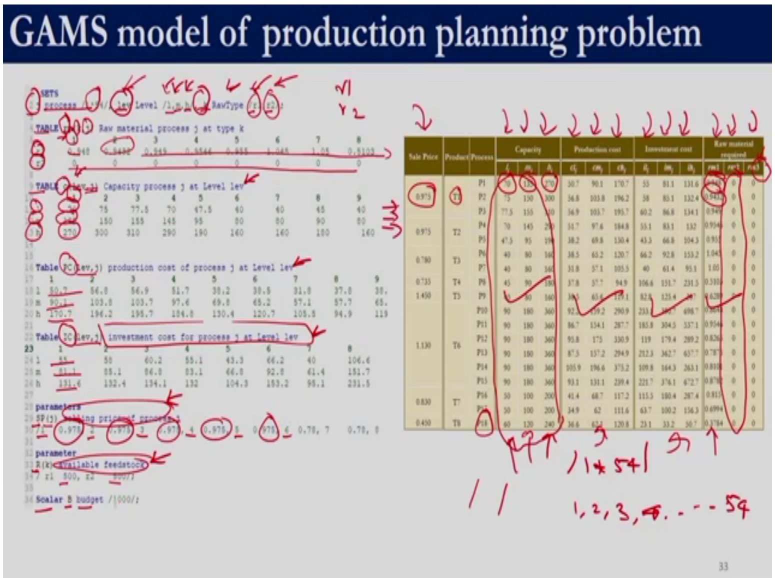
**Computer Aided Applied Single Objective Optimization**
**Dr. Prakash Kotecha**
**Department of Chemical Engineering**
**Indian Institute of Technology, Guwahati**

**Lecture – 35**
**Solution of Production Planning Problem using GAMS**

(Refer Slide Time: 00:31)




Welcome, so now, we look into how to use GAMS to solve the mixed integer linear programming problem formulation which we had developed for production planning problem. So, if you remember the production planning problem, we had this data of selling price and the capacity l, m and h. We had the production costs corresponding to this capacity levels c l, c m and c h.

Similarly, for investment costs we had i l, i m, i h and for raw material, we have three types of raw material right; raw material 1, raw material 2, and raw material 3. Here we have shown you only 18 process. But if you remember there were 54 processes, which can be used to produce 24 products right. For the case study which we had taken, there was no limit on raw material 3 right.

So, it was assumed that raw material 3 is available in abundant right, so there is no point of including a constraint on raw material 3. So, we will only use raw material 1 and raw material 2. So, first we need to code this data into GAMS right. Once we have got the data, we will then subsequently include the model equations right.

So, what we are doing here is we are defining 3 sets; process 1, process 2 process 3 or the name of the process or we can just have numeric values right. So, what we are doing is we are writing 1 star 54 right. So, this is equivalent to writing 1 comma 2 comma 3 comma 4 all the way up to 54 right. So, compactly it can be written as 1 star 54 within those slash.

So, remember whatever we give within slash corresponds to the value of that particular set right. So now, set j has 54 elements; 1,2,3,4,5,6 all the way up to 54. The set lev has 3 elements l m and h right, the set k right has 2 elements r 1 and r 2. Now, we have defined a set for the processes the levels and the type of raw material right. So, first let us get the detail of the raw material so, we define a table right, we define a variable rm right. So, rm runs on the set k comma j.

So, k comma j means k is for the raw materials and j is for the process right. So, it will be 2 rows because we have r 1 and r 2 and it will have 54 columns from 1,2,3,4,5,6,7,8 all the way up to 54 right. So, here we have r 1 r 2 because we have used r 1 and r 2 over here; the values are r 1 and r 2. So, we write the same values r 1 r 2 over here and over here we write the process number 1,2,3,4,5,6 all the way up to 54. That is how we have defined our set j to indicate the process right.

So, we get this value 0.948 so, that is process 1 raw material 1. Process 2 raw material 1 is 0.9432. Similarly, we enter this entire column over here right. Here we have shown you only 8 values, but all the 54 are to be entered. Similarly, we enter the values for the second raw material. So, for the 2second raw material, the first 18 processes do not use it right. So, that is why you see 0's over here.

So, if you go back and look at the data which we have given you, the subsequent processes do utilize raw material 2 right. So, those values would be entered appropriately then we want to enter these values right; the capacity values. So, we define a table right with a variable name c right to denote capacity and it is going to have 3 rows and 54 columns; 3 rows because, we have 3 elements in the set lev, this corresponds to this levels 3 levels and 54 process right.

So, over here the set used for indicating the levels l m h are lev so, we write lev over here and the set used for denoting the process are in the set j. So, we have j over here righ. So, and then we enter the values right, so, l m h are the values of the set lev right; so, l m h and those are in the rows. So, for example, level 1 process 1 is 70. So, process 1 indicated by this 1, level l is 70. Level m it is 135 so, for m process 1, it is 135; for process 1 h is 270 so, we write 270 over here right.

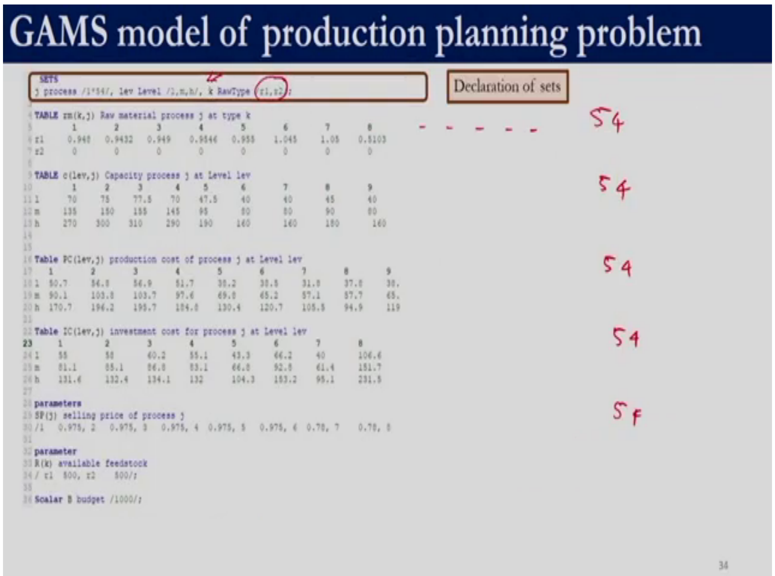
So, what we see over here as column vectors have become row vectors. So, similarly we defined two additional tables PC and IC. Their structure is similar to this capacity right lev comma j and we appropriately enter the production cost for the 3 levels and the investment cost for the 3 levels right. So, we have got this data, we have got this data, we have got this data, we have got these two raw materials raw material 3 we are not including because, there is no constraint on it.

Now the only thing that we need to include is the selling price. So, here if you remember the selling price of product T 1; no matter from which process it is produced it is 0.975 right. So, that is what we are doing now. We are defining a parameter SP, the name of the parameter is SP. It runs over the set j right and j indicates the process. So, j contains the values of 1, 2, 3 all the way up to 54. So, we have 1, 2, 3, 4, 5, 6 all the way up to 54 and then we enter the corresponding values of selling price.

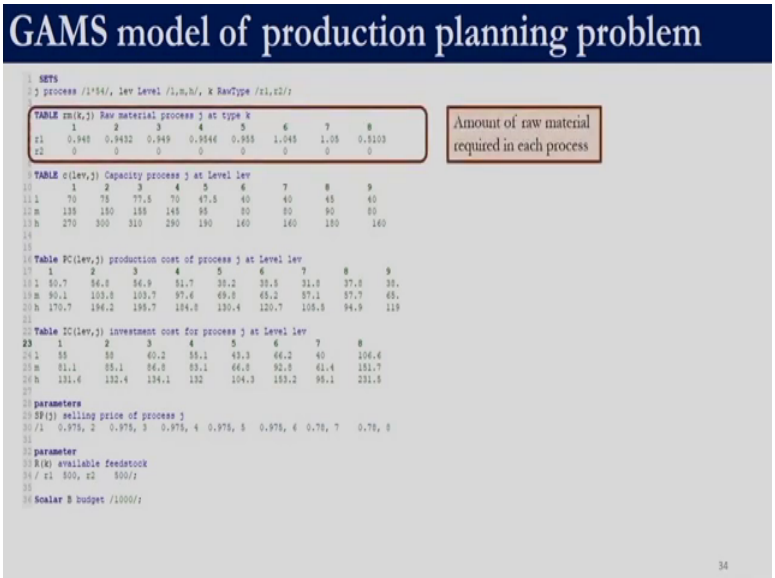
We have one more parameter on the amount of raw materials which are available. So, we have two types of raw material. So, we define a parameter R right. It runs over the set k right and set k contains these two values r 1 r 2. So, the value of raw material 1 that is available is 500 and the value of raw material 2 that is available is also 500 right. So, all these which you see over here are description of the appropriate variable.

We also define a scalar B to denote the budget right and the amount of budget that is initially available is 1000. So, we supply that using the syntax of GAMS. So, just to quickly run you through what we have done so far is; we declared 3 sets and we assigned values to them. Then we created a table to provide the amount of raw material which is required in each process. We have another table which gives the production capacity for each production level, and then we gave the production costs and the investment costs corresponding to each process and the appropriate level.

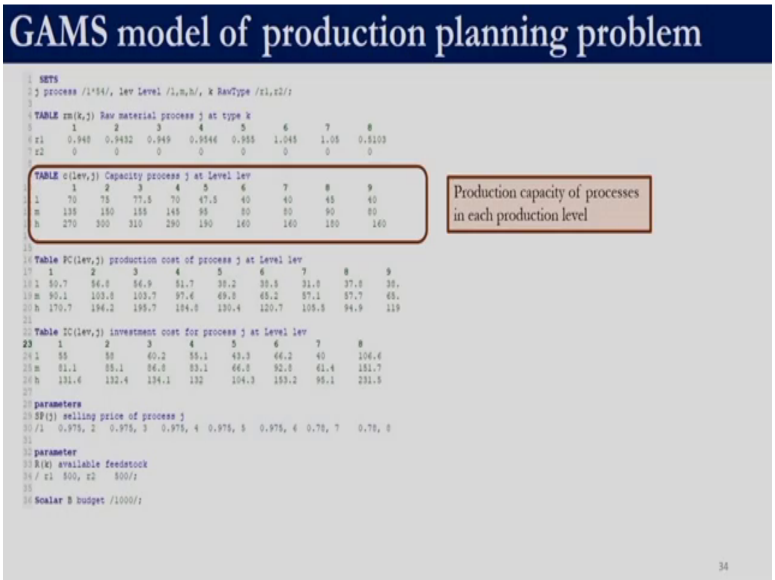
(Refer Slide Time: 06:36)



(Refer Slide Time: 07:43)




(Refer Slide Time: 06:47)



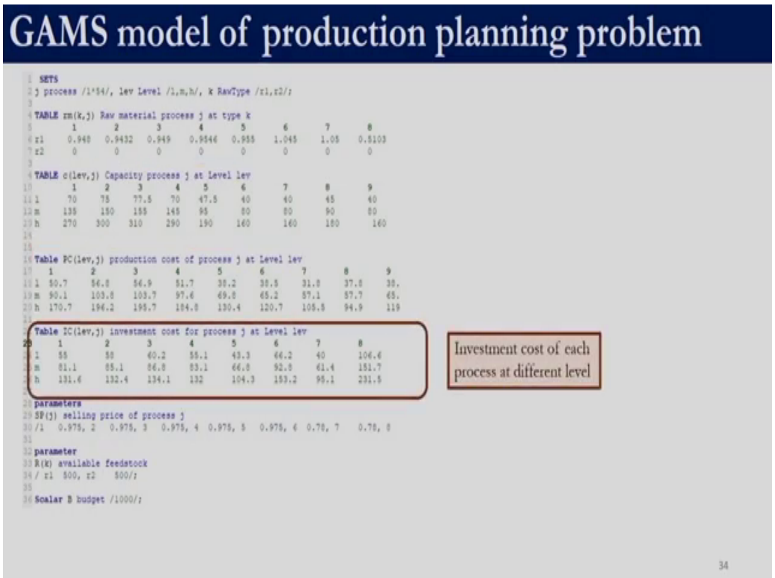

(Refer Slide Time: 06:53)

# GAMS model of production planning problem

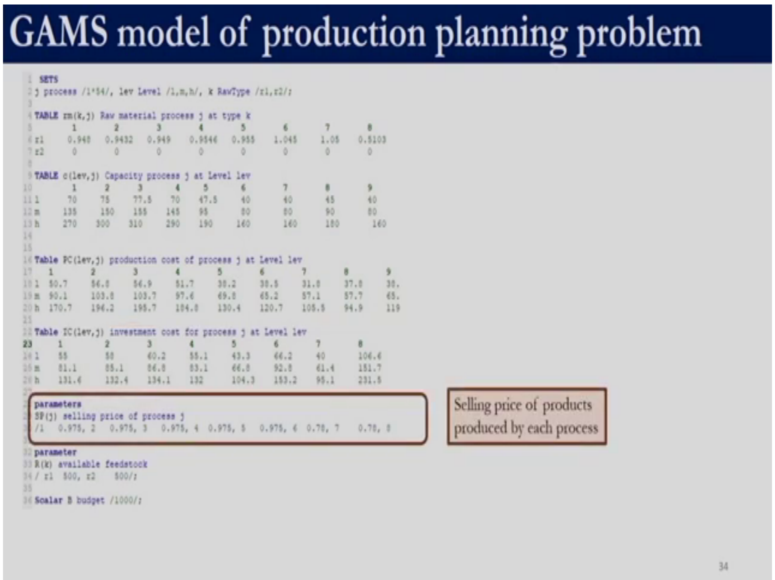
```
 SETS
j process /1*54/, lev Level /l,m,h/, k RawType /r1,r2/;

TABLE rm(k,j) Raw material process j at type k
         1        2        3        4        5        6        7        8
r1     0.948   0.9432   0.949   0.9546   0.955    1.045    1.05    0.5103
r2       0        0        0        0        0        0        0        0

TABLE c(lev,j) Capacity process j at Level lev
         1      2      3      4      5      6        7      8      9
l       70     75    77.5    70    47.5    40       40     45     40
m      135    150    155    145    95      80       80     90     80
h      270    300    310    290    190    160      160    180     160


Table PC(lev,j) production cost of process j at Level lev
   1        2        3        4        5        6        7        8        9
l  50.7     56.8     56.9     51.7     38.2     38.5     31.8     37.8     38.
m  90.1     103.8    103.7    97.6     69.8     65.2     57.1     57.7     65.
h  170.7    196.2    195.7    184.8    130.4    120.7    105.5    94.9     119

Table IC(lev,j) investment cost for process j at Level lev
     1          2         3         4         5         6         7         8
l    55         58        40.2      55.1      43.3      66.2      40        106.6
m    81.1       85.1      86.8      83.1      66.8      92.8      61.4      151.7
h    131.6      132.4     134.1     132       104.3     153.2     95.1      231.5

parameters
SP(j) selling price of process j
/1   0.975, 2   0.975, 3   0.975, 4  0.975, 5   0.975, 6  0.78, 7   0.78, 8

parameter
R(k) available feedstock
/ r1  500, r2   500/;

Scalar B budget /1000/;
```

Production cost of each process at different level

(Refer Slide Time: 06:56)

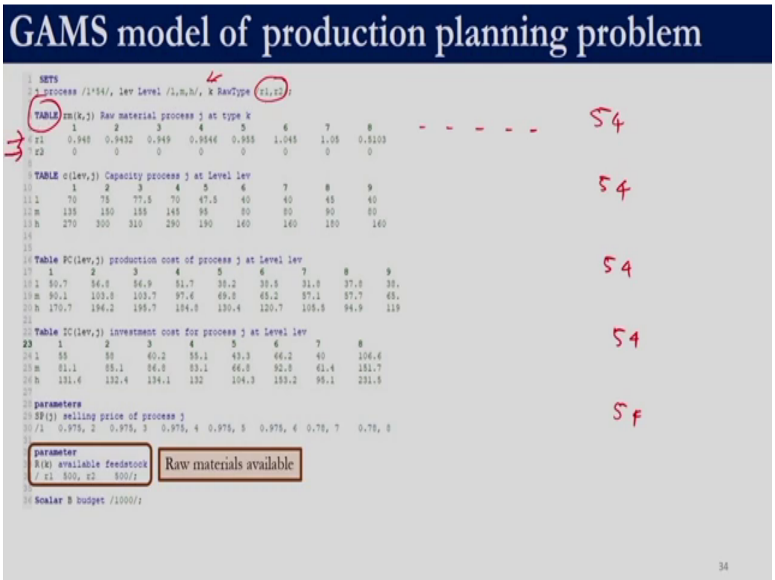



(Refer Slide Time: 07:00)

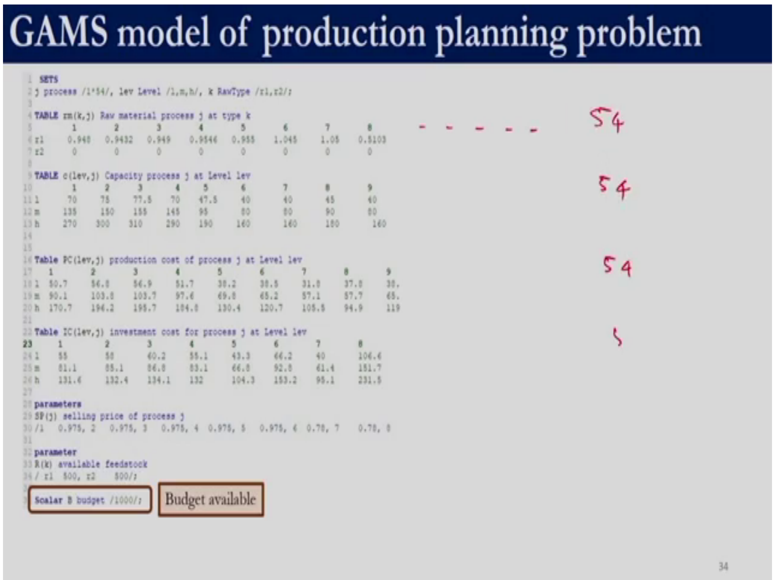



And we have this vector SP which denotes the selling price of the product and we have another vector over here which has the amount of raw materials which are available. Finally, we have the scalar which shows the amount of budget that is available right. So, as of now what we have done is we have included only the data into the GAMS model file right. Again remember here we are showing you only a cropped version; all of this runs all the way up to the 54 processes which we have.

