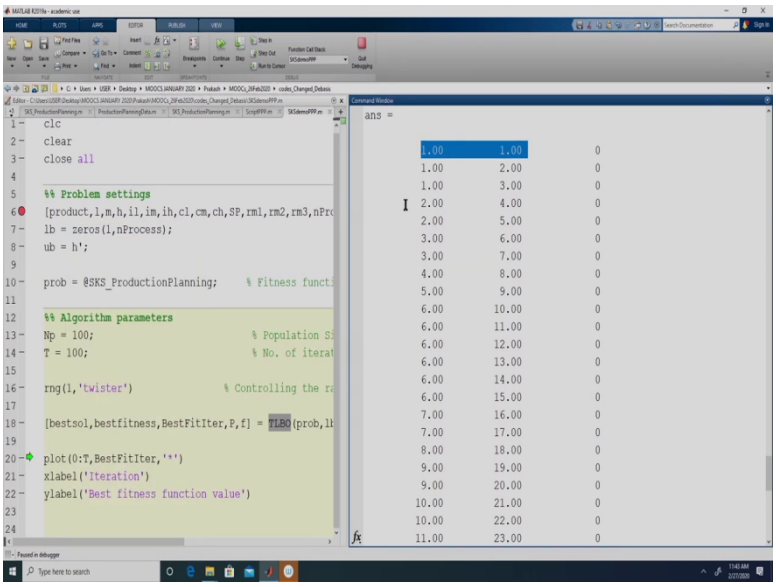
So, here if we see nothing is produced till process 32 right. So, process 33 we are supposed to produce 317.11, process 36 we need to produce 540 units.

(Refer Slide Time: 53:46)



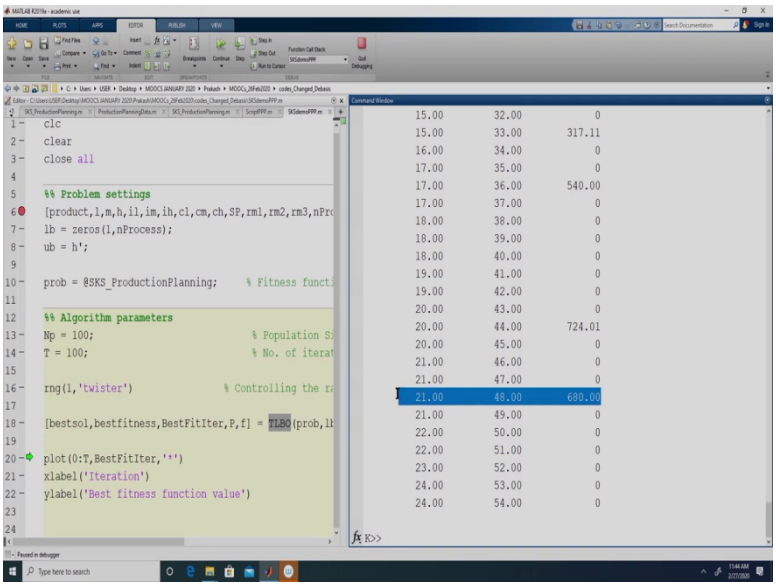
Process 44, we need to produce 724.01, process 48 we need to produce 680 right and that is it. So, only 4 processes are active right. If you want to know what is the product als so, we can have here product right.

(Refer Slide Time: 54:07)



So, this product vector we had discussed that each value will tell what is the product that is produced. So, for example, the first 3 values are 1 so; that means, process 1, process 2, process 3 produce product 1 that is how we are assigned this variable product right.

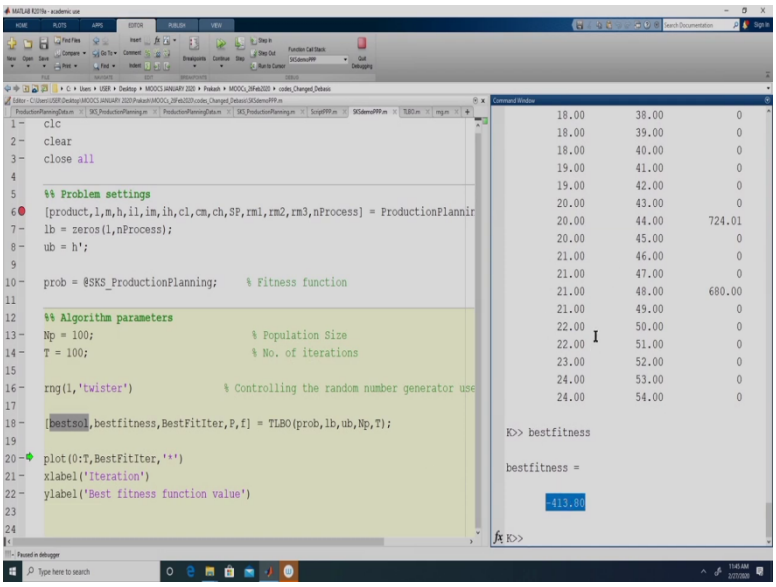
(Refer Slide Time: 54:26)



So, now we have product in the first column right. So, let us go to the non zero values. So, this is to be interpreted as from process 33, we need to produce 317.10 and the product that is produced by process 33 is product 15 right. Similarly this is product 17 is being produced from process 36 and the production quantity is 540. Over here the product 20 is being produced using process 44.

So, product 20, we can use 43, 44, 45, but the algorithm has chosen 44 for the optimal solution. So, product 20 from process 44 the amount of production is 724.01. Similarly over here product 21 is being produced from process 48 and the amount of production is 680. So, that is how we analyze this bestsol right.

(Refer Slide Time: 55:15)

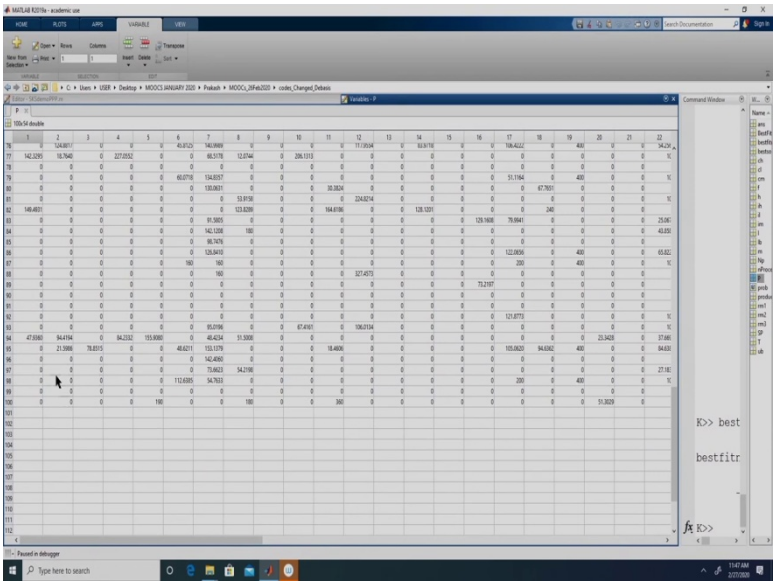


So, let us look at what is best fitness. Best fitness is the fitness function associated with this solution right. So, it is negative we had discussed previously at least for this case study with this data a negative value will indicate a feasible solution. So, this is a feasible solution what we have is a feasible solution. So, if you go back and plug this solution into this file SKS underscore ProductionPlanning, you can calculate what is the raw material that is required what is the investment cost that is required.

So, all that post optimality analysis can be done right. So, for this solution the fitness value is minus 413.80. For the current case this will happen only when the penalty is 0. So, the profit that we have is 413.80. Remember what the algorithm has solved is a minimization problem and what we had is a maximization problem right.

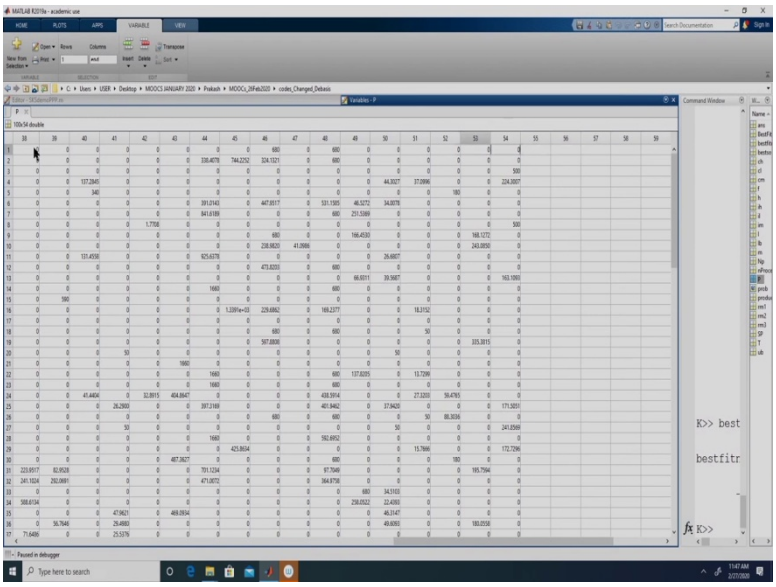
So, whatever value we get from the algorithm, we need to multiply it by a minus sign right. So, the actual profit is 413.80. Though it shows minus 413, it shows it with respect to the fitness, but the profit is 413.80 right. We will come to this BestFitIter a little bit later right, let us look at what is the population what is P right. So, P is the population in the last iteration.

(Refer Slide Time: 56:37)



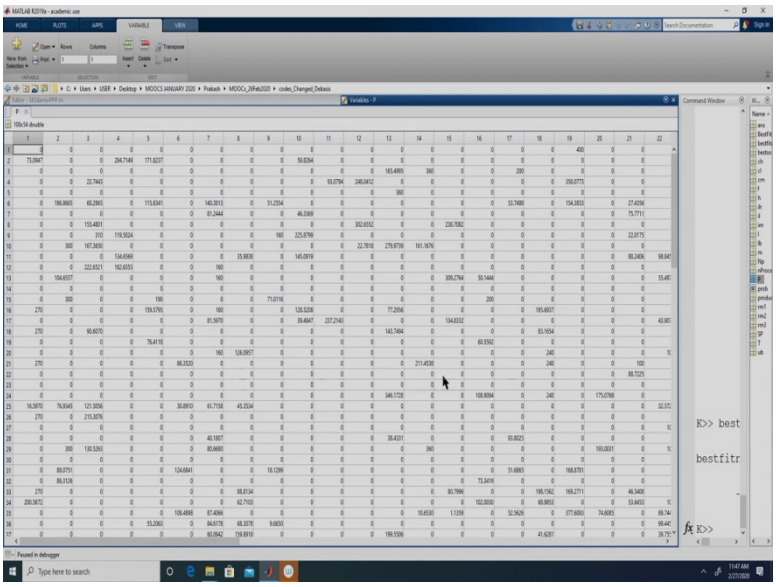
So, since it is going to have 54 values, let us look it in the workspace later. So, the P is over here. So, this we will have 100 rows right because each row indicates the population number and each column indicates the variable. So, we will have 54 columns right.

