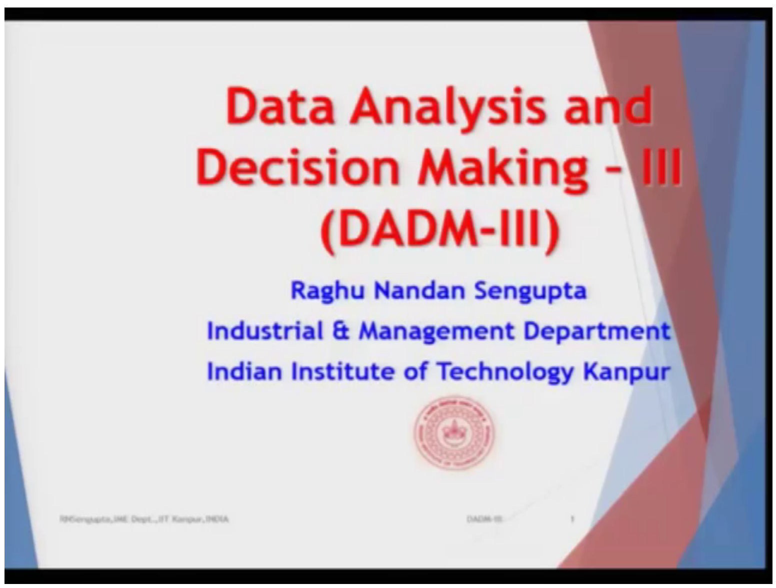
Data Analysis and Decision Making - III

Professor Raghu Nandan Sengupta
Department of Industrial & Management Engineering
Indian Institute of Technology Kanpur India

Lecture 41

Welcome back my dear friends a very good morning, good afternoon, good evening to all of you wherever you are in this part of the globe.

(Refer Slide Time: 00:21)



So this is Data Analysis and Decision Making 3 course under NPTEL MOOC series. And as you know this total course contact hours is 30 hours which when broken down into the number of lectures considering each lecture is for half an hour is there are 60 lectures and the total spread of this total course would be for 12 weeks considering we have 5 lectures in a week. And after each week we have that assignments and that is in totality 12 assignments and after the end of this course you will basically be taking the final examination And my good name Raghu Nandan Sengupta from the IME department at IIT Kanpur. So if you remember we are discussing about the concept of integer programming and how would you solve it.

So basically there were two methods, one was the Gomory cut method where we basically had that constraint boundaries modified accordingly such that corner points where you will basically find the solution would be only integers and then one basically was the branch and bound method where you basically start from the primary known and considering one of the decision variables x or x1 whatever it is.

It is basically a non-integer, you will basically partition into two distinct sets and then proceed in the direction by taking, if you remember the problem was basically P knot was the original problem, you broke down into p1 and p2 depending on what branching and we will proceed in the direction where the optimum solution was increasing or decreasing depending on whether you are trying to basically maximize the problem or minimize the problem.

And obviously they would be degeneracy, inconsistency and so and so forth. You will stop in those directions, so if you remember the diagram so there were many notes which were basically color in different colors signifying that any improvement in the objective function either maximization or minimization would not be possible. So hence we will basically not proceed in that direction, once you find out one of the decision variables had been converted into integer you will basically consider the next set of decision variable which are non-integers and proceed accordingly.

So obviously the final result would be the best optimum solution for the linear programming concept where you consider initially all the decision variables were continuous. Now, before we proceed into other areas I thought that I will discuss two problems, the problems are not interesting, what is interesting is that how you formulate. The reason is that if you remember in the first 1 or 2 lectures when we have started the concept of operation research and optimization I did mention that the obviously the techniques which you are going to learn are the solution techniques.