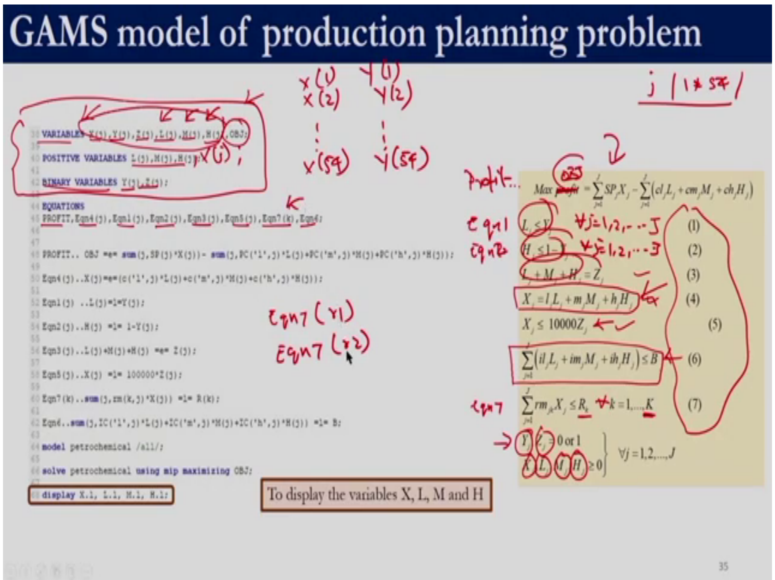
(Refer Slide Time: 07:04)




(Refer Slide Time: 07:10)




(Refer Slide Time: 07:28)




Now, let us define the variables which we have. So, this is the mathematical formulation which we had developed right. So, here if we see, we have 6 types of variable Y j, Z j, X j, L j, M j and H j right. So, that is what we are defining variables with the keyword variables X of j, Y of j, Z of j, L of j, M of j and H of j right; and it runs across the set j and we know that the set j we have defined it as 1 star 54.

Since, we have defined the set j to have the values 1 to 54, defining the variables over here means X of 1 X of 2 all the way up to X of 54. Similarly, Y of 1 Y of 2 all the way up to Y of 54. So, this holds true for all these 6 variables right. And if you recollect our previous discussion, we need to define a variable for the objective function right. So, in this case what we are doing is we are defining the variable to be OBJ, we will use the variable OBJ to determine the objective function value right.

So, among these variables which we have over here right L j, M j H j are positive variables right. Once we say L j, M j, H j are positive variables in view of this equation right, X j will also be positive or we can even give comma X of j over here and then a semicolon wherein we define all the positive variables. Y j and Z j are binary variables which we have right. So, we have this key word binary variables and have Y of j comma Z of j and then ended with a semicolon.

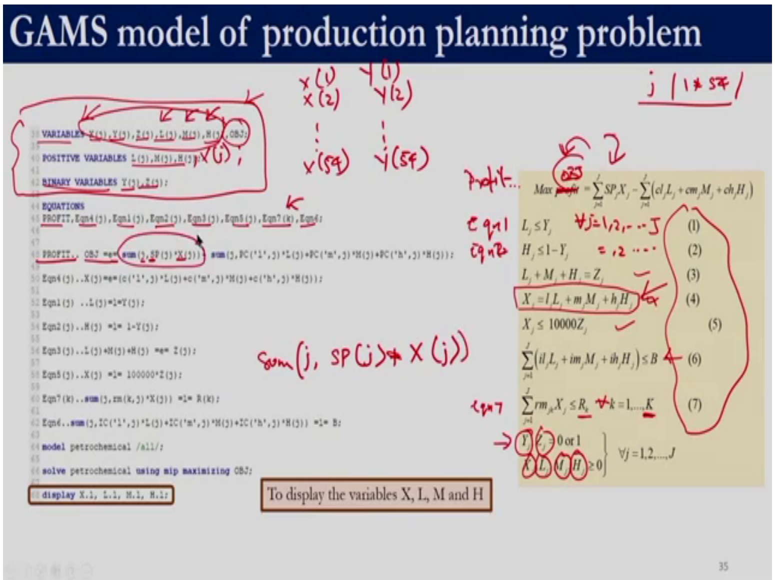
So, in this block we have defined all the variables and their types right. Now we need to define the equations so, what we will do over here is we will take the same equation numbers which are given over here. So, this equation we will call it as E q n 1, E q n 2; E q n 2 and all the way up to equation 7 right. And all of this equation if you remember, it is to be written for all the processes right 1, 2 all the way up to J.

So, this is true for equation 3 equation 4 and equation 5 whereas, this will be only a single constraint because there is only 1 investment costs right. Similarly, this equation will come twice right or k times. In this case, we have two raw materials and the objective function we define it with a equation name profit. So, the name of this equation is profit, the variable is OBJ right.

So, the name of the equation which we will use for coding the objective function is profit and then we have all the equations right. So, equation 1 is this one which runs across J equation 2, 3, 4, 5 run across j right. So, this is equation 2, 3, 4 and 5. So, these equations run across the set j. Equation 6 is a single equation right. So, we do not write equation 6 within brackets anything right equation 7 is for the k types of raw material.

So, over here we have defined the set k to contain two values right. So, when we write equation 7 of k it basically means, we will have two equation; equation 7 of r 1 and equation 7 of r 2 because this set k contains values r 1 and r 2. So, now that we have defined the data which is required, we have defined the variables and their types. We have defined the equations which we are going to use. Now we need to define the equations themselves right.

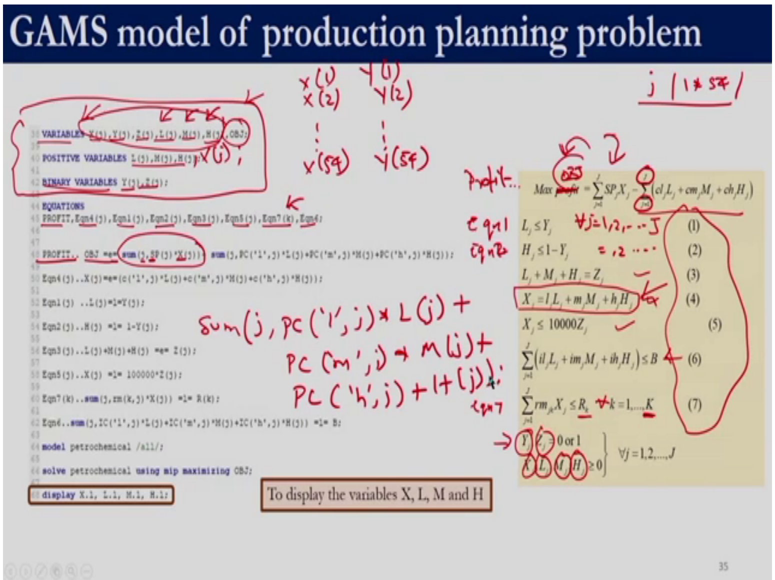
(Refer Slide Time: 11:12)



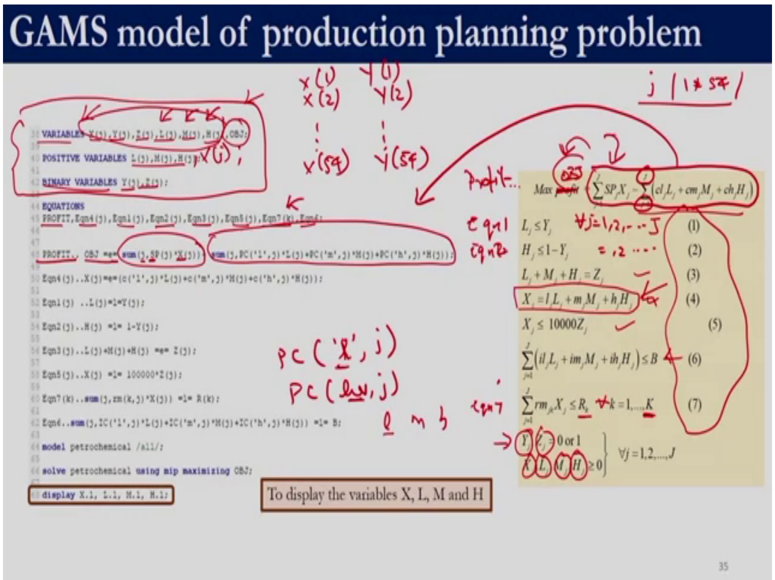

So, over here we say name of the equation PROFIT dot dot and then OBJ equal to e right because the objective function is now written as an equality constraint. So, the value of the objective function is carried over to OBJ. Over here if we see this particular equation right, it is summation of SP and X j right. So, summation of the variable SP and X right over the set j right so, that is why we write sum over j right the variable SP and the variable X.

So, that is this particular term over here right. Then we have this second term which we need to write. So, for that what we would do is sum, we need to sum over the index j right. So, sum over the index j c l is indicated by PC over here right; so, PC of l right. So, PC of the set l comma j for the j'th process multiplied by l of j plus PC of m right comma j into M of j plus PC of h comma j into H of j right. So, that is what we have written over here.

(Refer Slide Time: 11:56)

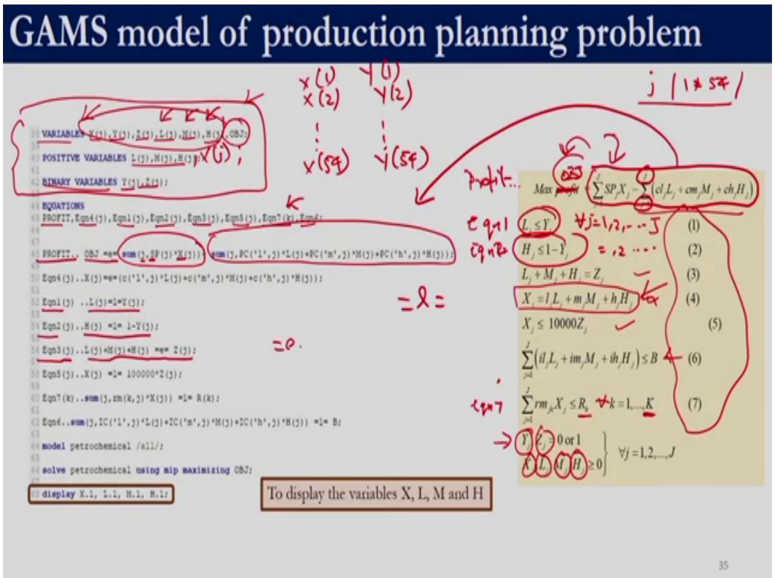



(Refer Slide Time: 12:46)



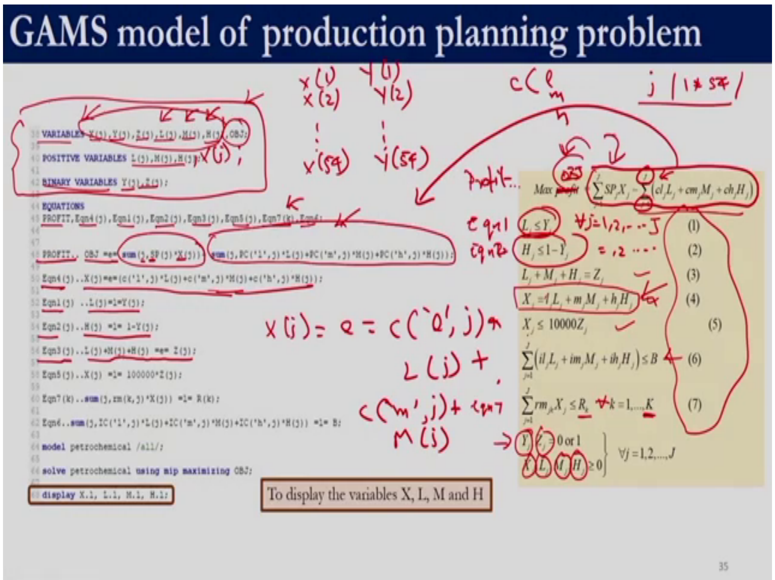

So, over here if you see whatever we have written in the typical mathematical form over here can be directly coded in GAMS. So, again here we have used PC of l comma j and not PC of lev comma j. So, PC of lev comma j if you write, it will access all the three elements which is not what we want over here, we only want the production costs corresponding to the low level right, and the set lev contains the values l m and h. If we want to access a particular element from that we can directly write that element within single quotes. That is how we have written the second part of the objective function.

(Refer Slide Time: 13:29)




So, then equation 1 right is L of j is less than equal to Y of j which is straightforward over here right equal to l equal to is for less than equal to constraint. Similarly, equation 2 is H of j is less than or equal to 1 minus Y of j. Equation 3 of j right is L of j plus M of j plus H of j is equal to e equal to right. So, equation 4 is over here right. So, here again similar to this c l j we need to access L j right.

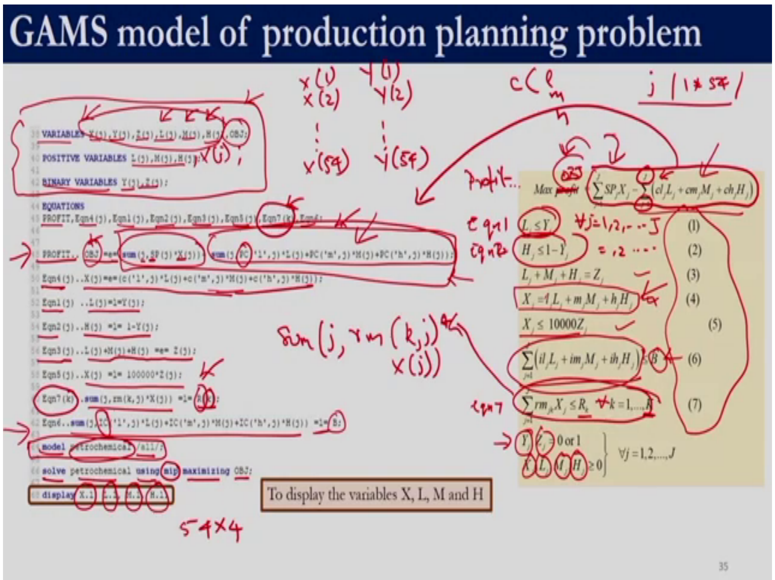
(Refer Slide Time: 14:03)




So, what we are doing is X of j is equal to e equal to; so, if you remember L M and H, we have saved in the table c and we have used l m h over here right. So, what we do is C of l within single quotes comma j star L of j plus C of m comma j plus M of j and similarly, we write for the last term also right. So, this equation is similar to how we constructed the second part of the objective function.

Equation 5 again is straightforward so, equation 6 over here is similar to how we wrote the second part of production cost. So, this is similar to this part over here except that instead of PC which denotes production costs, we have defined the investment costs as IC right. So, if you compare this second part of this profit equation and this equation 6, you will see that it is similar. So, this is the investment cost which will be required for whatever we decide to produce and the amount that is available is B right.

(Refer Slide Time: 14:53)




So, this B we have defined it as a scalar initially. So, this should be less than equal to the budget right;so, similarly the 7th constraint over here right. So, now this constraint is not to be written over j it is need to be written for all types of raw material right; over here also we had defined equation 7 of k. So, here we have equation 7 of k and then over here we have summing over the j th raw material so, sum j comma r m which is the name of the table of k comma j.

So, if you look into this table right. So, this k was the type of raw material so, that was in the row and the processes are in the column right. So, that is why we write r m of k comma j multiplied by X of j so, that has to be equal to capital R of k right. So, capital R is a parameter which will contain two values because it runs over the set k right, the set k has values of r 1 and r 2.