(Refer Slide Time: 56:52)



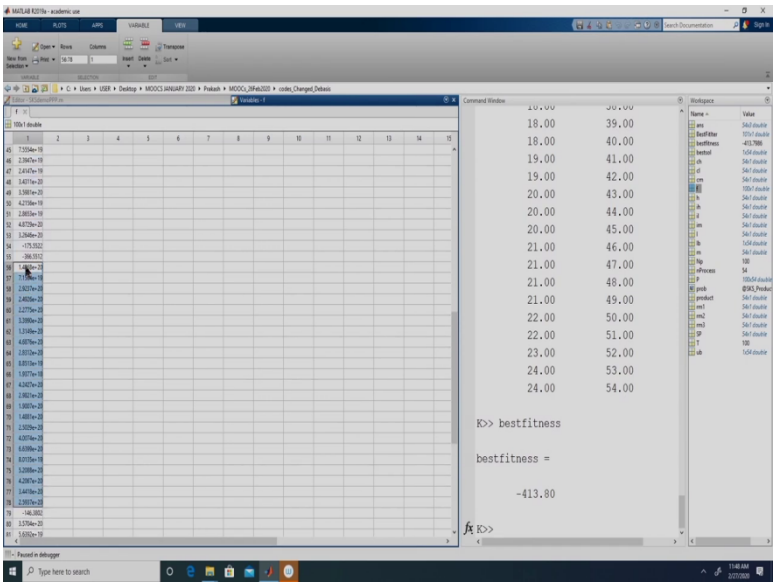
So, if we keep going we should have value till 54 right. So, each of this is a solution right.

(Refer Slide Time: 56:57)



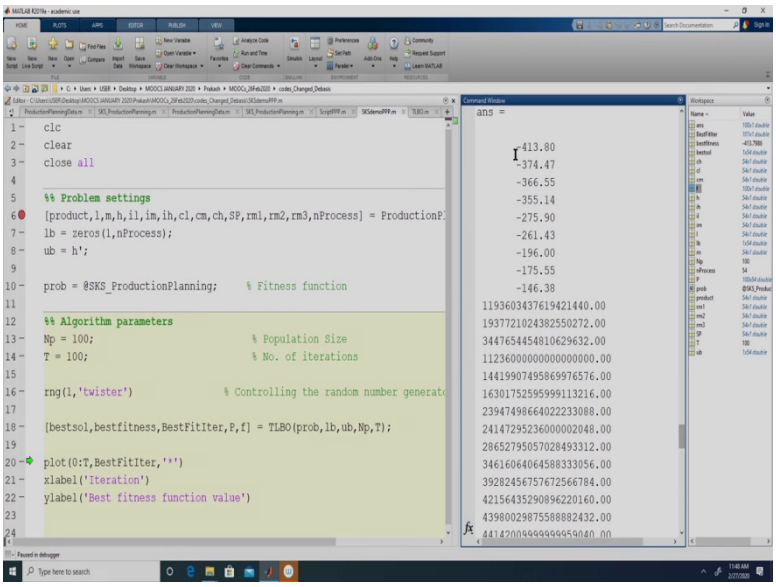
So, you can see that lot of these values have become 0. So, 0 means from that process, we are not producing anything right. So, the fitness function corresponding to each of this solution. So, these are the solution; not only the optimal solution the algorithm also gives us 99 other solutions right. So, if you want to look at the fitness of this 100 solution, it will be in the vector f. Let us look into the vector f over here right.

(Refer Slide Time: 57:21)



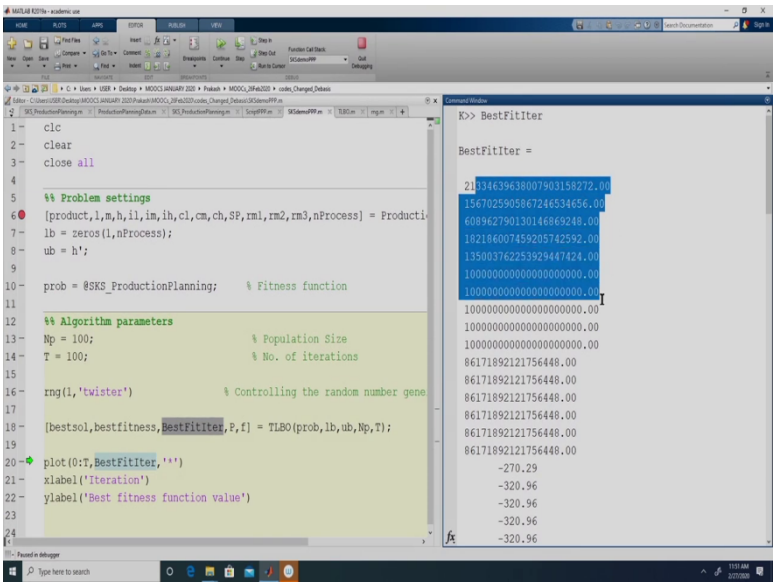
So, here if we see many of these solutions are still infeasible right. All of these are infeasible only those which have a negative sign are feasible right. So, this is the fitness function corresponding to each of the 100 solutions.

(Refer Slide Time: 57:40)



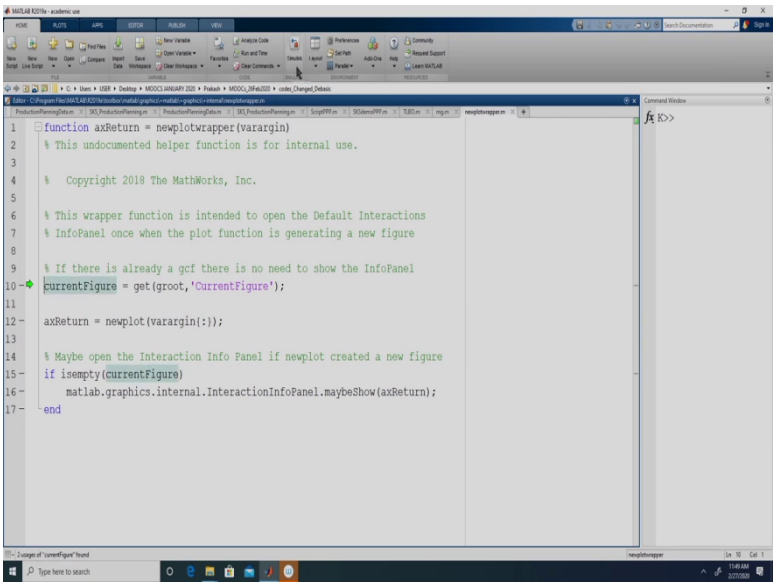
As a small post optimality analysis what we can do is we can say sort of f right. So, this is f that has been sorted right; so, out of 100 solutions 3 4 5 6 7 8 9. So, even after 100 iterations right out of the 100 population number, we have only 9 feasible solution. So, the rest of the solutions are infeasible solutions that is the meaning of this P and f right.

(Refer Slide Time: 58:05)



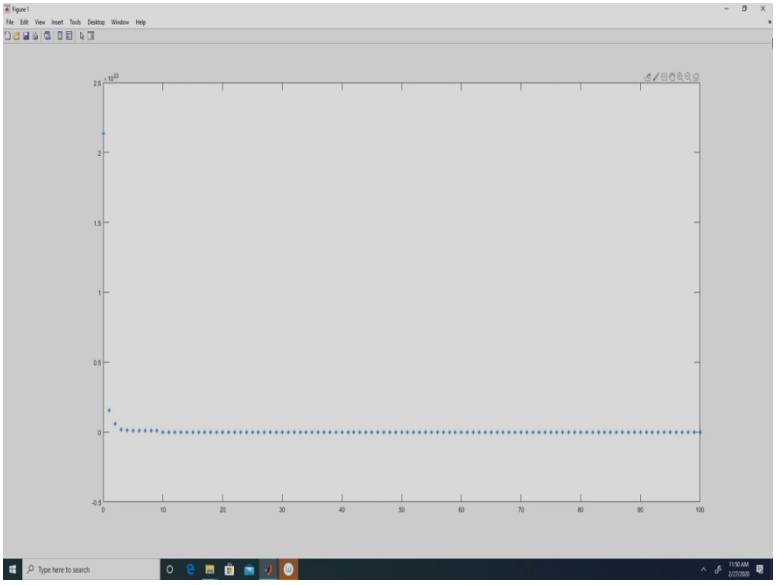
So, now let us look at BestFitIter right. So, BestFitIter we can actually plot right because if you remember the previous discussion that is the value for convergence curve right. So, what we are you doing over here is we are plotting on the x axis 0 to T; 0 to T because BestFitIter will have 101 values though our number of iterations are 100 because it will also store what is the best value in the initial population right

(Refer Slide Time: 58:32)



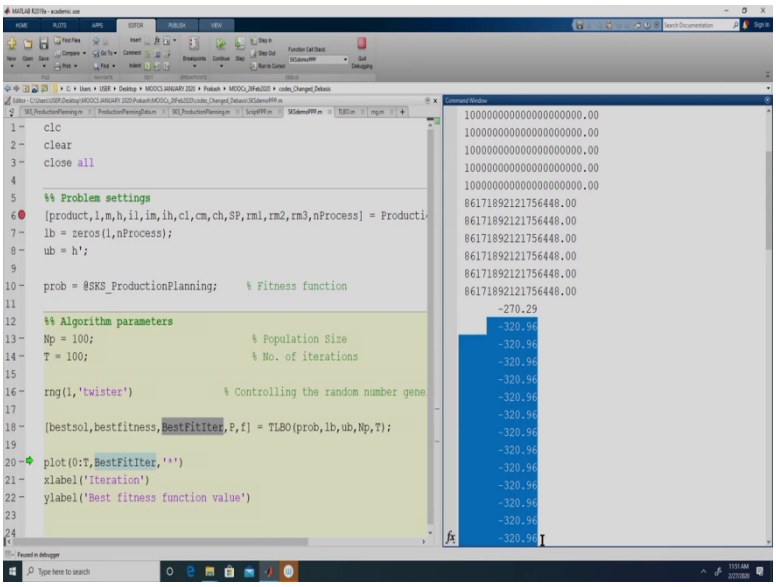
So, let us have a look at that figure right.

(Refer Slide Time: 58:35)



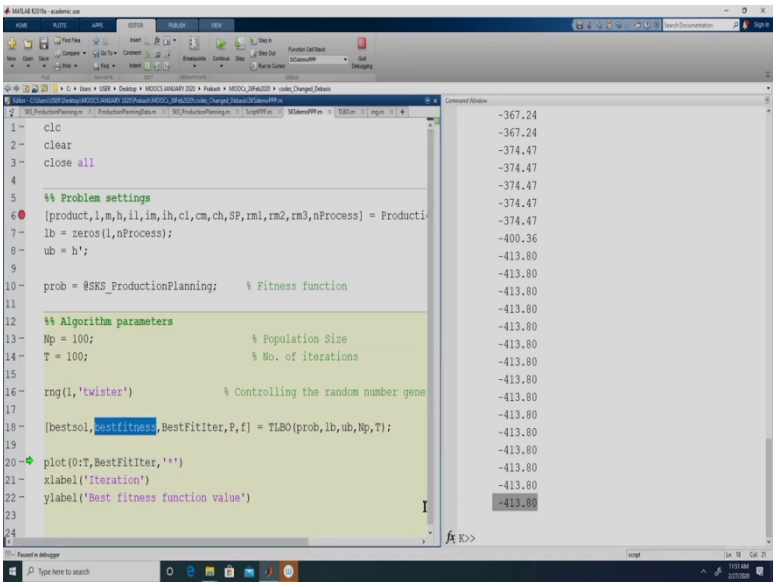
So, this is the convergence curve. So, if you look at this convergence curve, it might seem like nothing much is happening after let us say 11th or 12th iteration, but that is because this magnitude is very high 10 power 22 right. So, if you look at this BestFitIter values right so, initial population had a fitness of this one. The best member in the initial population had a fitness of this value right and as iteration progressed the solution started to improve right.

(Refer Slide Time: 59:04)



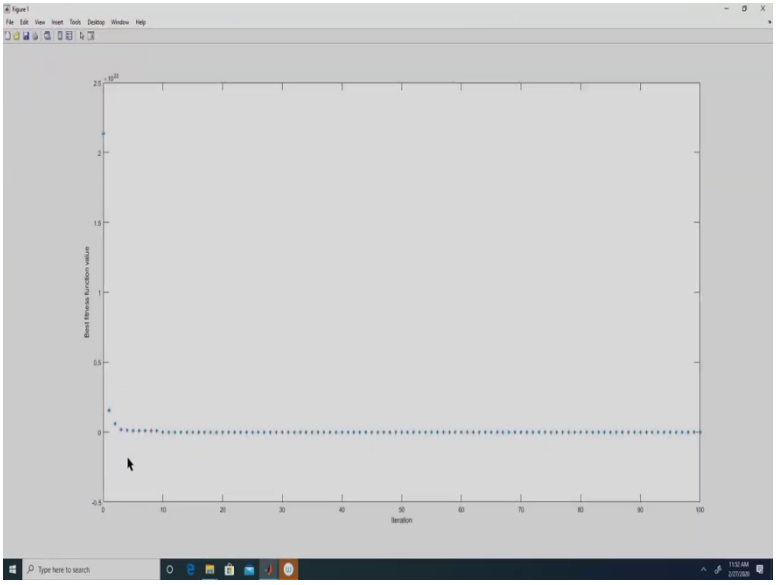
And even after getting this minus 270, it kept improving this minus 320 trade on for considerable number of iteration and then minus 338.

(Refer Slide Time: 59:20)



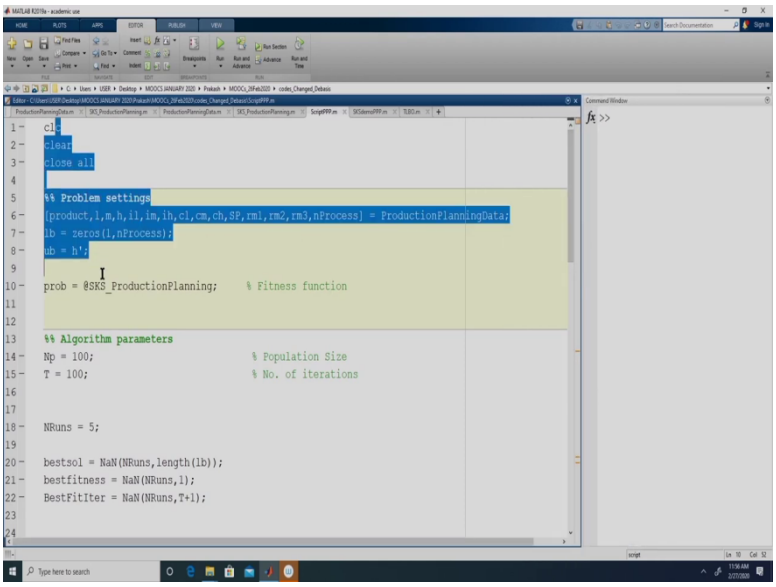
And minus 362 right, minus 367, minus 374, minus 400, then 413 right and so on. So, minus 413.8 is what it is the best value that we have. So, this should be minus 413.80 right.

(Refer Slide Time: 59:34)



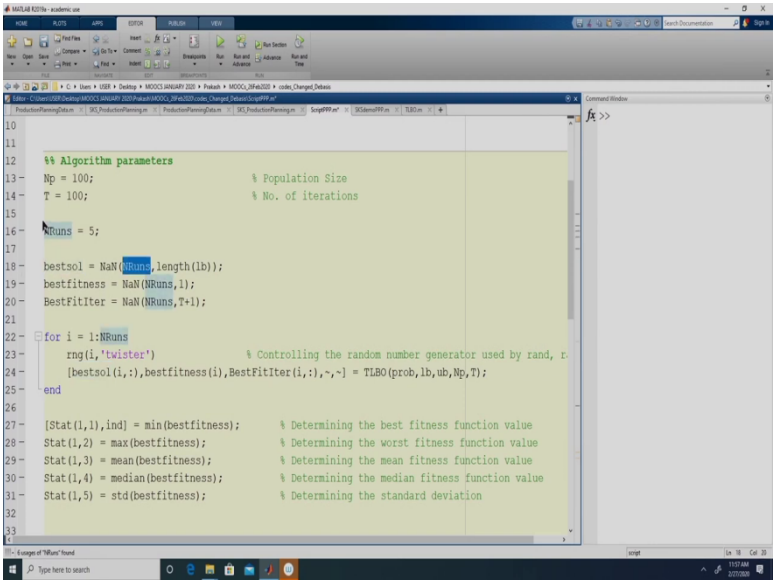
Let me just remove this breakpoint and let us see the figure directly right. So, this is the convergence curve. As you might remember right so, the sarcastic techniques are to be run multiple times right with multiple runs have to be done by changing the seed of the random number generator right. So, that is what we will do now on this problem.

(Refer Slide Time: 59:50)



So, let us do that right. So, here what we have done is the major part of the code remains the same right. So, this we have discussed right. So, this is these 3, these 4 lines help us to get the problem details right.

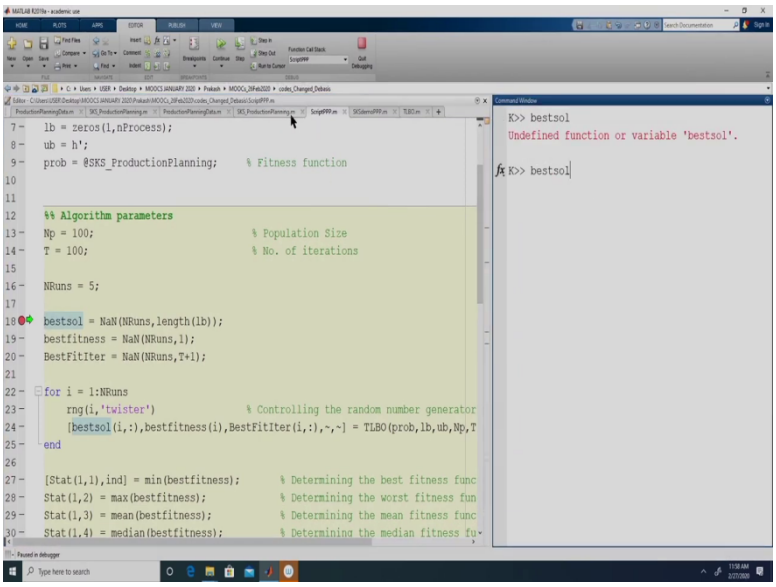
(Refer Slide Time: 60:07)



This is the algorithm parameters and let us say now we choose to run 5 times. Just to demonstrate it to you we are restricting our self with 5 runs. This best sol vector previously when we had contains the decision variable right. So, and it is going to contain 54 decision variables even for a single run right. So, what we are doing is here we are defining the matrix bestsol. So, it will have 5 rows because the number of runs is 5 and the number of columns is the length of lb

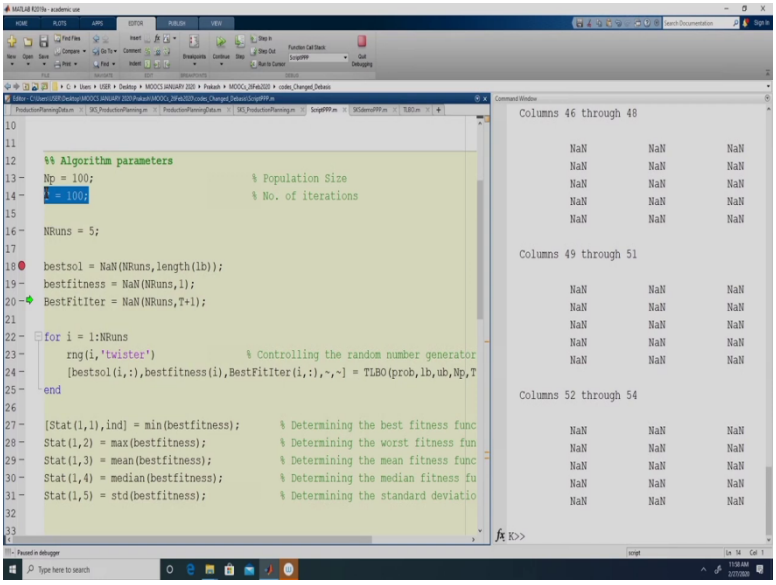
So, the length of lb is 54 for the current set of data right, but if we change this data this will appropriately change with respect to the number of decision variables right. So, that is this bestsol.