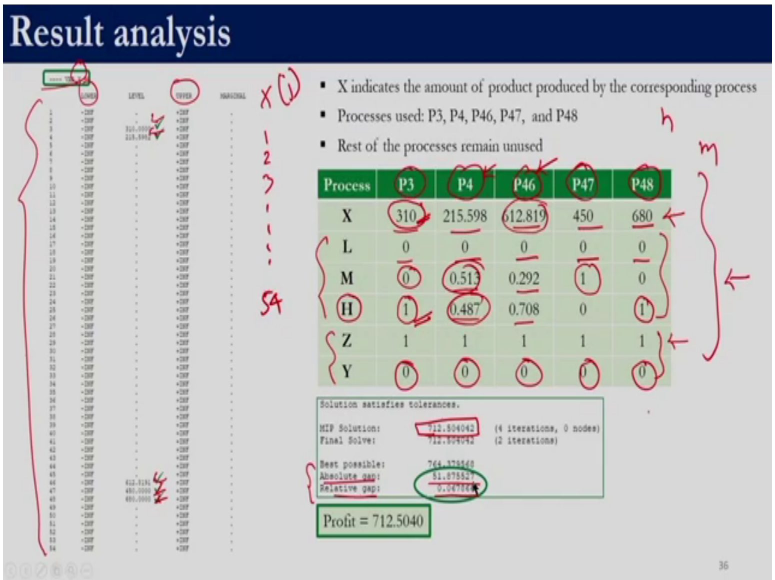
So, now we have coded all the equations which we have developed in GAMS. Now we need to construct the model so, we use the key word model right, and name of the model that we are giving is petrochemical right. And we want to include all the constraints which we have written over here so, within two slashes we write the word a double l and end it with a semicolon.

So, this line over here will help us to construct the model. Once we have constructed the model. In this case, we know it is a mixed integer linear programming. So, in GAMS as we discussed earlier mixed integer linear programming is denoted by m i p right. So, here we have solved the name of the model petrochemical using the default solver of mixed integer programming while, maximizing the variable OBJ. So, when we maximize the variable OBJ we are basically, maximizing this expression over here which is nothing, but the PROFIT right.

So, we also want GAMS to display the final values of the decision variable X, the decision variable L, the decision variable M and the decision variable H; it will give all the values, but in the display part we want to display these particular four variables. So, each of this variable will have 54 values because X, L, M and H each of them run over the set j and the set j contains 54 values right. So, here it will display 54 into 4 values L M H and X.

So, now that we have constructed the GAMS model, we can execute it right. When we execute the GAMS model, we will get two files; one is the list file and the log file. You can browse through the list file right. So, over here we will just discuss the values of the decision variables so, this is a section of the list file. So, right now we have shown only the variable x right. So for variable x if we see the lower bound is given, the upper bound is given. We did not explicitly specify X to be a positive variable right.

(Refer Slide Time: 17:53)




So, that is why it is showing the lower bound to be minus infinity and the upper bound is plus infinity right. So, since we have 54 processes right; so if you remember X was running over the set j and the set j had values of 1, 2, 3 all the way up to 54 right. So, that is why we have 54 over here. Out of the 54 processes only process 3, process 4, process 46, process 47, process 48 produce a particular product whereas, the other 49 process do not produce anything. If you execute this program and also look into the L values, M values, H values; you will be able to observe these values.

Similarly, you can also look at the Y and Z values right. So, for P3, P4, P46, P47, P48 the amount produced is given as over here. And the L values are 0 for all of them so, for all this process the production is either at M or at H or in between M and H right. So, for example;

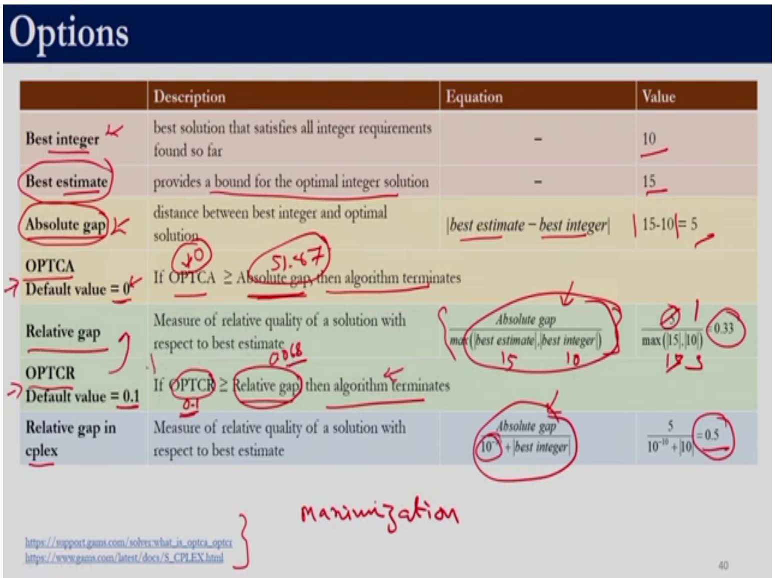
for process 3 if you see even the M value is 0 and the H value is 1 right, remember this H is the uppercase H, not the lowercase H.

If the lowercase h value for process 3 would be 310 right. So, that is why when it produces 310 from process 3, the capital H takes the value of 1. Similarly, for process 48 the amount of production is at the high level. For process 47 the amount of production is at the medium level corresponding to the value m right. For the rest of the two processes, P4 and P 46 it is somewhere in between M and H.

Since, all these processes are used the value of Z would be 1, which is as expected and the value of Y would be 0 because the production is between M and H. So, this is the production plan that we obtained using GAMS right. So, the profit for this production plan is 712.5. So, here if you see we have something called as absolute gap and relative gap right so, here absolute gap is 51.875527 and relative gap is 0.067866.

So, we have got this profit of 712.5040. But here if we see the absolute gap shows a very large number and over here, we also get the statement solution satisfies tolerance right. And that is why the solver terminated right. So, first let us understand what is this absolute gap and relative gap and then we will come back to this model. So, we need to understand two terminologies over here, one is absolute gap and one is relative gap right.

(Refer Slide Time: 20:38)




So, to understand absolute gap, we need to know what is best integer and best estimate right. Since, we have already looked into mixed integer linear programming, you would be able to understand that; best integer is integer solution which has been obtained so far right. So, far in the sense; as far as the branches have been explored; so, let us say while exploring some note, let us say the best integer is 10. And the best estimate is a bound for the optimal integer solution so let us say 15 so this discussion is for a maximization problem.

So, absolute gap is the absolute distance between best estimate and best integer right. So, over here the absolute gap is 15 minus 10 which is 5. So, when we are exploring the search space this absolute gap can be calculated right. So, the default value of the parameter OPTCA is set as 0. If our absolute gap is less than OPTCA, so this is a user defined parameter right the default value is 0.

So, if we get an absolute gap which is less than or equal to OPTCA then the algorithm terminates. In this case our OPTCA is 0, the default value so, the absolute gap which we had was 51.87 right. So, this condition is not satisfied yet the algorithm terminated right. So, in addition to this OPTCA we also have one more option known as OPTCR right to understand this OPTCR, we need to understand relative gap right.

So, relative gap is given by this formula right, the absolute gap which in this case is 5 because that is what we have calculated divided by max of absolute of best estimate comma absolute of best integer. So, best integer in the current case is 10 right and best estimate is 15 right. So, max of absolute of 15 comma 10 so, that is 15 right. So, this value comes out to be 15 so, this is 1 by 3. So, this is how we calculate relative gap right.

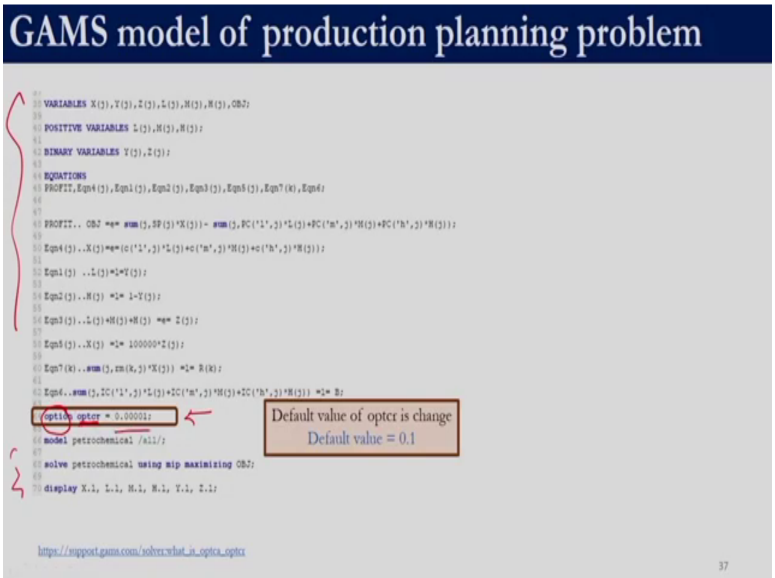
So, the default value of OPTCR is 0.1 so, similar to absolute gap during the search procedure we can also calculate the relative gap right. And if this relative gap which we calculate is less than or equal to OPTCR which is the default value is 0.1 right, then the algorithm would terminate. So, in the current case we had an a relative gap of 0.68 right. So, since this condition is being satisfied the algorithm terminated despite the fact that the absolute gap is a very large value.

So, this calculation of OPTCR varies from solver to solver right in CPLEX, it is not calculated using this formula, but it is calculated using this formula. So, all these details are available in the solver manual as well as GAMS manual right. So, over here it is absolute gap divided by 10 power minus 10 a very small number plus absolute of the best integer solution so, in this case it works out to be 0.5.

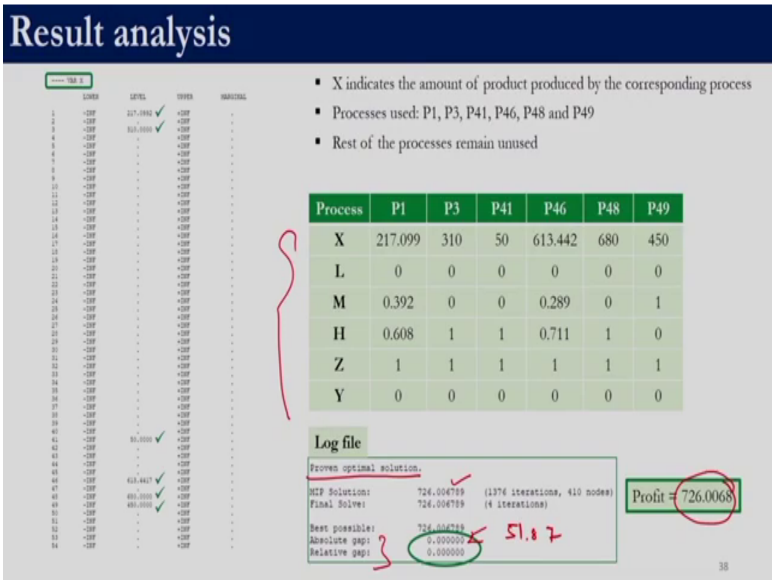
So, depending upon the solver with which we are working the relative gap would be calculated using this formula or this formula right. So, in any case if this condition is satisfied that whatever value of OPTCR is set if it is greater than or equal to relative gap right as and when we achieve a relative gap which is less than or equal to the OPTCR, the algorithm terminates right. So, in this case since this value was 0.1 and this value was 0.068 it terminated right.

So, the reason for termination is this OPTCR right. So, we can decrease the value of OPTCR. So, this is the same code which we have discussed previously except for this line in which we are changing that default value of OPTCR. So, the syntax is the key word option the name of the variable so, in this case we are changing OPTCR is equal to a very small value right.

(Refer Slide Time: 24:08)

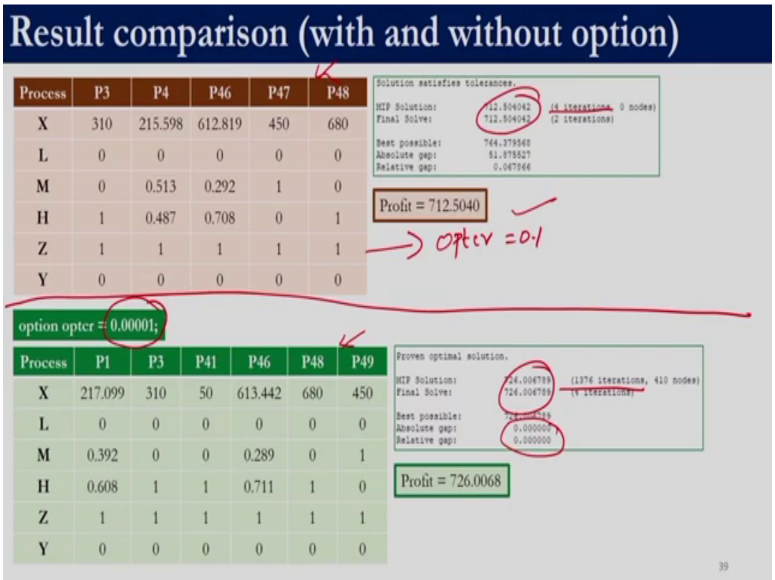



(Refer Slide Time: 24:38)




So, when we solved with this statement, we get a solution of 726.06789 and over here previously we had got a statement which said solution satisfies tolerance here in the log file we get proven optimal solution. And this absolute gap and relative gap if we see previously this was 51.87 now it is 0 the absolute gap is almost 0 and the relative gap is almost 0 right. So, the profit now is 726.0068 right and these are the corresponding decision variables which you can get from the list file.

(Refer Slide Time: 25:01)



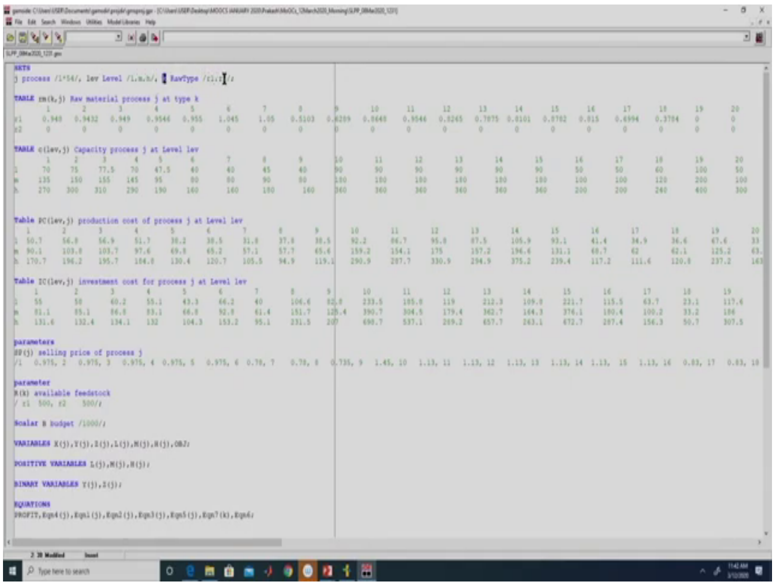

So, this shows the change in decision variables and the objective function value for default options so, in this case the OPTCR was not set; so when it is not set its default value is taken which is 0.1. So, for the second case for the same problem when we change optcr to a very small value we were able to get a better profit and the absolute gap and relative gap are now almost close to 0 right.

So, previously we got the solution in just 4 iterations now it requires 1376 iterations so, that this optcr condition is satisfied right. So, as you can see there can be significant difference so for example, let us say the unit of profit was millions of Dollar right. So, here we get a difference of almost 14 million dollar right. So, this part of the discussion particularly on options we have intentionally included, so that you understand that it is not sufficient to just solve a problem.

We also need to ensure that the algorithm has a termination which is satisfactory to us right so, previously we had seen that the absolute gap is very high right. We should not end our exercise just by determining the objective function value and the decision variable. But we also need to analyse the reason for the termination of the algorithm right and if certain settings need to be revised we need to revise this those settings and solve the problem so, that we make it a better solution.

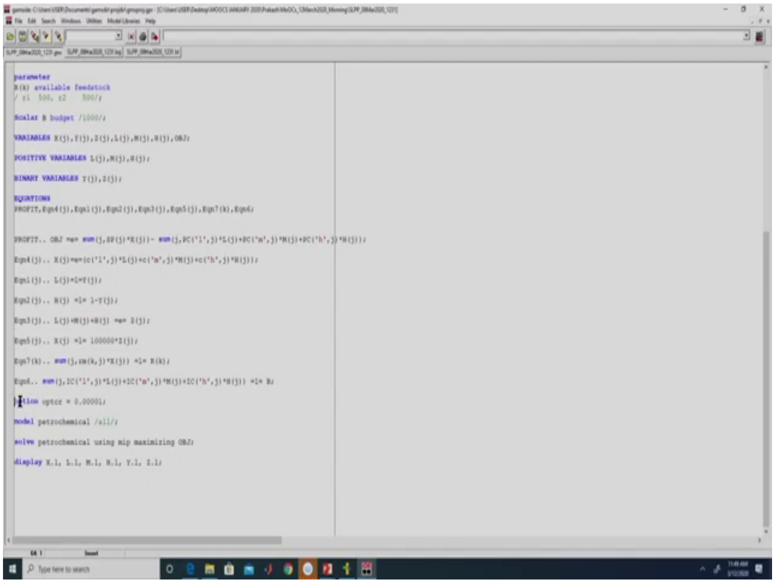
So, additional details on the settings of OPTCA; OPTCR with respect to GAMS as well as with respect to solver can be obtained from here. So, now let us execute the code of production planning problem into you so, we have three sets set j, set lev and set k right. So, set j contains 54 values the values are 1,2,3,4 all the way up to 54, the set lev contains the value l m and h and the set k contains the value r 1 and r 2 right.