(Refer Slide Time: 60:46)



So, bestsol if we look at it is the line has not been executed right.

(Refer Slide Time: 60:57)

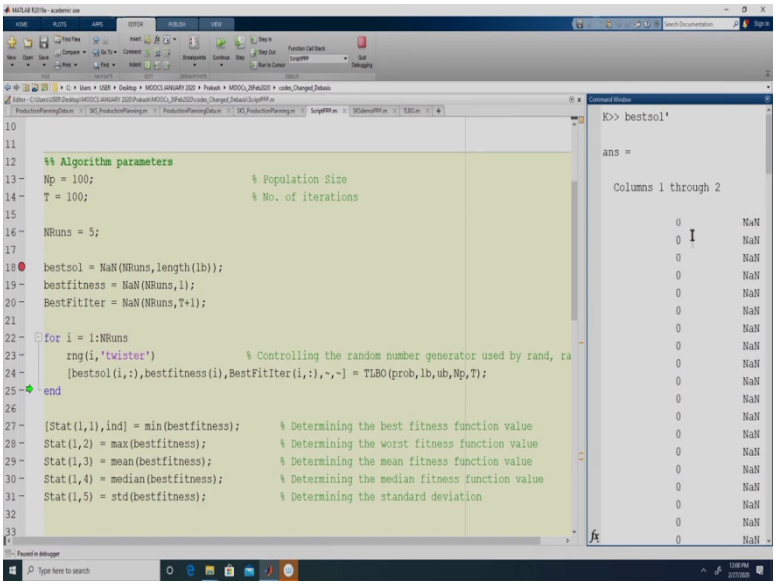


So, f 10 so, this is bestsol. So, it will have 54 columns and 5 rows ok. So, fitness corresponding to each of the solution will be stored in best fitness right. So, this will be a vector of 5 rows and 1 column. So, this convergence curve right will have T plus 1 values right.

So, T is the number of iterations, it will have the best fitness function value for the 100 iteration. In addition it also will have the best fitness function value for the initial population right. So, that is why this is T plus 1 right and we are executing 5 runs right. So, this BestFitIter is a matrix of 5 comma 101 in this case or T plus 1 f 10 right. So, we are executing it for 5 times. Every time we need to change the seed so, we are saying that we will run the twister algorithm with the seed as first as 1, second time as 2, third time as 3, forth time as 4 and the fifth time as 5.

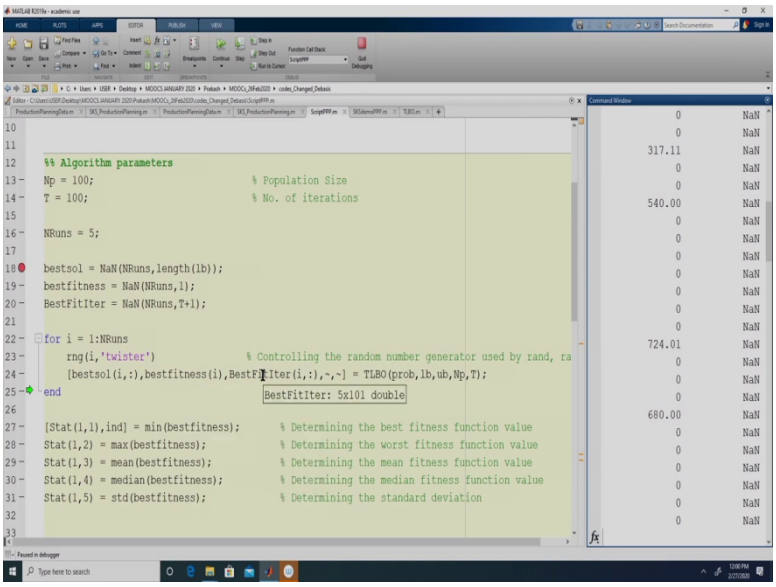
So, now, we are ready to execute the algorithm right. So, every time we send the same details right. So, but the answers could be different because the random number selected for each run is different right. So, we are not going inside this algorithm because we know what is happening over there.

(Refer Slide Time: 62:05)



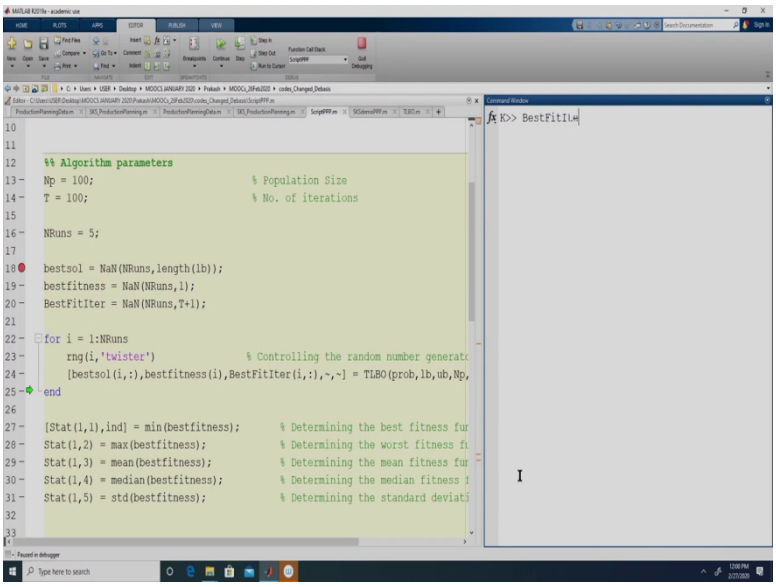
So, bestsol transpose, it has values in the first column because only the first run is complete right.

(Refer Slide Time: 62:11)



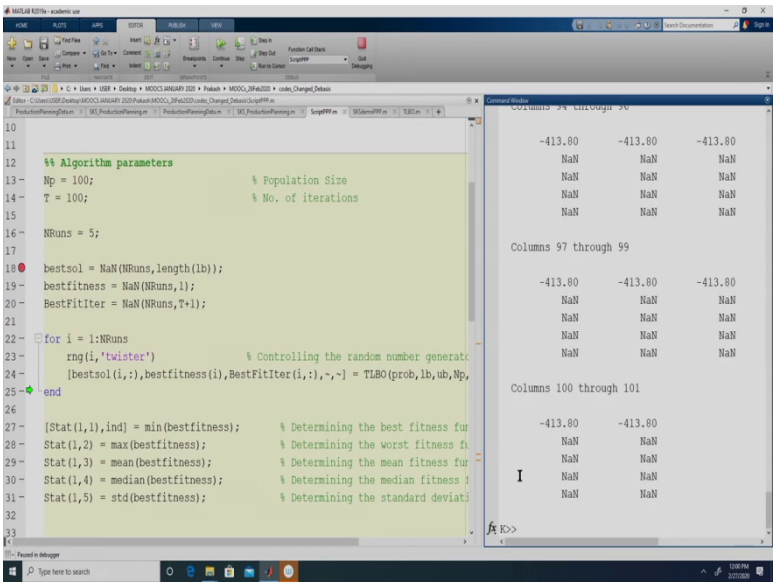
So, it has the values in the first column right, best fitness. Again if we see over here, the first value is minus 413.80 which is the same when we ran it for a single run right because that time we had use the same algorithm and seed was fixed 1 right. So, this best fitness is minus 413.80 for the first time. Similarly we will have this convergence curve.

(Refer Slide Time: 62:35)



In this case the first row would be populated right. So, BestFitIter right.

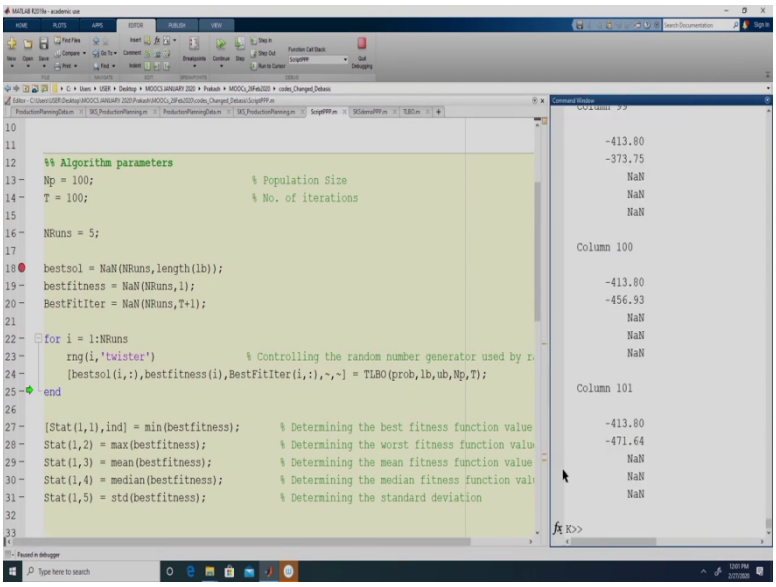
(Refer Slide Time: 62:40)



So, the first row is populated because the first row contains the convergence curve for the first run right. So, in this case, we are not looking at the final population and its fitness function. If you are interested you can save the final population and the fitness function of the final population in each run right. So, this is the second run right. So, this is executing it for the second time.

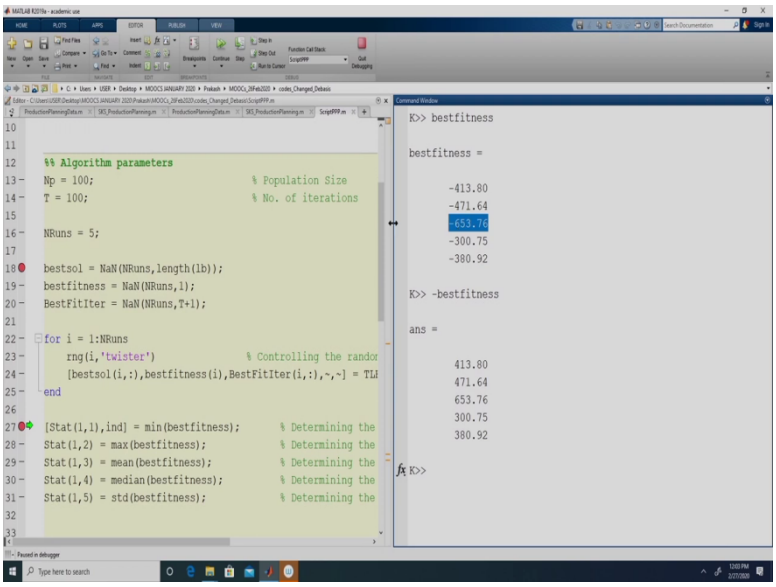
So, now if we see 2 values are populated. So, first time we got minus 413.8, second time we got minus 470.64 right. Similarly this BestFitIter will now have 2 rows which are populated right.

(Refer Slide Time: 63:16)



So, here if we see 2 rows are populated right. So, similarly the rest of the 3 runs could be executed right. So, we will just put a breakpoint over here and continue ah, it is executing the rest of the 3 runs right.

(Refer Slide Time: 63:35)



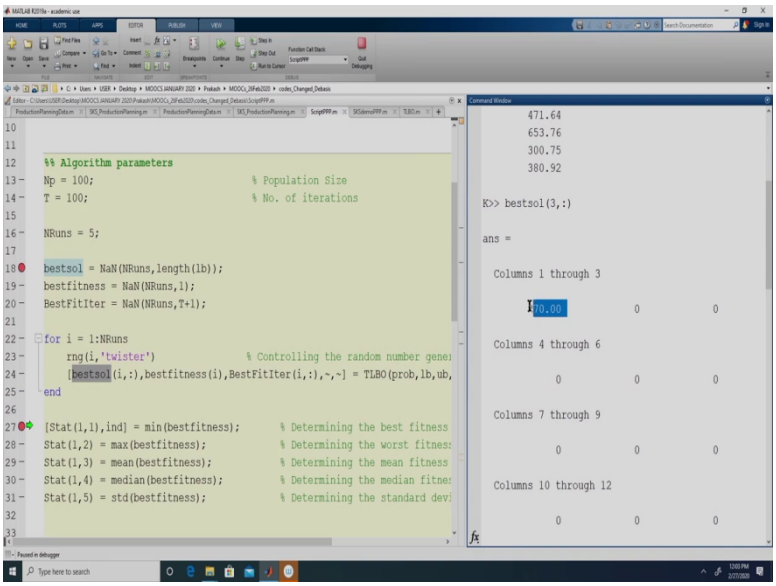
So, here let us wait for the command prompt. So, we have a command prompt now over here. So, best fitness if we see it has 5 values right. So, this is the best solution which we got in the first run, this is the best solution which we got in the second run, this is a fitness function value of the best solution in the third run and the forth run and the fifth run right.

So, right now if we see third run gives us the best solution right because it has the lowest value right. So, this best fitness is the fitness function value and it is for minimization right. If you are talking about profit then these are the profit right. So, the maximum profit we get is 653. 6 right or the minimum fitness is minus 653.7 6.

So, the third run gives us the best solution. We want to extract that particular solution right ultimately it is the solution which will help us to realize the objective function value right. So, if someone tells us that profit is 653.76 for some production plan. It is not useful to us right

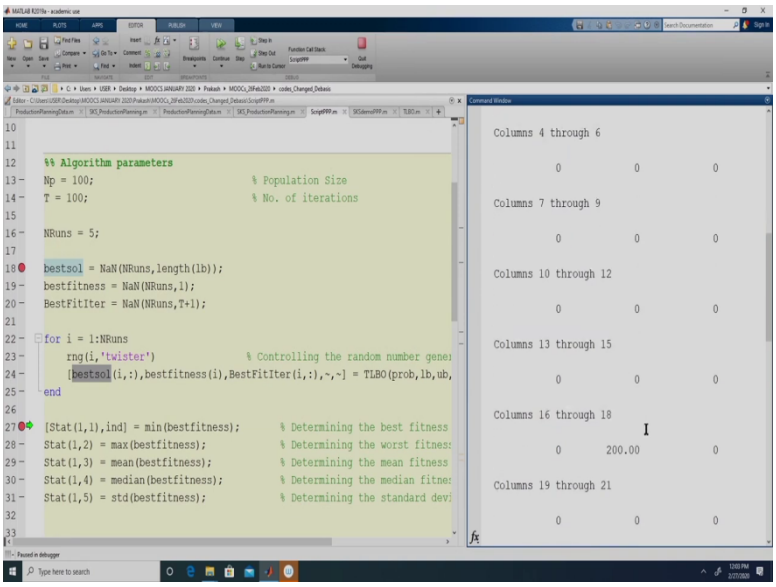
because unless we know the values of the decision variables, the value of the objective function hardly helps us to implement it right. So, it is necessary for us to get the decision variables right and the decision variables are in this bestsol right.

(Refer Slide Time: 64:50)

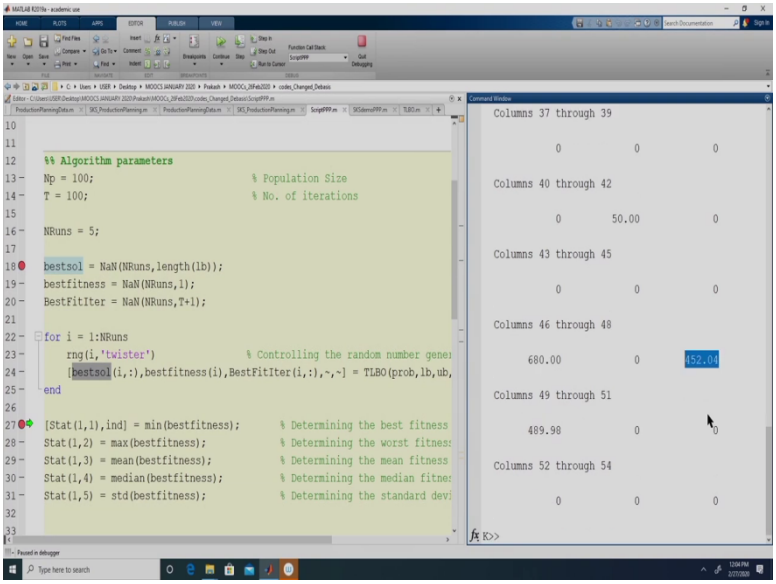


So, the best sol we are looking at the third row all the columns that is what we are interested right. So, third row all the columns is given over here. So, for the third run the best solution is process 1 is to be used to produce 270, process 17 right; this is 16, this is 17, this is 18 because it write 16 to 18 over here.

(Refer Slide Time: 65:01)

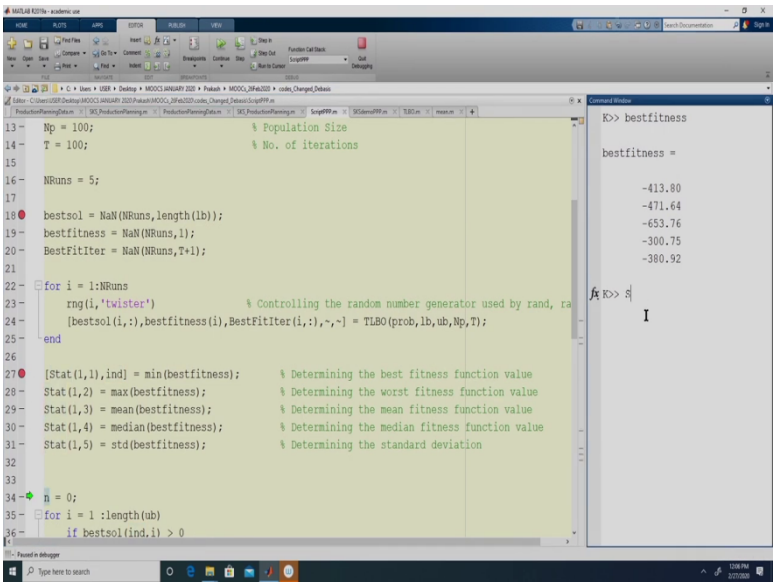


(Refer Slide Time: 65:11)



So, process 17 is to be used to produce 200 and then process 41 is to be used to produce 50 process 46 and 48 are used to produce 68 and 452.04 and process 49 is to be used to produce 489.98 right. So, that is the production plan.

(Refer Slide Time: 65:31)



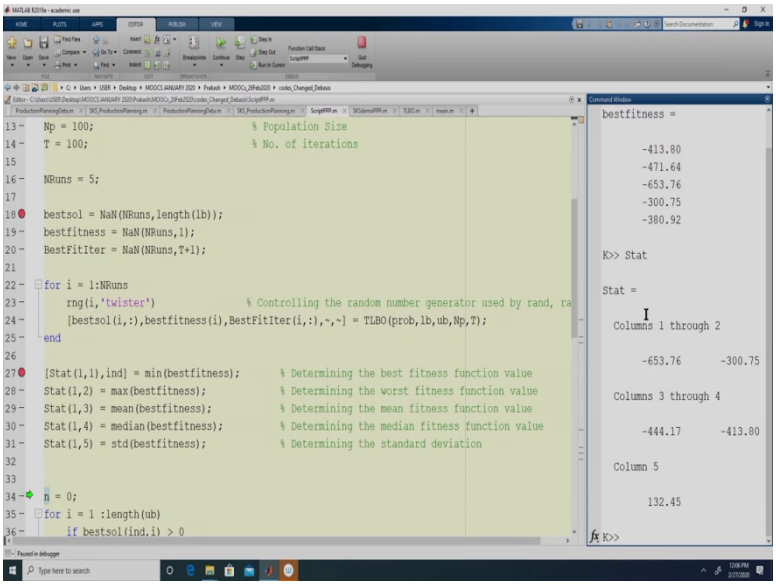
So, we can directly determine that by having this small piece of code right. Remember for this best fitness again we are supposed to do a statistical analysis. So, here in line 27 what we are saying is we are finding out the best fitness value right in best fitness. So, out of the 5 runs which is the runs that gives us the best solution since we have converted the problem into minimization right. So, min of best fitness gives us the best solution right. So, that value would be stored in Stat of 1. Remember this value Stat which we had previously used for doing statistical analysis right.

So, it is going to be a row vector, the first column is going to have the best run, the second column is going to have the worst run, the third column is going to give us mean of all the runs, the forth column is going to give us the median and the fifth column is going to give us the standard deviation right.