(Refer Slide Time: 26:32)



So, over here we are defining all the data which has been given to us, and then we have defined these two parameters right. So, these two parameters are for the raw material 1 and raw material 2 this is the scalar which defines the budget value these are our 6 decision variables right so, all of them run over the set j so, set j contains 54 values. So, this is 54 into 6 decision variables right in addition to that the objective function has to be defined as a variable in GAMS so, that is why we have this additional variable OBJ.

(Refer Slide Time: 26:56)



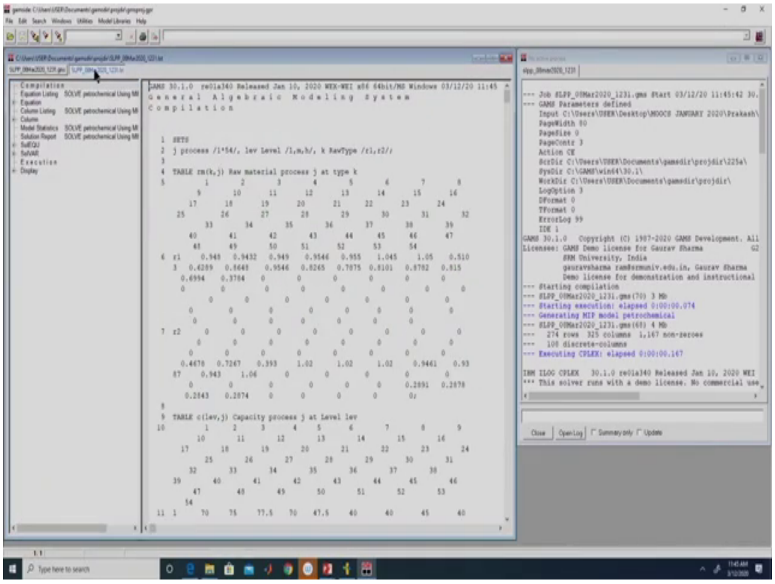

And we have two sets of binary variables so, Y and Z are binary variables so, we defined binary variables and define what are of those variables. Over here, we have 7 types of constraint and then we have the equation for profit right the names of all the constraints are defined over here among this equation 4, equation 1, equation 2, equation 3, equation 5 run over all the process right.

So, those have to be written 54 times and equation 7 has to be returned 2 times because, we have 2 raw materials which are available in limited quantities right. So, when we showed you the data we showed you 3 raw materials, but since there is no constraint on the 3rd raw material. There is no point of including it GAMS file right had there been any constraint involving it we can include it in a manner similar to r 1 and r 2 right.

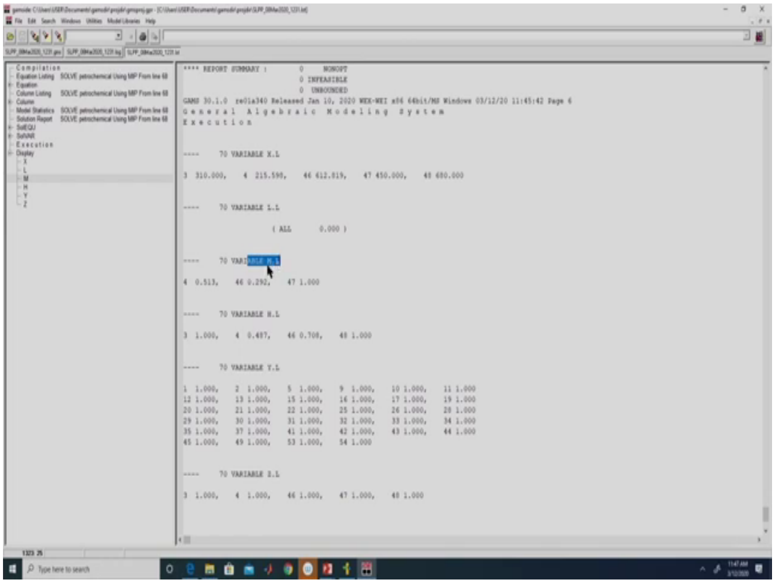
So, over here as discussed earlier we have written all the constraints right, then we are building the model whose name is petrochemical and we are including all the constraints written in this file in the petrochemical model. And then over here we are solving we are solving using a mixed integer programming solver we want to maximize the objective function.

The objective function corresponds to this variable OBJ and we are solving the model petrochemical right. So, at the end of the solution procedure we want GAMS to display the final values of the 6 set of decision variables right. We can either click on F 9 or click on this button to run it right. So, over here yes right so this is the list file right. So, this is the GAMS file which we had written this is the list file and this is the log file over here.

(Refer Slide Time: 28:45)



(Refer Slide Time: 28:59)

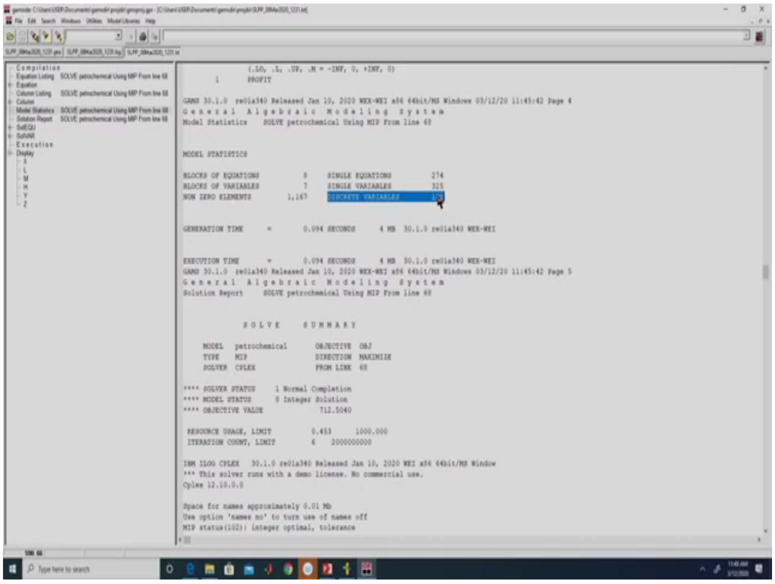


So, if we say open log file will also be opened in the editor window. So, this is the editor window right. So, let us look into the list file this is the indexing for the list file right. So, if we go to the display part, these are our solutions right. So, for process 3 we need to produce 310, for process 4 we need to produce 215.59 for 46, 47, 48 the appropriate value.

So, all the l variables have taken a value of 0. So, over here if you see when we display X dot L it only displays the nonzero values. If you remember what we did in MATLAB, we had to write a small piece of code to ensure that we get only the nonzero values over here when we display it displays only the nonzero values right.

So, only 3 variables of M are active right for process 4, process 46 and for process 47 right for all the other process L value is also 0 M value is also 0 except for these 4 processes the H value is 0 for all of the other processes. Similarly, the variable Y value is given over here and Z value is given for only these 4 processes 3,4, 46 ,47, and 48. Those are the only process, which are having some production quantity right so, that is the display statement.

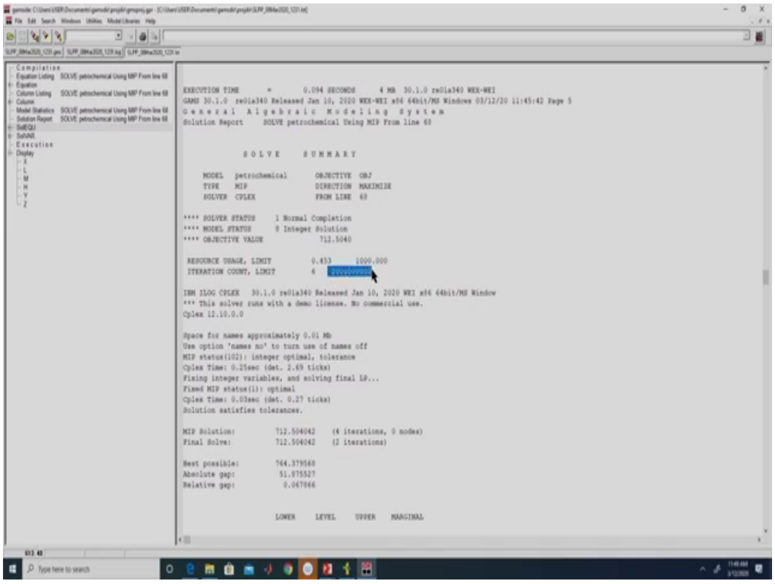
(Refer Slide Time: 30:12)



Let us look at the model statistics. So, the model which we solved has 325 single variables so, we have 54 processes so, 54 into 2 right for Y and Z. So, we get 108 variables and there are 274 equations. So, here if we look at the solution report right so, the objective function value which we obtained is 712.504 right and, it indicates that the completion is a normal completion what is being reported as an integer solution right.

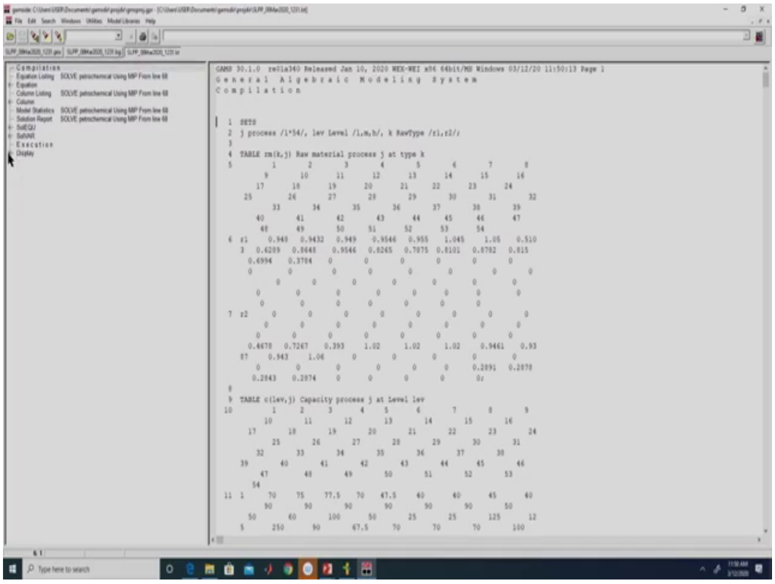
And resource usage if we see out of 1000 seconds it has used 0.453 seconds to solve this problem and out of these many iterations which is the default value in GAMS. It used only 6 iterations right. So, now if we go and include this option over here right, so we are changing the default option the default option is OPTCR is equal to 0.1 right. So, the solution which we got over here satisfies the tolerance right.

(Refer Slide Time: 30:33)

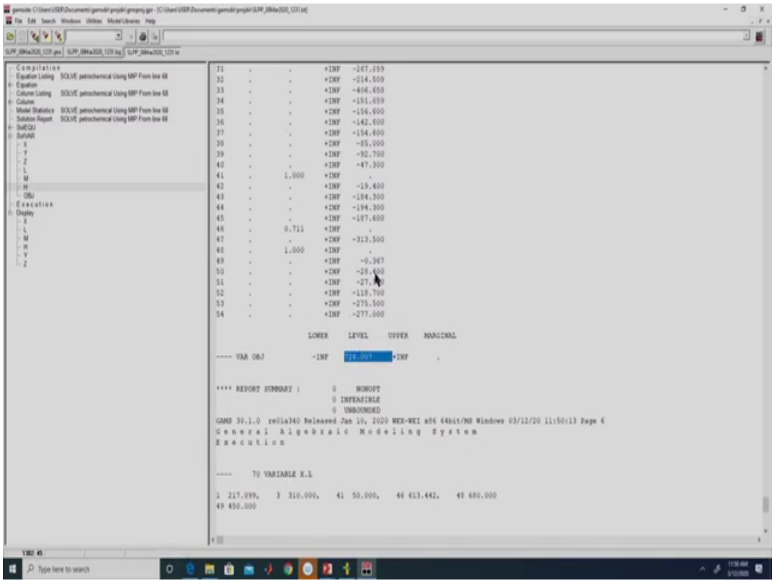



So, that is why it terminated right. So, let us close this file now we are solving after including this option statement right so, this one. So, if we look at our list statement if you remember our previous X this has changed right and our objective function value previously it was 712 now it is 726.007 right so, we get an improved solution. So, that is why it is important that once we get a solution, we also look into the reason for termination right.

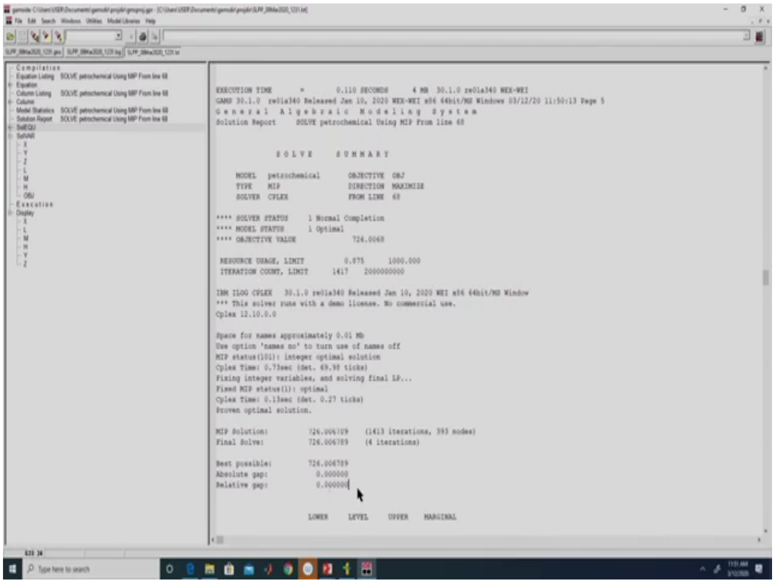
(Refer Slide Time: 31:25)



(Refer Slide Time: 31:33)




(Refer Slide Time: 31:48)




So, in this case the reason for termination would be solution report right. So, here previously if you remember we had solution satisfies tolerance right. So, now it is Proven optimal solution so for CPLEX solution is proven optimal solution, previously it took very few iteration, now it has taken 1417 iterations to reach this solution right. And now if we see the absolute gap and the relative gap is 0 right.

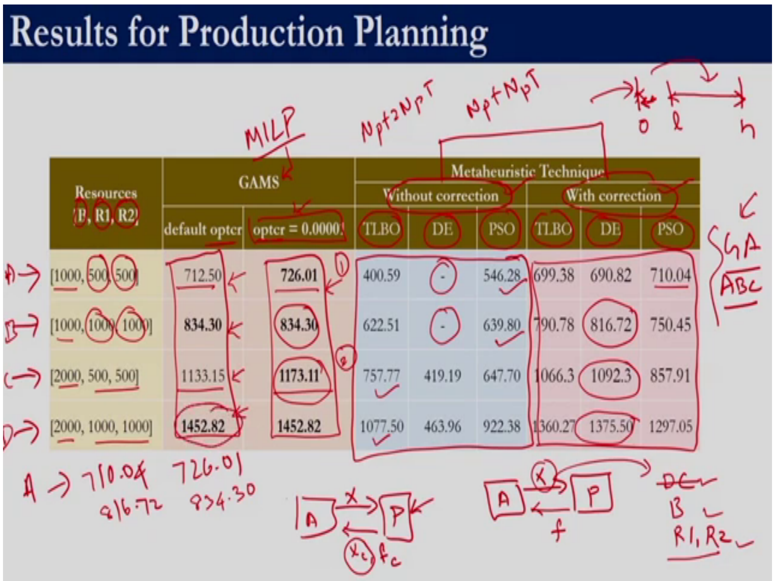
So, previously without this if we execute the absolute gap; if you say it is 51.877 and the relative tolerances 0.06 right. Since, the relative tolerance satisfies a default value, it is terminating. Now if we look at the absolute gap it is very high; and here also it explicitly says that the solution satisfies tolerance, it is not saying proven optimal solution right.

So, that is why it is important to look at the result right, not only in terms of the values given by GAMS, but also the reason for termination and if required, we need to change the options and then rerun the problem so, that we can get a better solution.

So, now that we have solved the production planning problem with metaheuristic techniques as well as we have formulated it as an MILP and solve it with GAMS.

Now let us compare the results which we obtained from metaheuristic techniques and those we obtained from GAMS. So, in this slide we have consolidated all the results right, this section shows the results by metaheuristic technique right. So, in metaheuristic technique if you remember we had employed two approaches right.

(Refer Slide Time: 33:08)




So, the first one was we were doing without correction, where in without correction our algorithm would only send the decision variables right whereas, the problem would evaluate the decision variable and would send back the fitness function value right. So, over here we had three types of constraints a domain constraint, budget constraint and two raw material constraints right.

So, in this case what we were doing is we were merely evaluating fitness of the solution given by the algorithm and returning it back to the algorithm right. So, here in without correction we were penalizing for all these three types of constraint right. Whereas, in the correction approach our algorithm would still send only the decision variables right whereas, our problem statement would send back the corrected decision variables and the fitness function corresponding to it.

So, here we in this problem statement itself we corrected for the domain constraint right. So, domain constraint if you remember for the metaheuristic technique the bounds are between 0 and h; it can give any value between 0 and h right. Whereas, in our problem we had this constraint that a value should be either 0 or it has to be greater than or equal to l and less than or equal to h right this was the permissible domain.

So, in correction approach what we did is if the value comes somewhere over here in between l and h it is fine if it is 0 it is fine. But if it is greater than 0 we reassign the value of the decision variable to 0. So, that is the correction approach which we had employed right. So, when we employed correction approach, we no longer have a penalty for domain constraint because that constraint is being taken care by the correction approach.

So, we had penalty only for the budget constraint and the two raw material constraints right and we evaluated the fitness of the corrected solution right. And the corrected solution also has to be returned back to the algorithm so, that was our correction approach right. All this we have discussed previously, you can go back and have a look at it.

So, for both the approaches with correction and without correction, we solved the production planning problem with Teaching Learning Based Optimization, Differential Evolution and Particle Swarm Optimization. We have chosen these three techniques because for these three techniques the number of fitness function evaluation is uniquely fixed if we fix the number of iterations and the population size right.

So, for TLBO it was Np plus 2 NpT whereas, for DE and particle swarm it was Np plus NpT right. So, that is why we had taken only these three algorithms, you can do similar comparison including Genetic Algorithm and Artificial Bee Colony optimization. So, here we have four cases, so this is case 1, case 2, case 3 and case 4. How do these cases differ is? Based on the amount of budget and the raw material 1 and 2 which are available.

If we say let this is case a right. So, in case A the amount of budget that is available is 1000. So, the amount of raw material 1 and 2 that is available is 500, 500. In case B the amount of budget that is available is still 1000 whereas, the amount of R 1 and R 2 which are available are 1000, 1000. For case C and case D, the total amount of budget available is 2000 and for case C the amount of raw material available is 500 and for case D, the amount of raw material available is 1000.

So, these are the results which we had obtained using GAMS right. So, GAMS also if you recollect depending upon this optcr setting, we did get different results right; so, first we executed all these four cases with the default optcr right. So, we did not specify optcr itself, the relative gap was at its default value right. So, in those four cases this is the profit that we obtained, and these are the four cases wherein we had set optcr to be this value.

Among these four cases if we see in two cases, we get a better result. So, here we get a better result and here we get a better result. So, here it is 113.15 here we have 1173.11 here it is 712.50 here it is 726.01 right. So, we have given you the GAMS code we expect you to go back and execute the program with this optcr and again check whether with this optcr do we really get the proven optimal solution or is it terminating for some other reason right.

So, if it is terminating for some other reason, then we need to appropriately change the option corresponding to it to see if we get a better solution. So, these are the solutions obtained for the four cases by the three different algorithms right. So, DE was not able to determine even a feasible solution. So, here if we compare within the metaheuristic techniques in this case we got PSO performing better.

Whereas, here we got, again PSO to be better and in these two cases TLBO was better right. So, d was consistently inferior for the algorithm settings which we had taken with different parameter settings, we might get a different value. And these are the results with respect to correction approach right. So, in this case we had also previously seen that correction approach gives a better solution than without correction right.

So, in this case if we look for case A. So, the best solution that we obtain for metaheuristic techniques is 710.04 right whereas, that what we obtain from the MILP formulation. So, when we say here GAMS it is the MILP formulation which we have discussed called with GAMS. So, here we get a value of 726.01 right. For the second case if we see the best value that we get from any metaheuristic technique is with DE right.

So, here we have 816.72 whereas, what we get from the MILP formulation is 834.30. Similarly, for the third case if we see the best value that we have is 1092.3 whereas, the best value that we get from the MILP formulation is 1173.11 and for the fourth case also the best value that we have is 1375.50 and over here it is 1452.82 right.

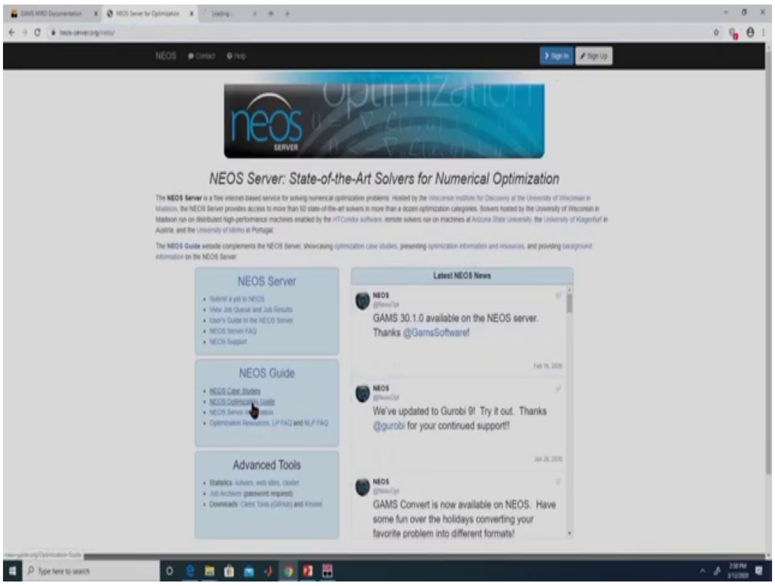
So, in all the four cases the solution that we get from the MILP formulation in GAMS with the appropriate settings is better than what we got in the metaheuristic technique right. This result comparison should help you to understand that it is worth the effort for transforming the problem statement into a mathematical formulation right.

So, if the mathematical formulation happens to be a linear programming or mixed integer linear programming, there is a reasonable chance that we will be able to get the global optimal solution. Another thing you need to remember is that when we use metaheuristic techniques right, we have to execute multiple runs right, so what we have shown you is the result of the best run.

And again, there are issues with setting the tuning parameters right whereas, in the MILP formulation we did not have to execute it multiple times it is a deterministic algorithm. So, similar to the settings we have to do for metaheuristic techniques for solving the MILP formulation also, we need to set a large number of factors.

So, the results that we get is dependent to a large extent on the settings that we employ for solving the Mixed Integer Linear Programming problem. So, now that we have seen GAMS right. So, let us look into NEO server; NEO server is particularly useful when we do not have a license right, the demo license which we use is only for a restricted number of variables and constraints right. Now let us look into how to solve a problem using the NEO server.

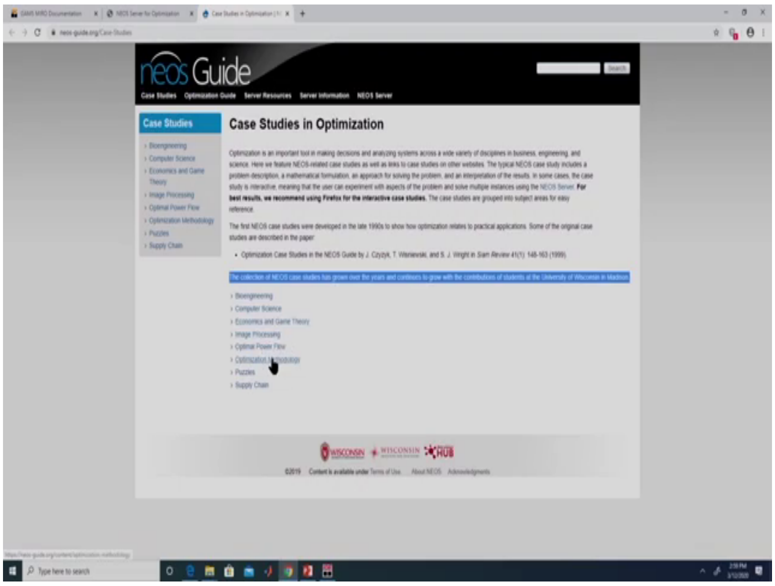
(Refer Slide Time: 40:40)



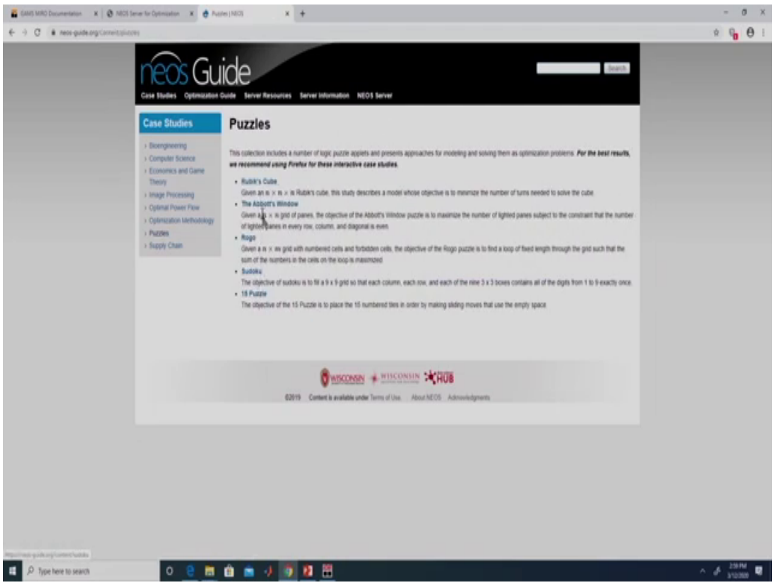

So, this is the website for NEOS right. So, neos hyphen server dot org slash neos. So, if you look into here, you will get lot of information about NEOS server including some case studies.

The case studies have been divided into certain areas like Bio engineering, Computer science, Economics and Game Theory, Image Processing, Optimal Power Flow, Optimization Methodology, Puzzles and Supply chain right.

(Refer Slide Time: 40:55)

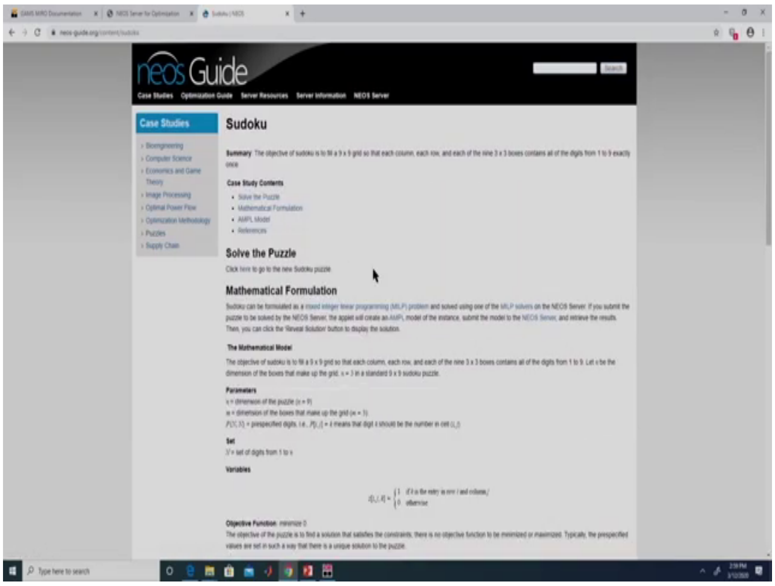



(Refer Slide Time: 41:06)

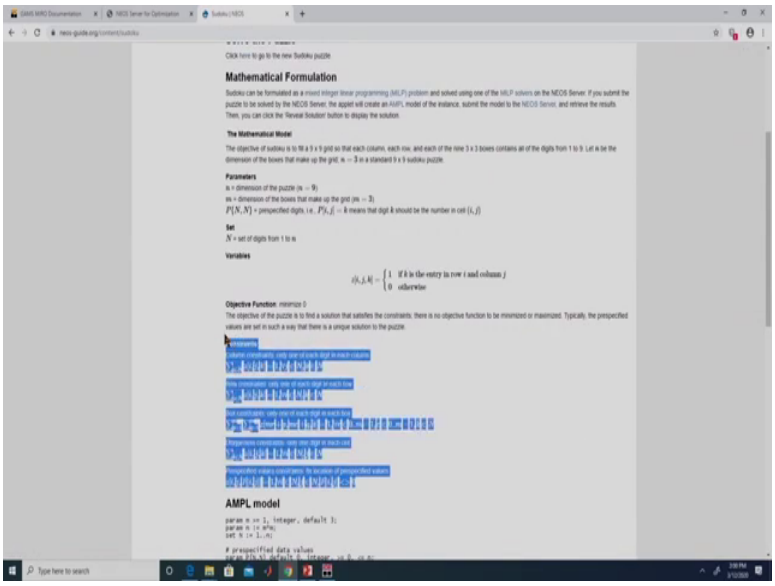



If we look into puzzles so, there are these five puzzles which can be solved using a GAMS code right. So, if we click in on the Sudoku problem this is a mixed integer linear programming problem for sudoku problem right. So, here they have given AMPL model. So, this NEOS server supports not only GAMS model, but it supports various other types of models right.

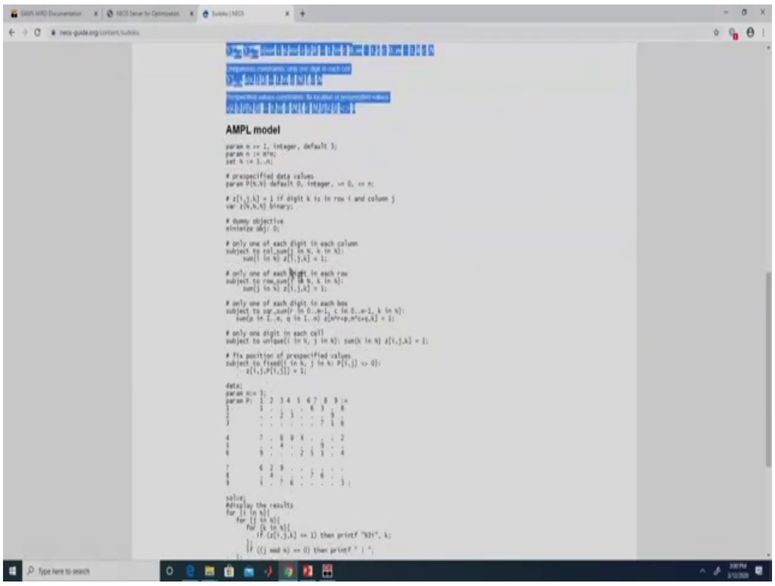
(Refer Slide Time: 41:11)



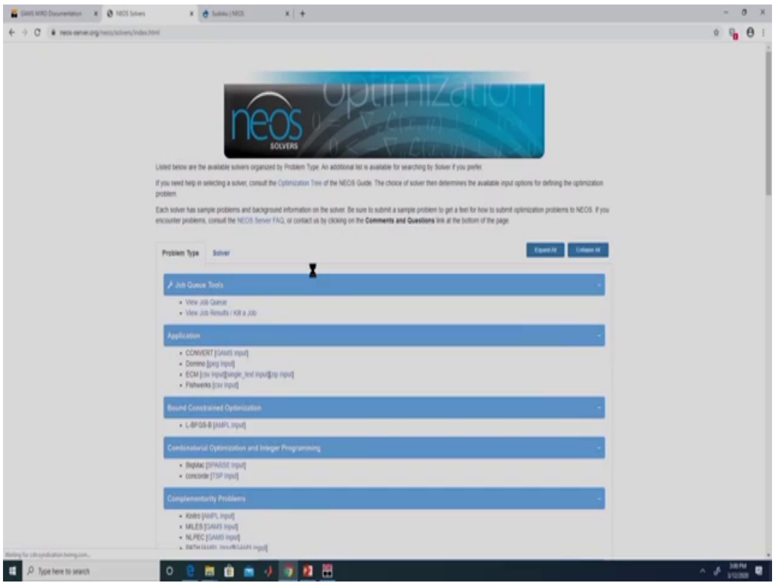
(Refer Slide Time: 41:13)