So, that is how we had arranged it previously. Stat will be a row vector; first column is best solution, second column worst solution, third column mean, forth column median and fifth column is the standard deviation in this 5 runs right. So, here what we are doing in addition to getting this value right of minus 653.76, we also getting the location which run it is occurring right. So, that will be stored in ind.

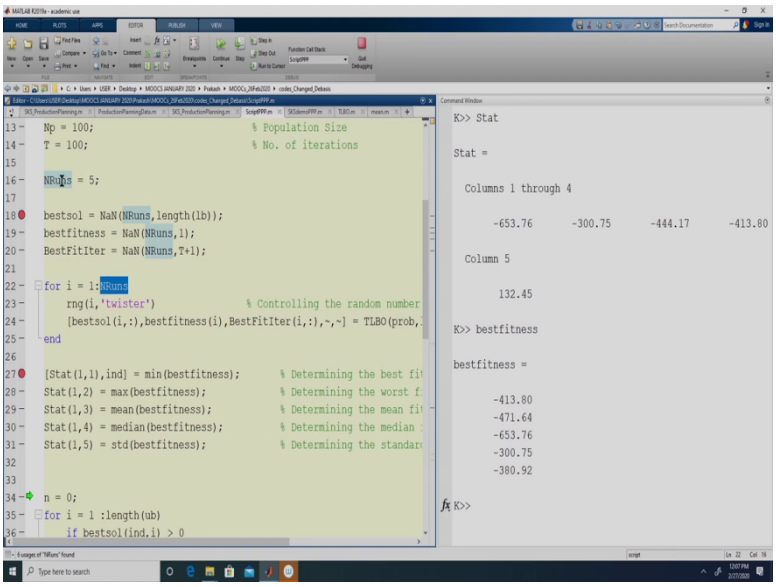
Remember this mean function we have used multiple times in addition to giving the best value, it can also give us where it is located right. So, where it is located is required so, that we can extract that solution from this bestsol right. So, that is why we are having this ind over here, it is just name of a variable.

(Refer Slide Time: 67:08)



So, if we just execute this so, now if we see this variable Stat.

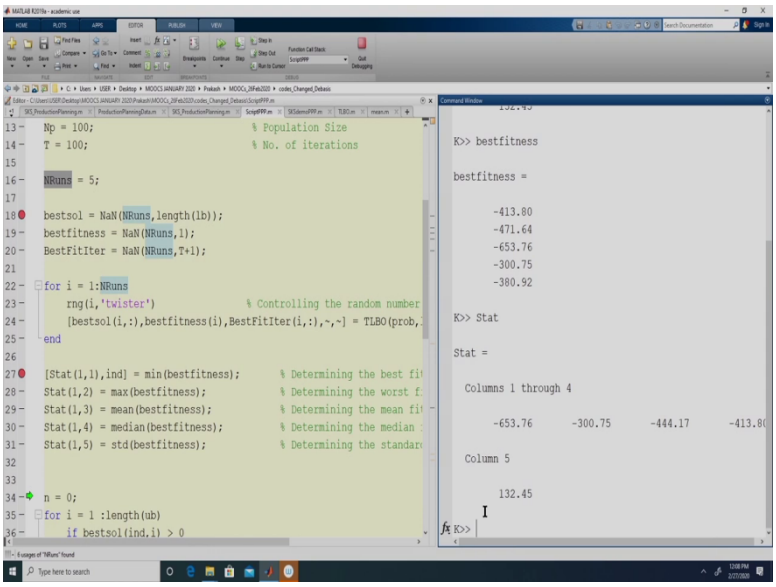
(Refer Slide Time: 67:10)



So, the best run is minus 653.76, the worst is minus 300.75 the mean of the 5 runs is minus 444.17, the median of the 4 runs is minus 413.8 and the standard deviation is 132.45.

So, here you can see the standard deviation can be significantly higher right or even best fitness the difference is quite significant right. We can get as low as minus 300 and as high as minus 653.76. So, if we increase the number of runs to let us say from 5 to let us say 25, we may even get a better solution than minus 653.76 right. So, that is why it is important to run multiple times.

(Refer Slide Time: 67:56)



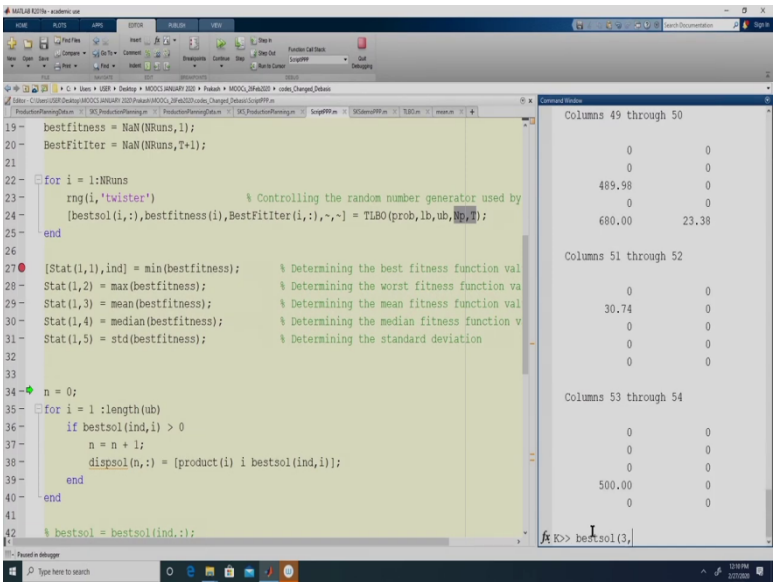
So, this is the stat variable, it gives the statistical analysis of all the 5 runs right. So, here we have plugged it with TLBO, you can use any of the other 4 meta heuristic techniques that we have discussed. Nothing would change right, you need to just call the appropriate function and give the appropriate input to the materialistic techniques right.

So, these 3 input is going to remain constant because that is coming from the optimization problem which we are solving. So, these 2 would also remain the same for all the 4 meta heuristic techniques which we have discussed right. But if you use simulated annealing or something else you might have to give a different value over here otherwise the rest of the parameters, you need to appropriately give.

So, for example, if you are choosing differential evolution, then you will also have to give the crossover probability and the scaling factor. If you are choosing to solve it with particle swarm

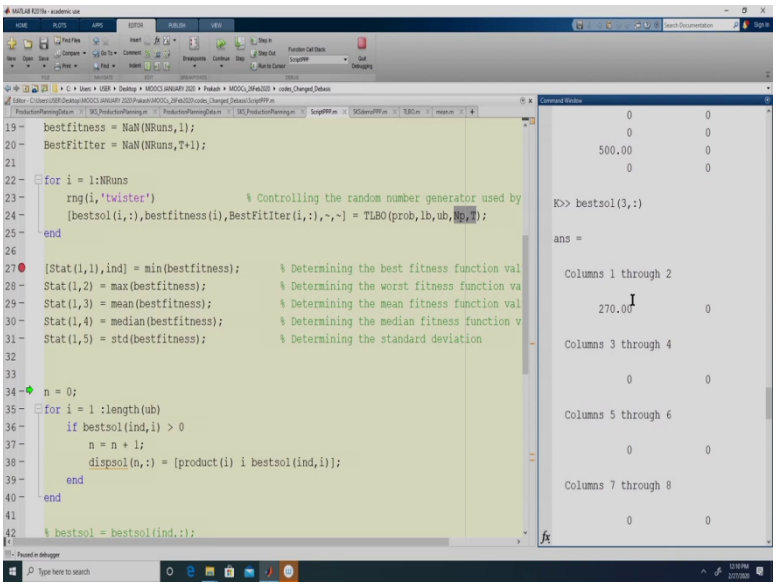
optimization in addition to N p and T you will have to give the inertia weight and the to acceleration coefficient c 1 and c 2 right. So, the rest of the discussion is going to just involve on post optimality analysis right.

(Refer Slide Time: 69:00)



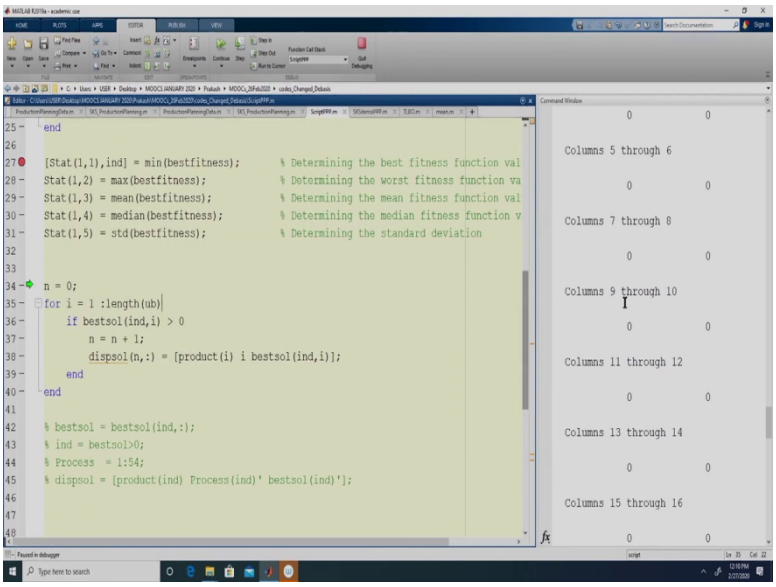
So, what we are doing over here is this best solution if we see it contains; so, the third run is what we are interested right.

(Refer Slide Time: 69:09)



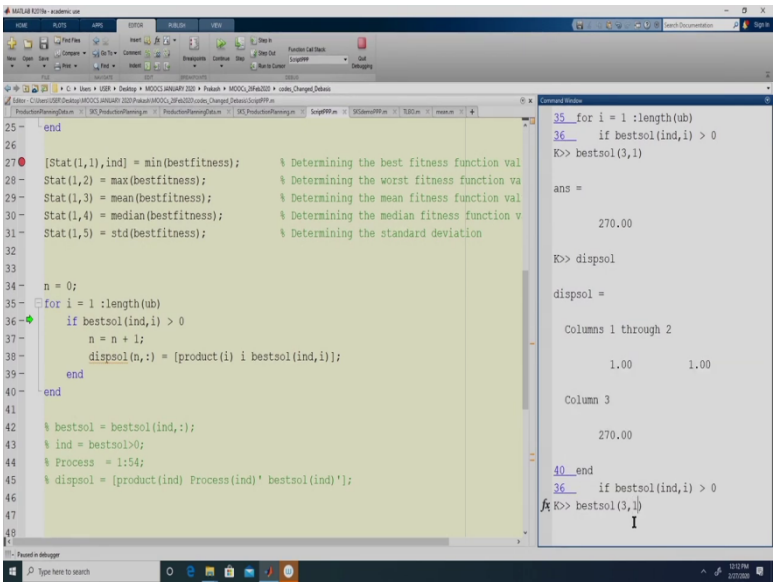
So, right now it is displaying all the 0, let us say we want to display only the non zero values right along with the product as well as the process number right.