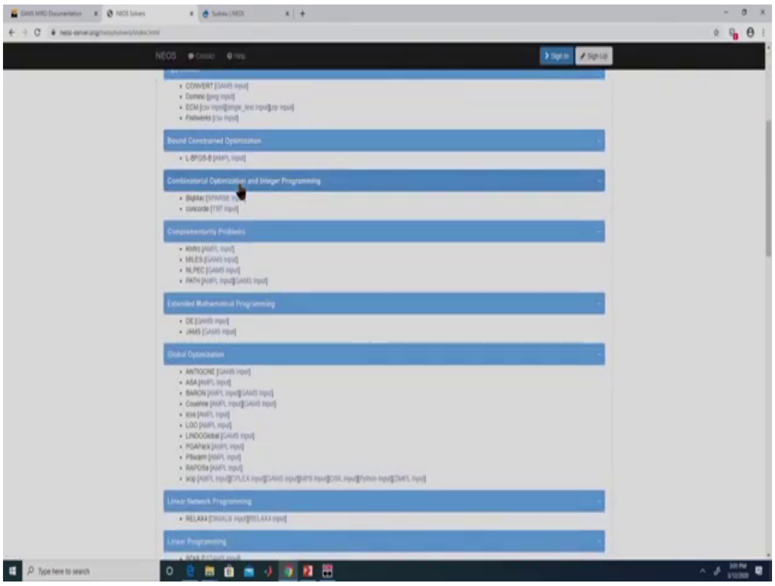

(Refer Slide Time: 41:17)




(Refer Slide Time: 41:30)


neos
SOLVERS
optimization
Listed below are the available solvers organized by Problem Type. An additional list is available for searching by Solver if you prefer.
If you need help in selecting a solver, consult the Optimization Tree of the NEOS Guide. The choice of solver then determines the available input options for defining the optimization problem.
Each solver has sample problems and background information on the solver. Be sure to submit a sample problem to get a feel for how to submit optimization problems to NEOS. If you encounter problems, consult the NEOS Server FAQ, or contact us by clicking on the Comments and Questions link at the bottom of the page.
Problem Type
Solver
Job Queue Tools
View Job Queue
View Job Results / Kill a Job
Application
CONVERT [GAMS input]
Domino [jpeg input]
ECM [csv input][single_text input][zip input]
Fishwerks [csv input]
Bound Constrained Optimization
L-BFGS-B [AMPL input]
Combinatorial Optimization and Integer Programming
BiqMac [SPARSE input]
concorde [TSP input]
Complementarity Problems
Knitro [AMPL input]
MILES [GAMS input]
NLPEC [GAMS input]
Type here to search

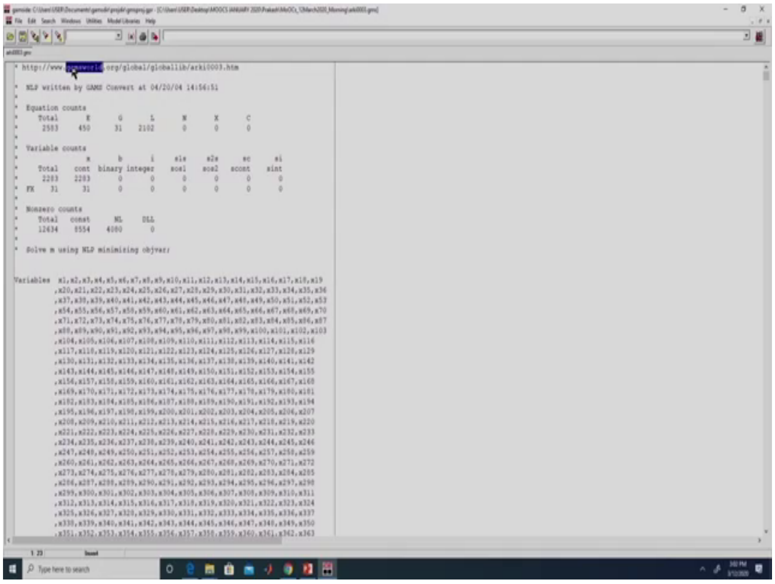
(Refer Slide Time: 41:32)



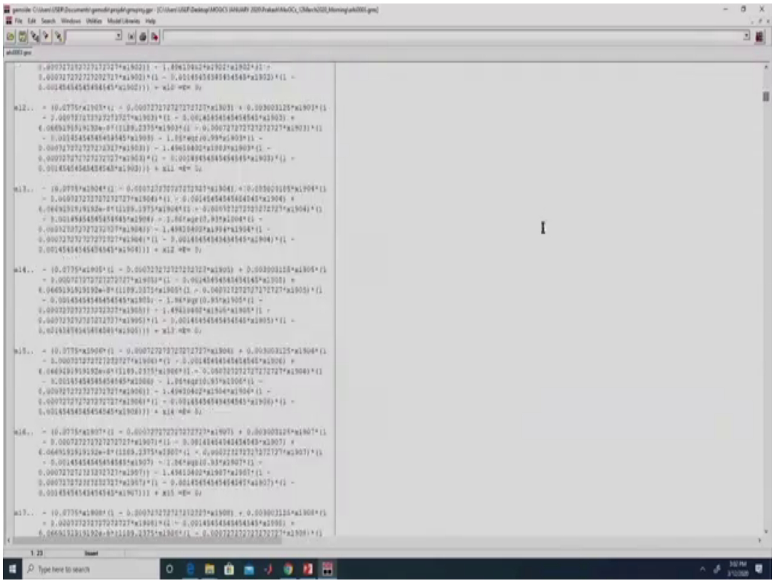
So, if we want to submit a job to NEOS we need to click over here submit a job to NEOS and then, over here we need to look for the appropriate solver that we want to solve right. So, for example, under Global Optimization, we have these many solvers. So, not every solver is compatible to get every type of inputs over here this ASA accepts only AMPL input whereas, the solver BARON accepts AMPL input as well as GAMS input right.

So, let us click on GAMS input because we have a GAMS file now. We are going to solve a non-linear programming problem right. So, the problem that we have is downloaded from gams world dot org this, gams world dot org has a collection of test problems right. So, as you can see this particular problem has a large number of variables right it has 2583 constraints and it has 2282 variables if we run this file over here right.

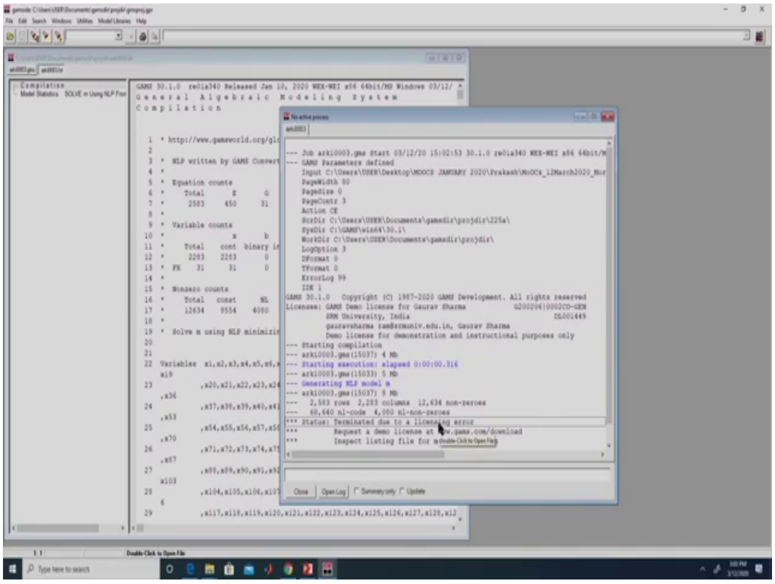
(Refer Slide Time: 41:56)




(Refer Slide Time: 42:08)

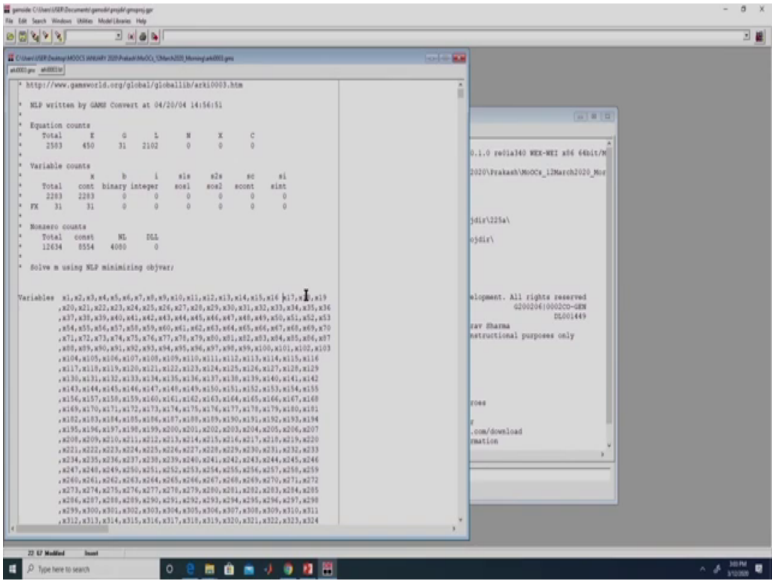

File Edit Search Windows Utilities Model Libraries Help

(Refer Slide Time: 42:18)

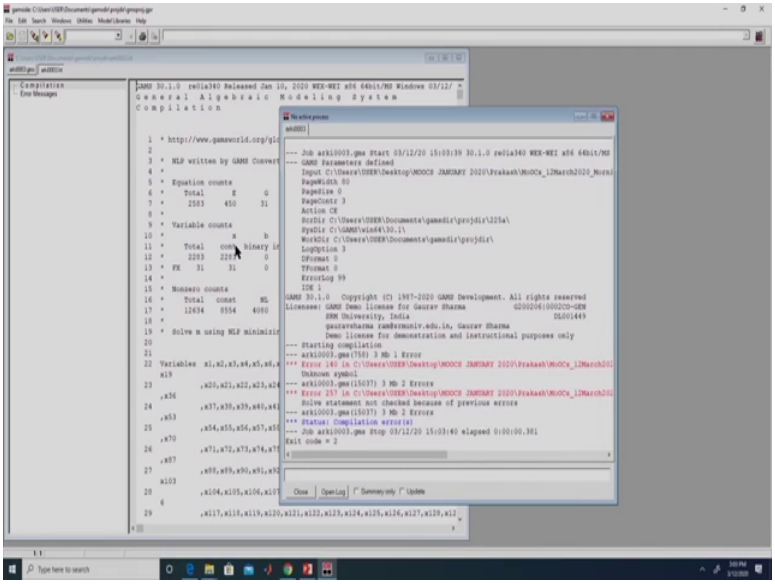


So, since we have only a demo license right it gives a message status terminated due to a licensing error, if you happen to face this type of licensing error so, then we can choose the NEO solver right. So, but if there is some error in this file; so, for example, let us just introduce an error over here let us just remove this comma right.

(Refer Slide Time: 42:31)

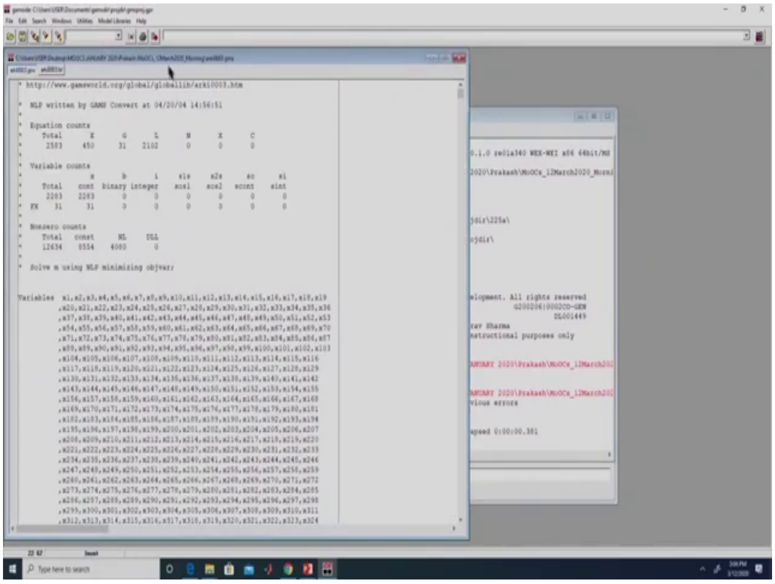



(Refer Slide Time: 42:42)

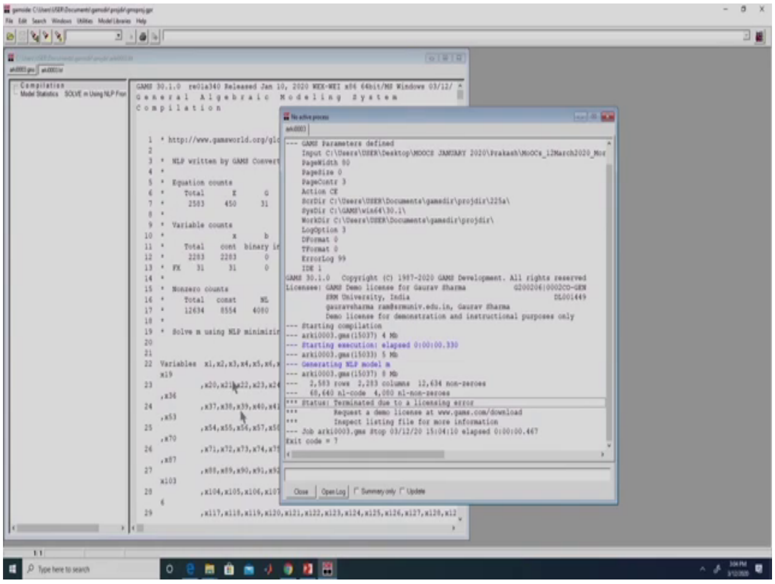


So, now if we run it over here it will not give a termination error, but it will give a compilation error right, so there is an error right. So, that way even though we have only a demo license we can ensure that our code is running fine. So, once we know that the code is working fine, we can upload it on to the NEO server for solving purpose right. So, let us go back and put this comma again right. So, now if we execute this file as previously, we should get a licensing error right, it is terminated due to a licensing error right.

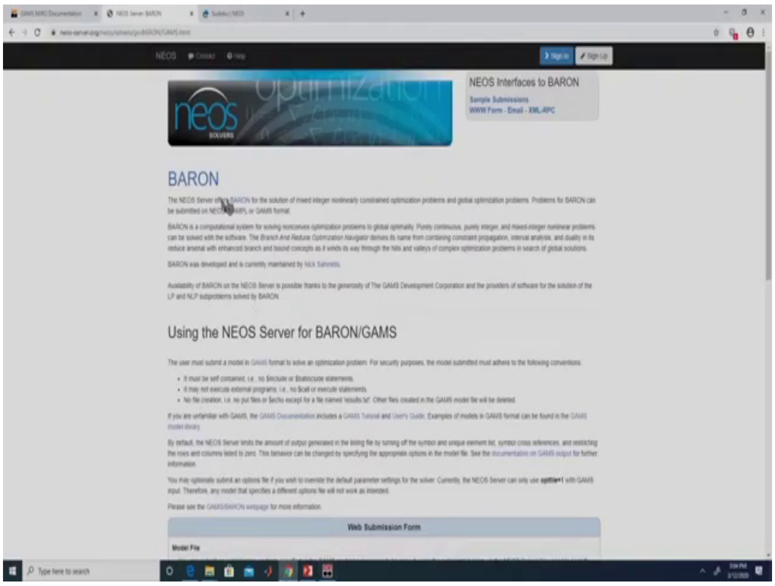
(Refer Slide Time: 42:59)



(Refer Slide Time: 43:05)

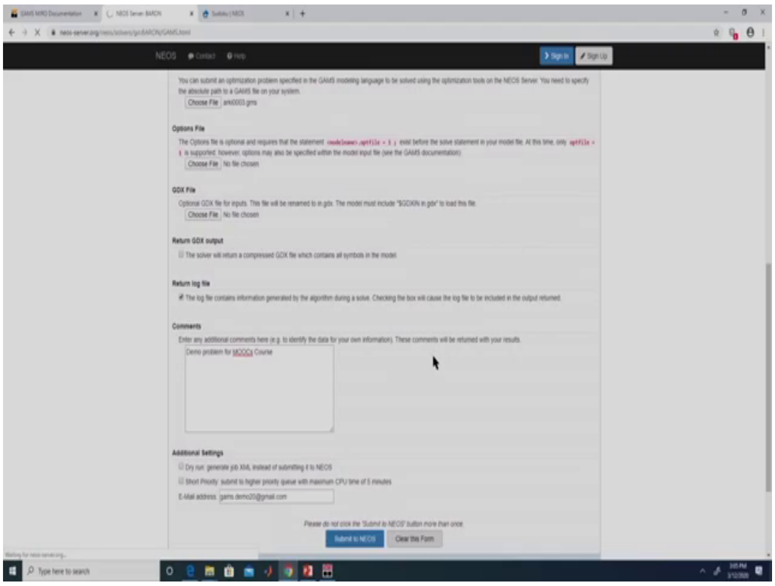



(Refer Slide Time: 43:10)



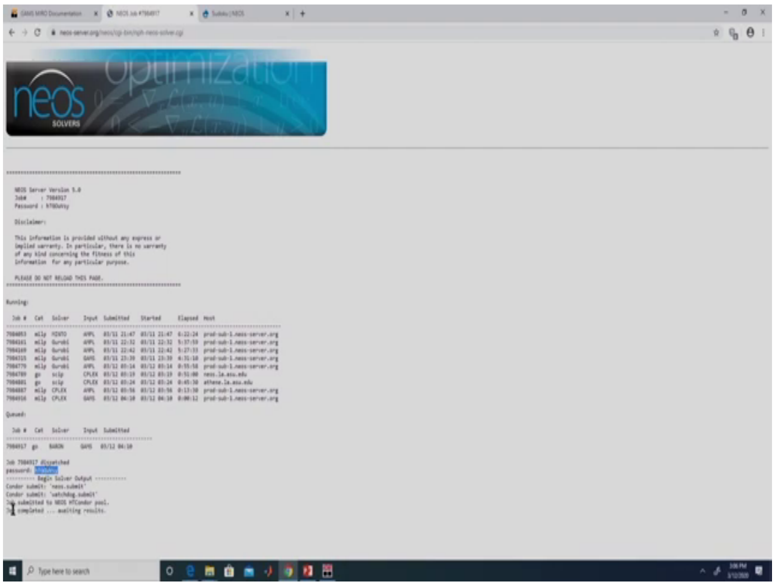

So, let us now go back to this BARON solver right. So, we need to first submit our model file so, choose file so this is the model file that we want to solve so, we upload that file right. So, if we have an options file, we can give that option file. This GDX file we have not discussed right, but it can be used to give inputs to the GAMS code right. So, if we require a log file, we can we need to click this checkbox and over here we can give some comments right.