(Refer Slide Time: 69:21)



So, then we can write a small piece of code right which is what we have done over here right.

(Refer Slide Time: 69:24)



So, what we are doing is we are initializing scalar n is equal to 0 right. So, this scalar is going to keep track of the solutions which are non zero right. So, and we run a loop across the decision variables right.

So, we have 54 decision variables. So, this length of this lb would be 54. So, we are just going one variable by one variable right and we know that the best solution is in this row, ind right. So, in this variable bestsol, it will contain 5 cross 4 we are interested in the third row right and we are checking for the i th variable. So, if it is greater than 0, we are increasing the counter of n by 1 and we are creating this dispsol right a matrix dispsol.

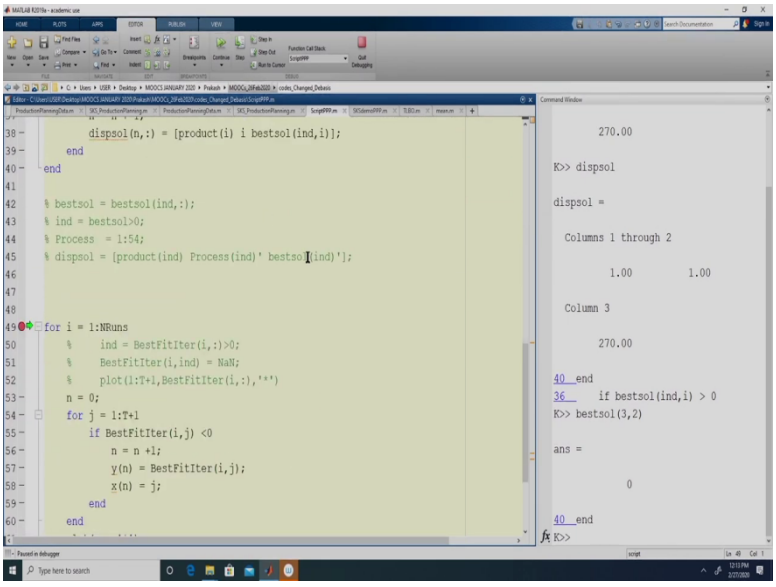
So, this is just to display the solution right. So, what we are doing here is we are stacking the product number right. This variable product contains which process produces which product right. So, we are saying product of i right what is the ith value in this vector product and the

value i right the value i because the location i indicates the process number that is how we have defined our decision variable and this bestsol of ind comma i will give us the value the exact value right.

So, for example, now i is 1 right. So, if we say what is best sol of 3 comma one right. So, bestsol of 3 comma 1 is 270 right. So, it will put 270 over here right and it will come inside this loop only if this value is non zero right. If it had been a 0 value, it would not come inside this if condition, it will not satisfy this if condition and it will not come over here right. So, this we run across all the decision variable right. So, dispsol if we see now, it says that the first product is produced from process 1 and the quantity produced is 270 right.

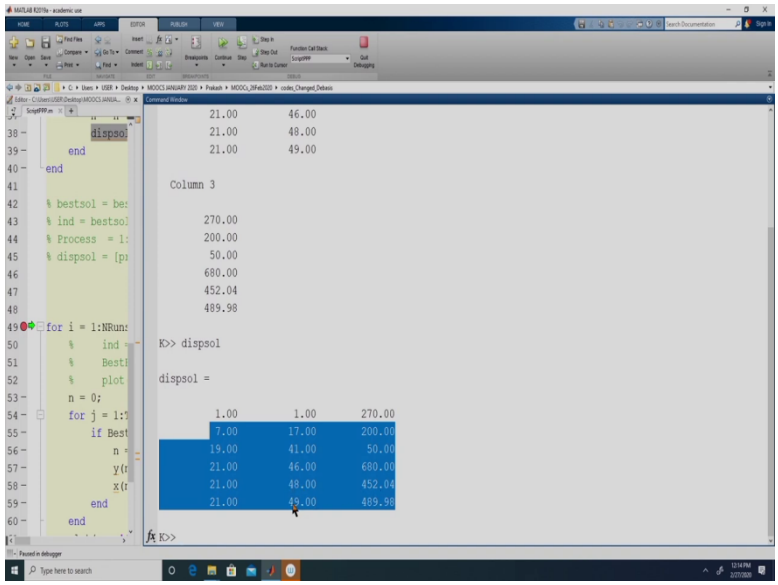
So, this has to be run for all the processes. So, second time this value is actually 0 right. So, best sol 3 of 2 would be 0 right.

(Refer Slide Time: 71:26)



So, it will not get into this if condition. So, n still remains one though we have checked 2 variable only the first variable went through this if condition because the production was greater than 0, this loop gets executed for 54 times right. So, let me just put over here and then execute it right.

(Refer Slide Time: 71:47)

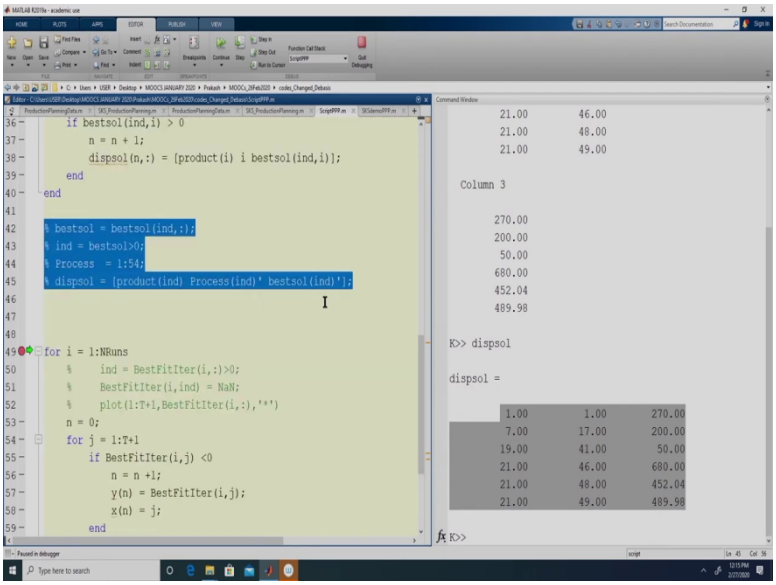


So, now we can just have a look at this dispsol right. So, instead of trying to figure out which are the processes producing manually, we have just made use of couple of lines of code to come up with this non zero values.

So, it tells us that the first product is produced from process one and the amount of production is 270 right we produce product 7 from process 17 and the quantity of production is 200. We produced product 19 from process 41 and the production quantity is 50 and

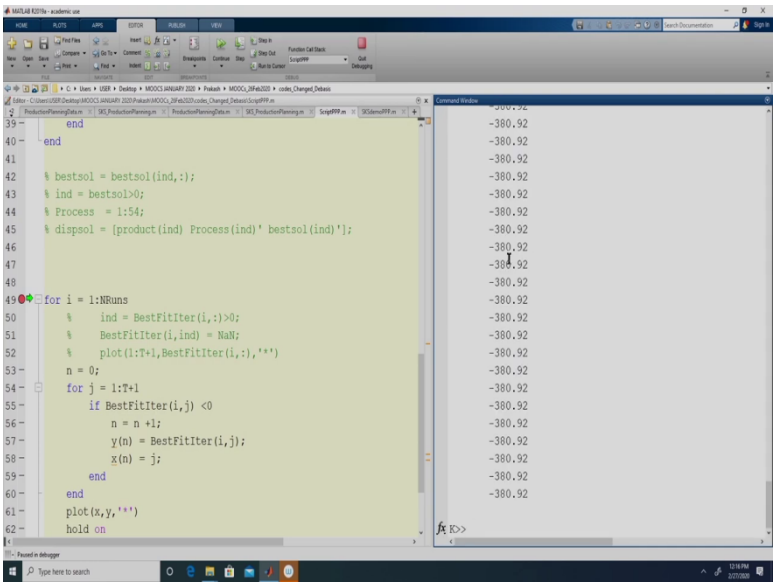
product 21 is produced from process 46, 48 and 49. The production from process 46 is 680, from process 48 is 452 and from 49 is 489.98 right.

(Refer Slide Time: 72:37)



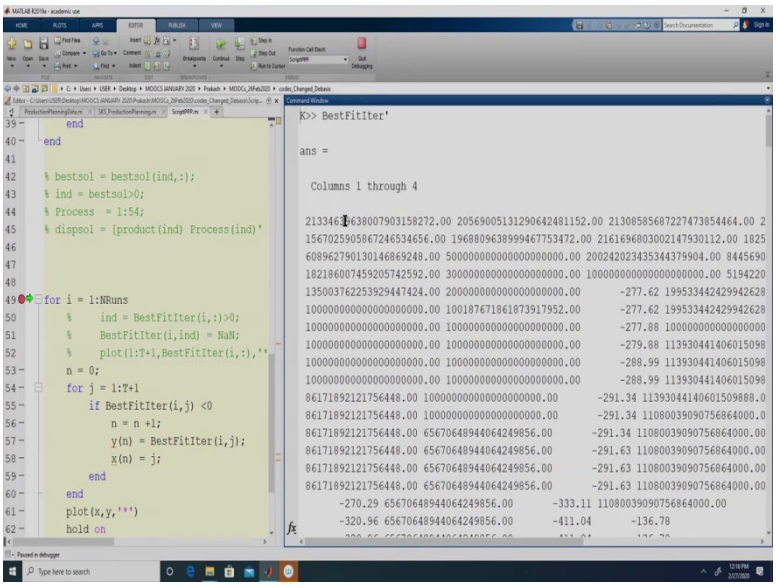
So, now we do not have to manually see what are the non zero values right. So, this piece of code can help us to actually get the non zero values right. So, we have explained you this you can also directly use conditional indexing right and also get the same values right. Now that we have analyzed this best solution now let us try to get all the convergence curve on the same plot right. So, since we had solve this problem 5 times right, we can also plot all the 5 convergence curve and now we are not interested to plot the entire convergence curve right; we are only interested to plot the convergence curve from the point it discovers a feasible solution right.