(Refer Slide Time: 43:13)



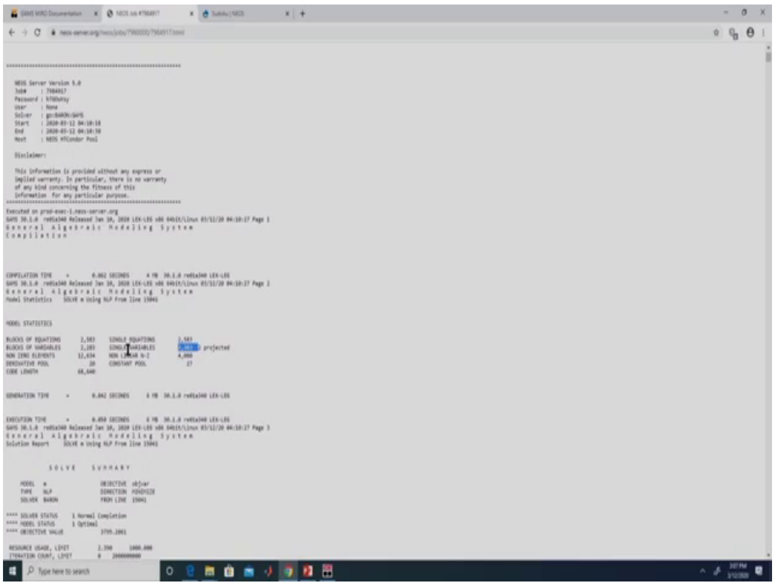
So, over here we can give a email id. So, the results would be sent to this email id also. We can see the results over here as well as we can get the results over email right. So, we need to give submit to NEOS. So, depending upon the queue on the server right our problem will be placed in a waiting list usually it does not take much time to clear the queued right. So, over here we have been assigned a job number so, this is our job number 7984917 and a password is shown over here right.

(Refer Slide Time: 43:59)



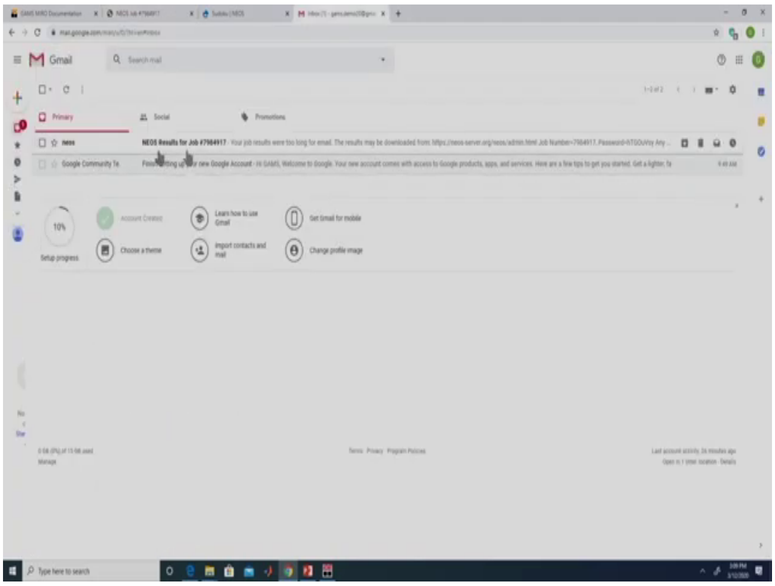
So, now the job is completed, it was awaiting for results. Now, we have been able to get our results right. So, the job number password right and after this whatever we had discussed in the list file holds true over here, this is the list file. So, our solver status is normal completion so, the objective function value is 3795.2061 right so, we had 2583 equations and 2283 variables.

(Refer Slide Time: 44:14)

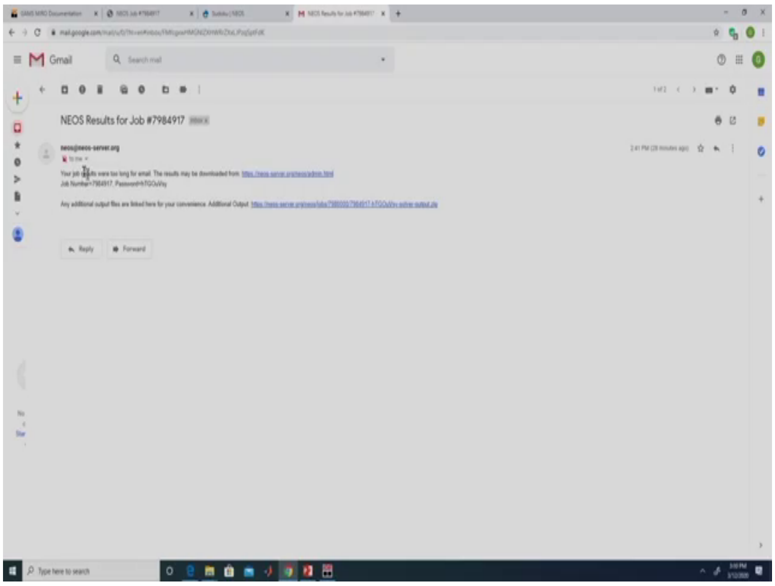


The results are also e mailed to us right in this case we had given our email id right. So, this is our job number so, we have received this email from NEOS right. So, since the job results were too long for email the results may be downloaded from this link, they have provided the job number and the password. So, if we click on this, so this is the job number that we gave and this is the password right. So, we want to view the job results right.

(Refer Slide Time: 44:44)

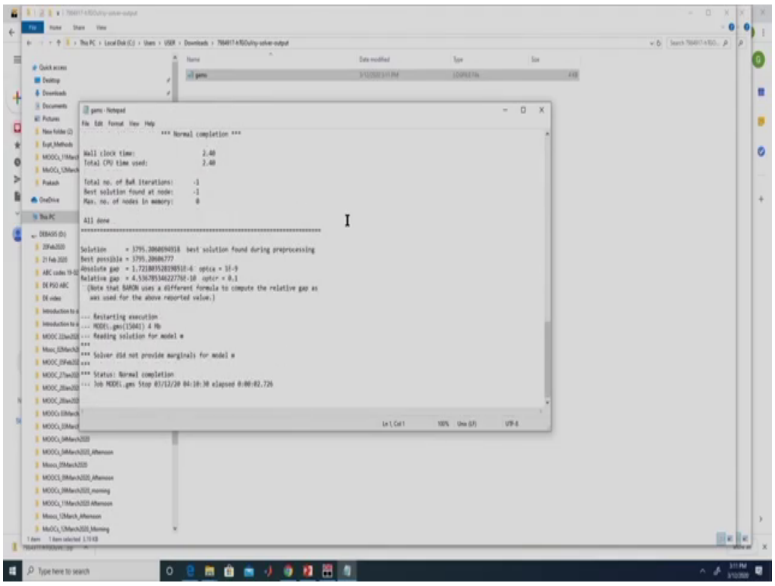



(Refer Slide Time: 44:47)



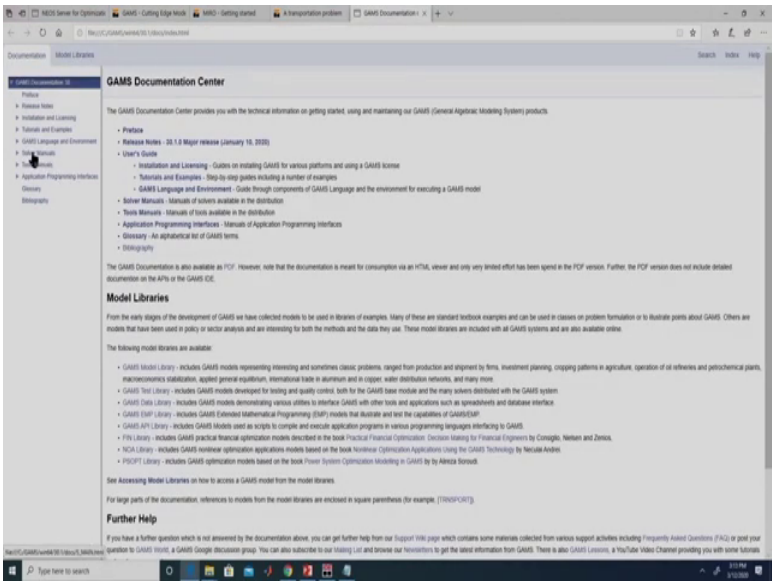

So, this is the same file which we see over here right which we had got after solving it right. So, if it takes a little bit longer to solve depending upon the queue, we can also get the results through the details given in the email right. So, in this case the problem was a bigger problem so, we did not get the list file as part of the email otherwise, we would also get the result file as part of the email itself. So, if you remember we had also requested for the log file.

(Refer Slide Time: 45:51)



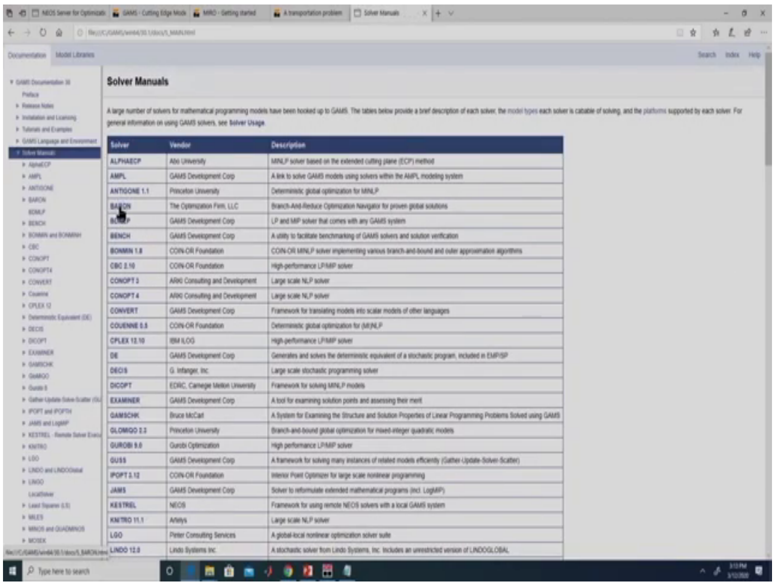

If you click on that then this will contain our log file so, this is our log file here we can see the actual absolute gap and the relative gap and the default options in this case we did not change the default option. So, the default option for optca is 10 power minus 9 and the default option for optcr is 0.1. Similar, to our discussion with CPLEX, BARON also uses a different formula to compute the relative gap, additional details on this you can find from the solver manual of BARON.

(Refer Slide Time: 46:17)

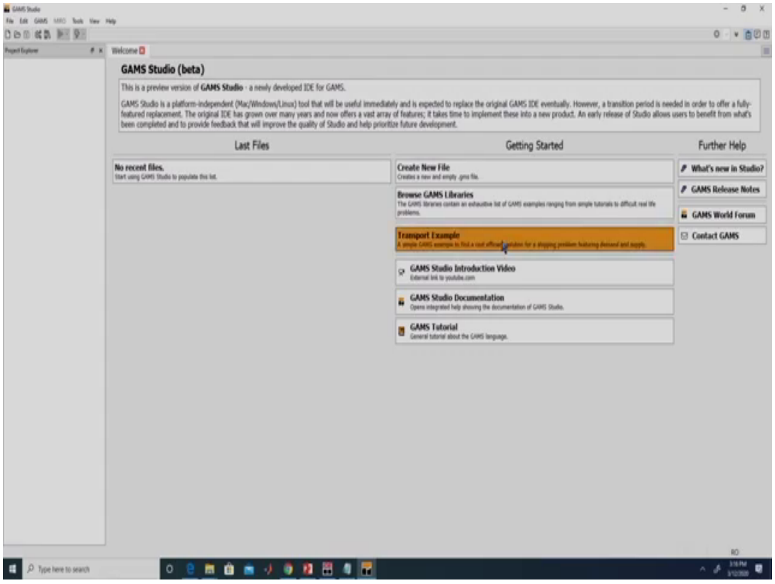



So, in this offline help documentation so, we have the solver manuals also. So, each solver has its own manual so, one can go into this BARON manual and look into further details about calculation of optcr and other relevant things. This is how we can use GAMS and NEO sever to solve problems involving larger number of decision variables and constraint. So, while installing GAMS you would have seen two options one is GAMS IDE and other one is GAMS Studio right.

(Refer Slide Time: 46:20)



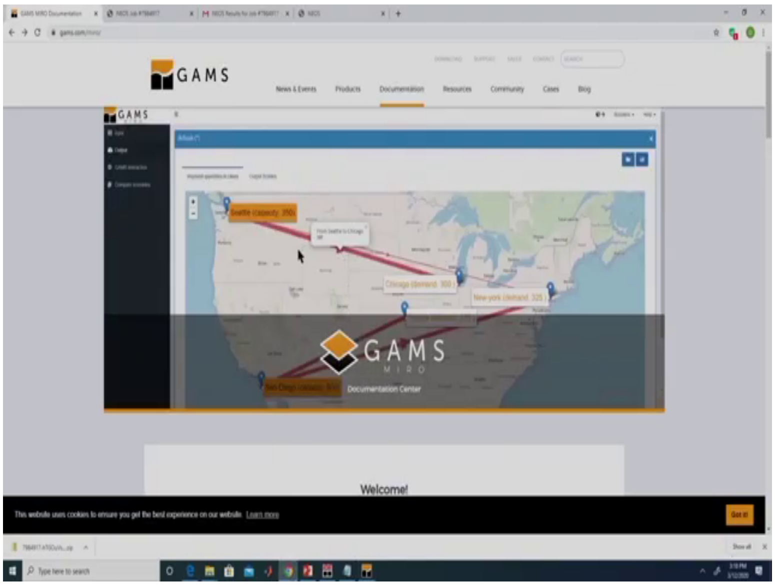

(Refer Slide Time: 46:42)



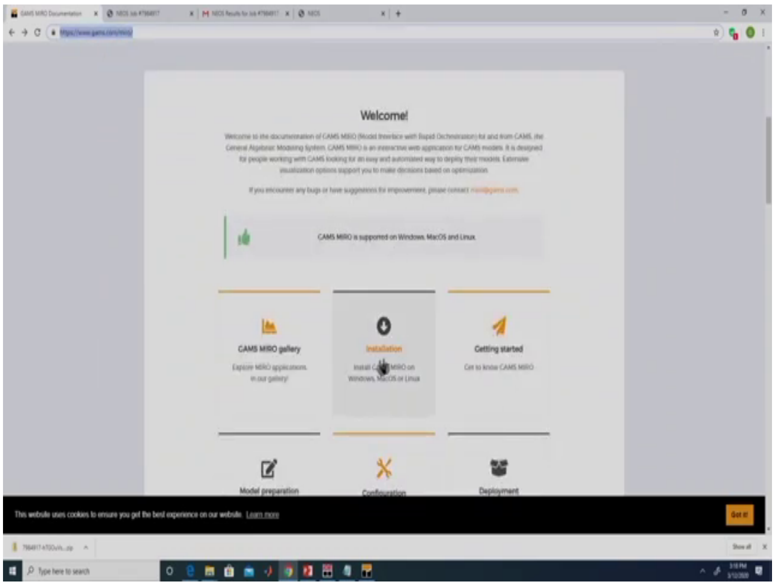

So, if you look into GAMS studio from here, we can get a lot of details right. So, for example, the transportation example which we have discussed previously is directly accessible from here, it also gives an introduction video about GAMS studio right. So, it links to YouTube, it also provides a lot of other information we leave it to you to explore that.