(Refer Slide Time: 73:14)

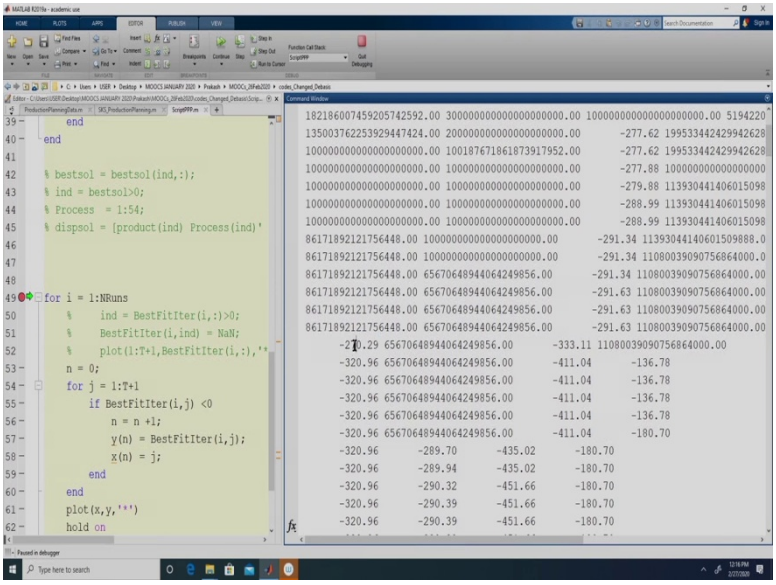


So, only when this BestFitIter let us say I make a column vector of it right.

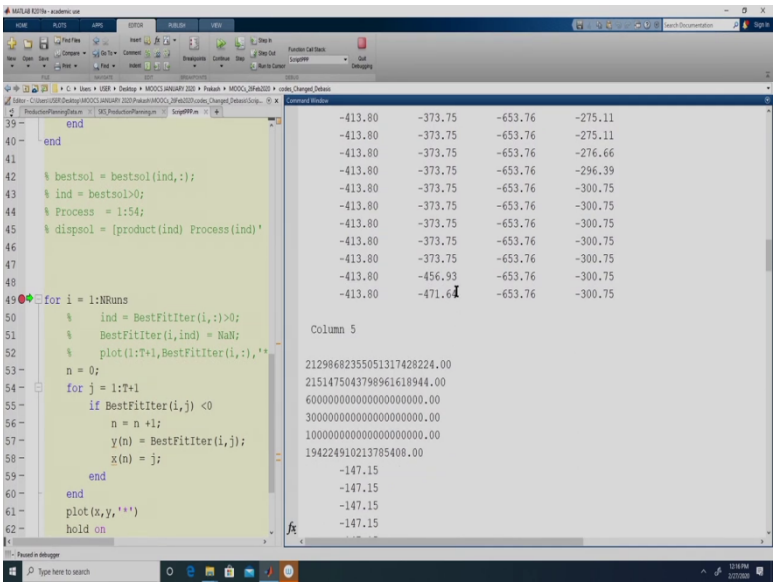
(Refer Slide Time: 73:20)



(Refer Slide Time: 73:23)

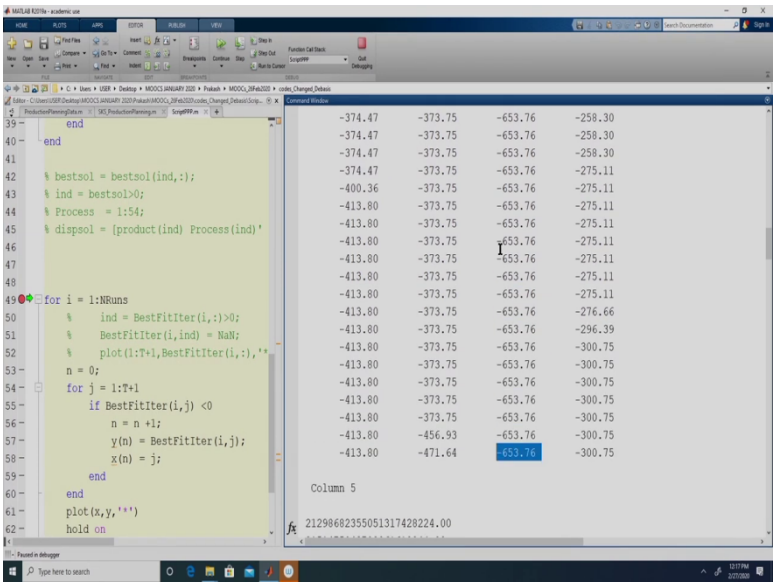


(Refer Slide Time: 73:37)



So, for example, the first column is the first run and now we want to plot only after it discovers a feasible solution that is we want to plot only after this minus 270.29 right and see how it is converging. So, minus 270.29 and all the way up to minus 413.80, for the second run we want to plot from this minus 289.7 and all the way up to minus 471.6. For the third run, we want to start with minus 435.02 and all the way up to this minus 653.

(Refer Slide Time: 73:55)



So, we want to plot only after the algorithm is able to get a feasible solution we are doing that because we want to actually see this dynamics right. If we plot the entire column, then you had seen that we are not able to see what is happening towards the end right. So, for this we have a small piece of code over here right. So, either you can choose to use this section right or you can come and this and just use this section. So, again similar to this we are having a counter n is equal to 0 right and we check for each value of BestFitIter right. So, BestFitIter is going to have j columns right. So, we start from j is equal to 1 and go up to T plus 1 right; T plus 1 because we have in addition to T iterations. We also have the initial population right.

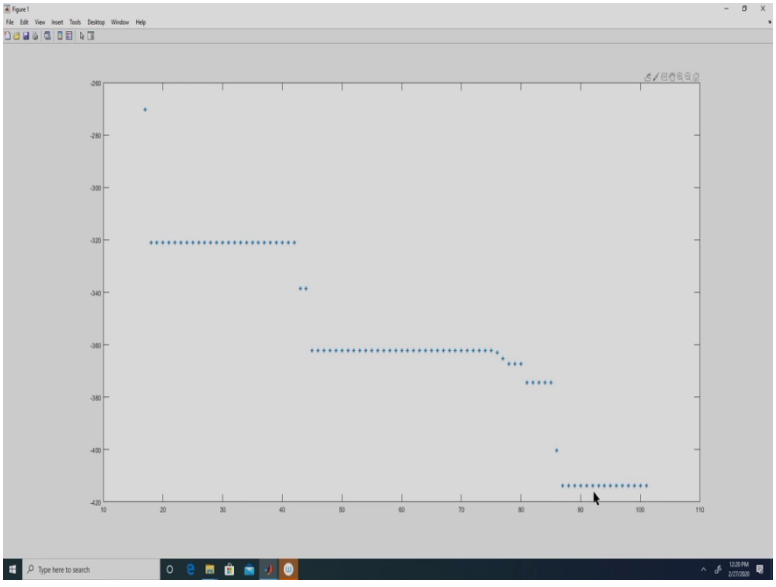
So, if it is less than 0 right so, it will come inside this loop right. If it is not less than 0, it will not come into this loop right. So, right now for the first time it did not enter this condition

right, similarly second time it did not enter, third time it did not enter, fourth time it did not, fifth time it did not right, sixth time it did not right. So, j value currently is 7.

Similarly j is equal to 8 it did not, 9 it did not 10, it did not; let us just keep moving right. So, j is equal to 17. So, when j is equal 17, we get this value to be less than 0 right. So, we increase the counter of n by 1 right and say y of n is equal to whatever value we have in this case it is minus 270.29. So, here if we see minus 270.29 was the first solution right and n is currently 1 right and this occurs at 17 th value right. So, that is why j is also 17 right.

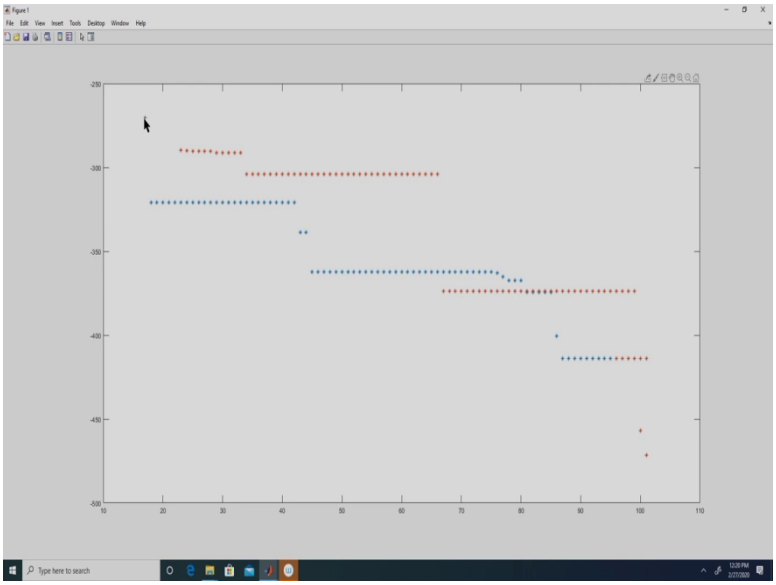
So, in this case we keep track of what is the value and we also keep track of the x. So, this will run for all the iterations for all the iterations T and also the initial population right. So, let me just put a breakpoint over here and then continue right.

(Refer Slide Time: 76:05)



So, this is the convergence curve for the first run. So, x axis is the number of iteration right, y axis is the best fitness function value obtained in that iteration. So, here now we can look at this and analyze whether the algorithm has really converged or additional number of iterations are required.

(Refer Slide Time: 76:30)



So, this if we continue right so, every time it will plot the convergence curve. So, the for the first time, we got a feasible solution in 17, for the second run we got a feasible solution in 23 and the value of minus 289.7 right.

(Refer Slide Time: 76:48)

So, let us just come over here. So, now, we have all the convergence curve. So, that completes this session. In this session what we have done is we have implemented the fitness function file for the production planning problem and then we independently checked it before deploying it with any optimization algorithm.

When we were reasonably sure that the fitness function file is doing things correctly, we plugged it with the optimization algorithm right and then we executed multiple runs on it and then did a statistical analysis right. So, after statistical analysis we just did some post optimality analysis wherein we looked at the convergence curve and we also saw how to plot the a convergence curve. With that will conclude this session.

Thank you.