You can also visit this GAMS world forum to interact with the user community right. So, here you will see something called us Miro and by default it will be disabled. If you are working with a demo license so, you can have a look at it over here, gams dot com slash Miro, for GAMS Miro you need to have an additional license right. So, over here a large number of examples are already given, let us look at a couple of examples which they already have.

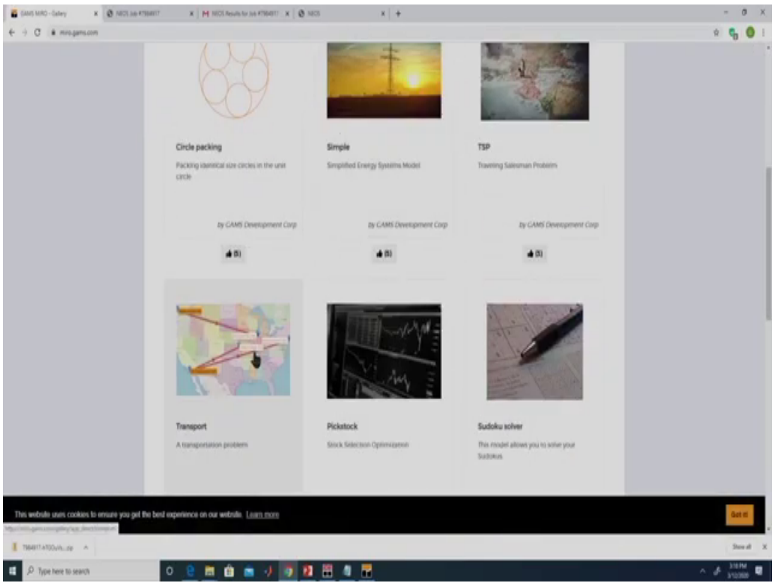
(Refer Slide Time: 47:13)



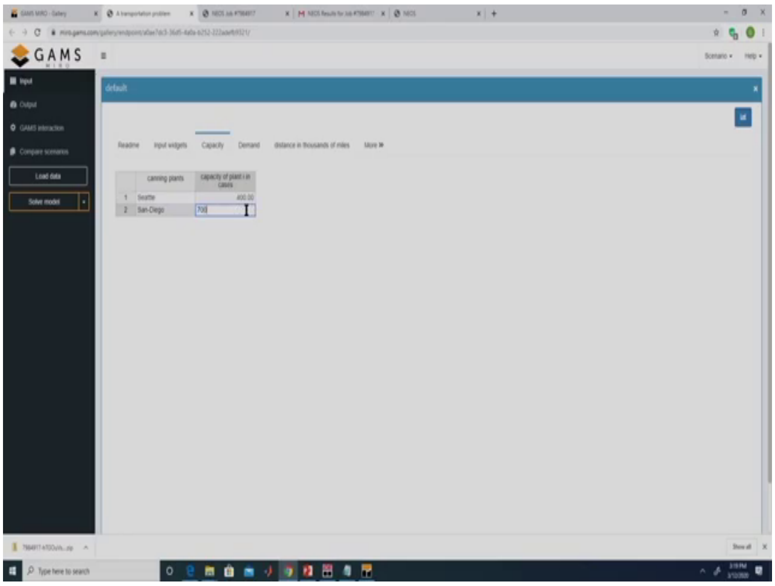

(Refer Slide Time: 47:19)




(Refer Slide Time: 47:28)

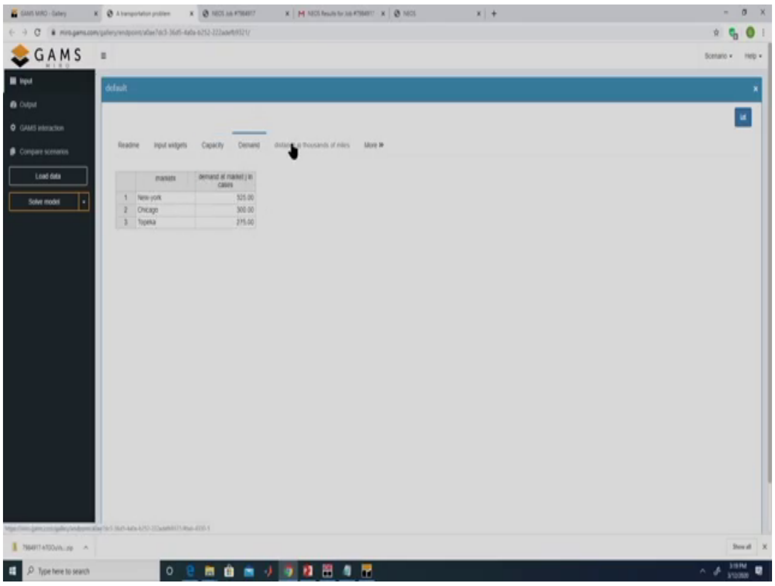



(Refer Slide Time: 47:32)

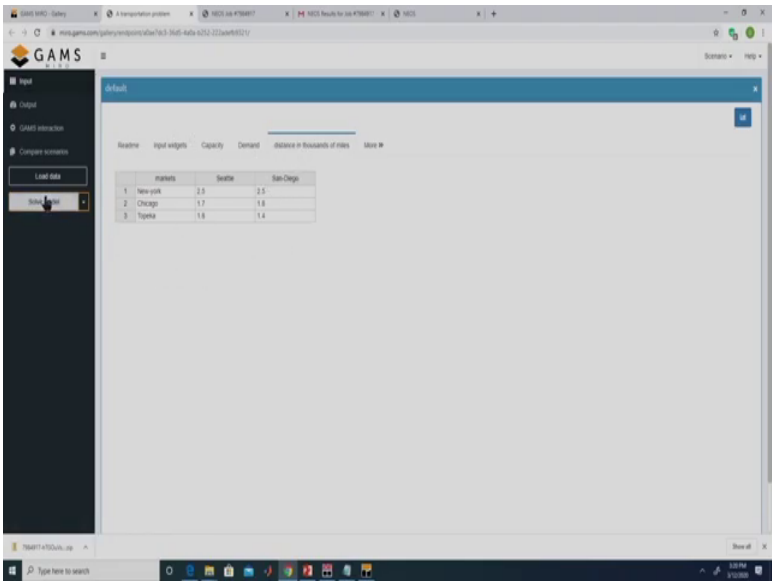



So, one is the transportation problem it gives details of the problem to remember the problem we had two plans whose capacity is to be specified right. So, over here we can change the inputs right 400 and 700 let us say. And then each of the three markets New York, Chicago, Topeka had its own demand and we also had specified the distance between the market and the plans right. So, the plans were located at Seattle and San Diego and the markets were New York, Chicago and Topeka.

(Refer Slide Time: 47:42)

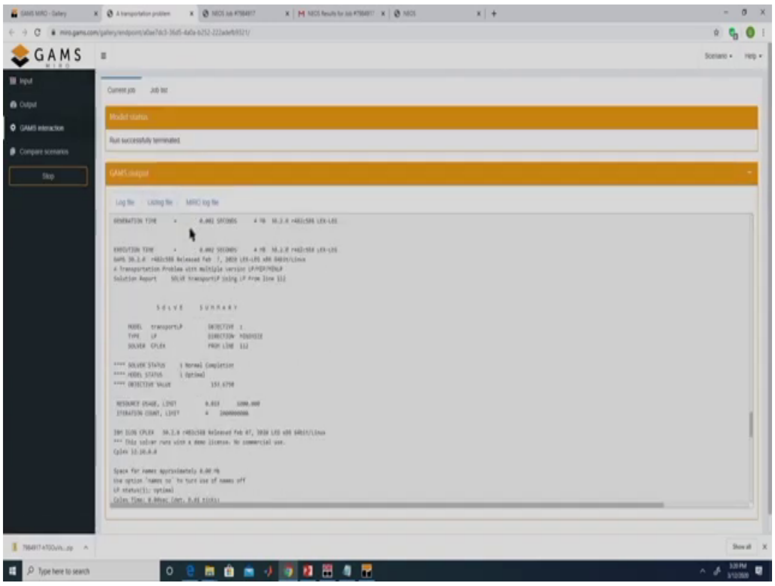



(Refer Slide Time: 47:46)

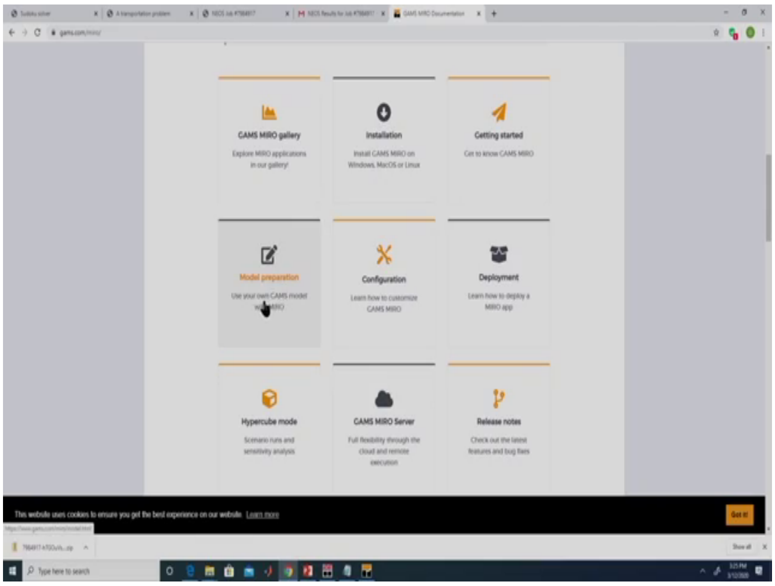



So, the model has already been deployed using Miro, that is why we are able to see this right. So, here if we solve this model so, we have this listing file we have discussed the contents of the listing file and log file earlier. So, over here if we see they have given the procedure how to develop how to use our GAMS model with Miro.

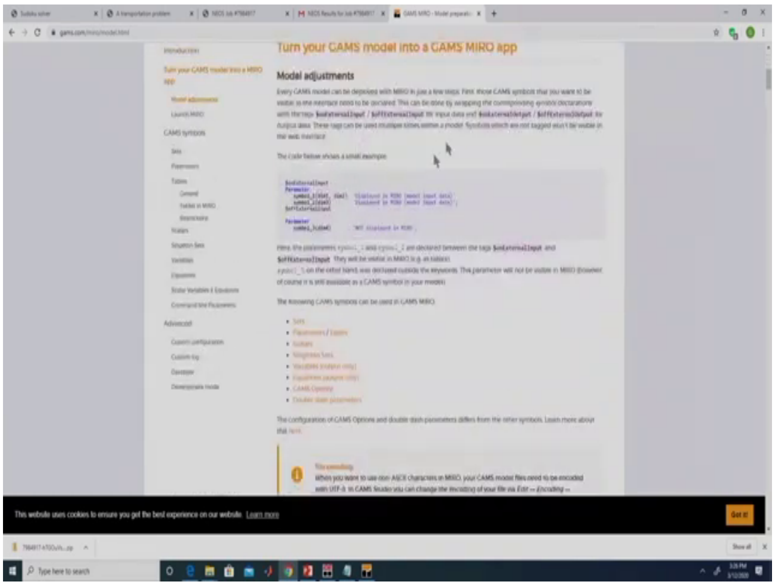
(Refer Slide Time: 48:04)



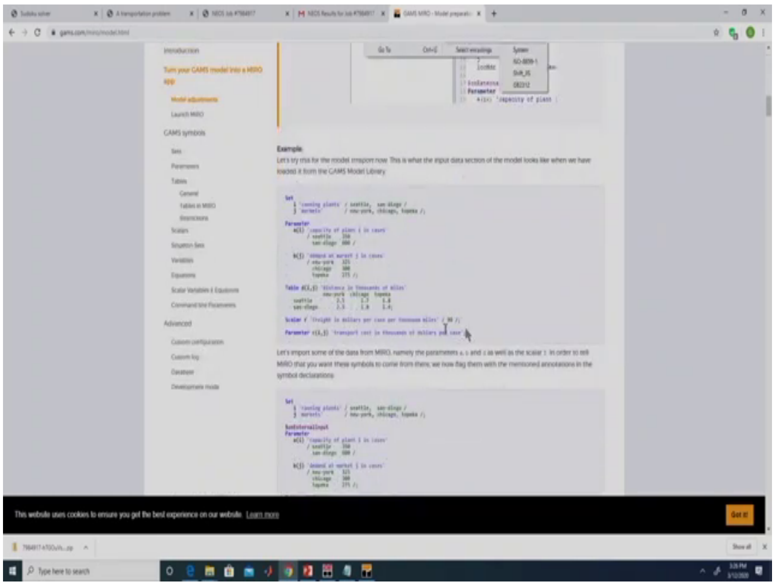

(Refer Slide Time: 48:11)




(Refer Slide Time: 48:18)

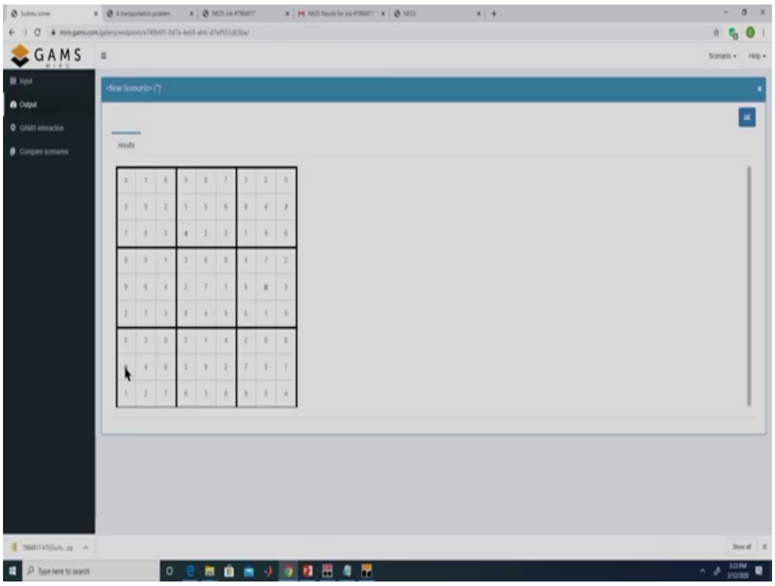

Turn your GAMS model into a GAMS MIRO app
Model adjustments
This website uses cookies to ensure you get the best experience on our website. Learn more

(Refer Slide Time: 48:22)

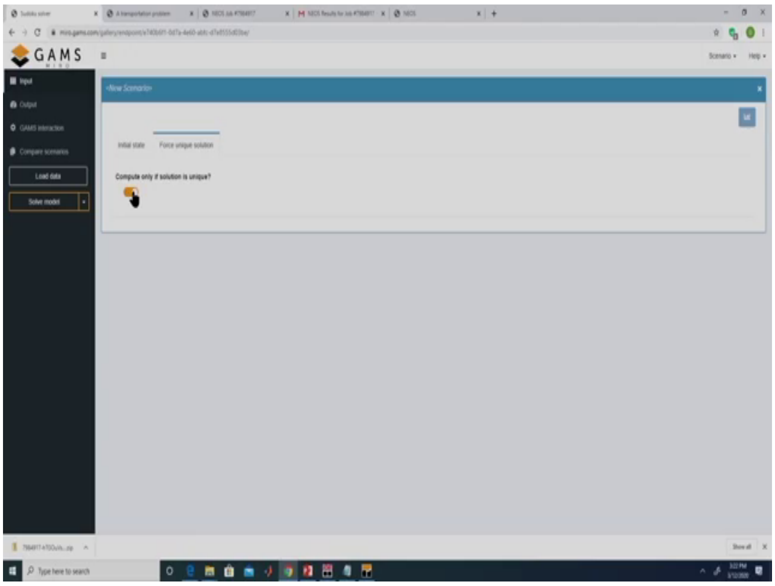


So, here they have shown it for the transportation problem right, so, this is the transportation problem which we had discussed, we need to add a few statements to our model so, that it can be executed on Miros. We can have our model deployed over the web. So, the another example which they have is the sudoku example; if you look at input this is the input right in this case, they have something called as Force unique solution.

(Refer Slide Time: 48:35)



(Refer Slide Time: 48:44)




So, if this is on it will compute a solution only if the solution is unique. Let us put it off right and over here a let us say we give a value of 4, 8, 7, 6 over here. And if we solve the model so, it displays the solved sudoku problem right so, the values which we gave here 6 over here, 4 over here, 7 over here and 8 over here. So, it has been able to find out the rest of the entries while satisfying all this constraints right.

So, here we have not looked into the formulation which is actually solving this problem. Similar, to the transportation example GAMS file can be developed for the mixed integer linear programming model right and that can be deployed using the Miro software. So, as it can be seen that GAMS along with the NEOS server can be used to solve problems involving large number of variables and constraints and we can also easily deploy our model using the Miro software.

In this session we have seen how to use GAMS to solve linear programming problems, non-linear programming problems, mixed integer linear programming problems and mixed integer non-linear programming problems. Additionally, we also saw how to code the production planning problem in GAMS and how to analyse the results, with that we will conclude this session.

Thank you.