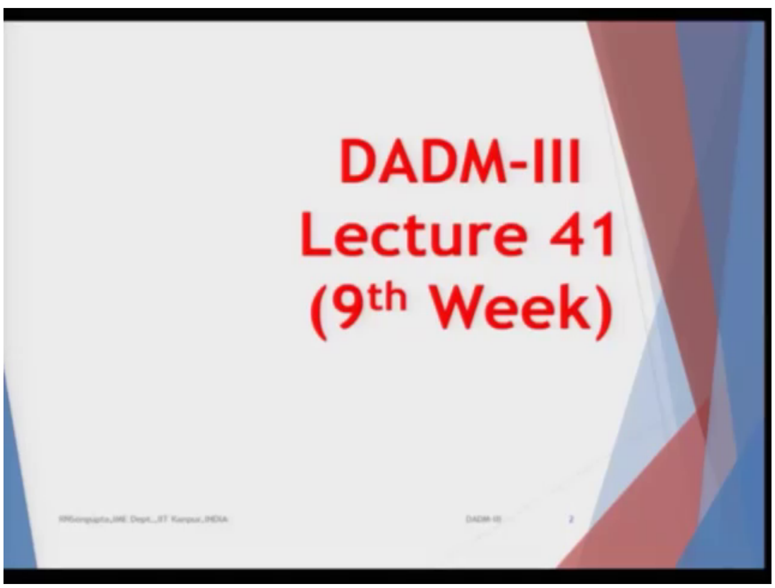
So, solution techniques, linear programming would be generally the concept of north west corner method, Vogel approximation method then you will consider later on the steepest descent method, steepest ascent method depending on which way you are trying to handle the problem, there would be quadratic programming, there would be integer programming, the concept of

Gomory cut, branch and bound, reliability method, robust optimization method, obviously we will consider the goal programming method so and so forth.

So, the general techniques which you are going to learn are the solution method. But what was very important to understand is that once you are given a general scenario for problem with all the details, our main task if you remember I had said basically to formulate the problem in the right manner such that you are aware that what is the objective function, what are the constraints, what are the decision variable based on which you should proceed.

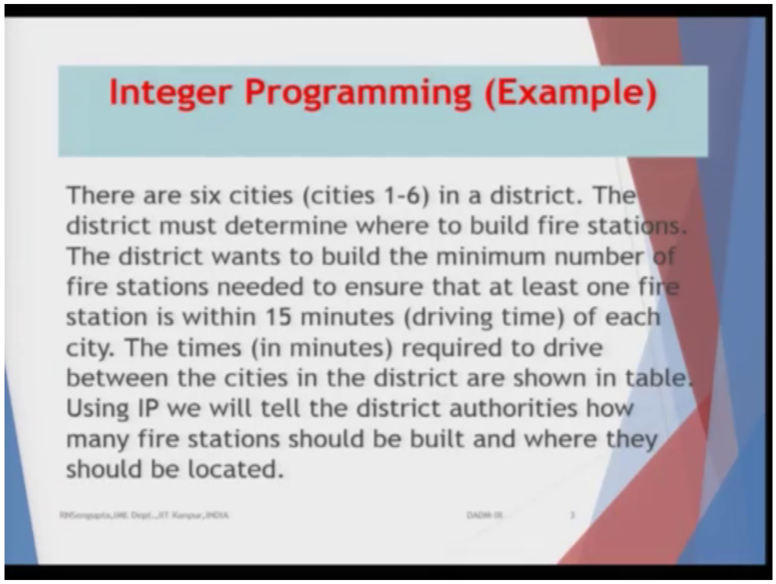
So I will basically today either in this class and also in the next class which is the forty first and the forty second class I will discuss a general simple but interesting integer programming problem with their all their details. So here we start the first problem.

(Refer Slide Time: 04:31)



So this is the forty first lecture as I mentioned which is basically we are going to start the ninth week of the classes.

(Refer Slide Time: 04:34)



Now, consider the scenario is like this, there are 6 cities so in a district, so the district can be either in UP, can be in Andhra Pradesh, can be in West Bengal, can be any other State of India. And the district it could have been like a city also with different municipalities and areas.

So but we will consider this is a district, the district must determine where to build the fire station so obviously when there is an emergency the time taken for the fire engines to leave their stations and reach the area where there is an emergency where the fire has broken out or some emergency has taken place should be as fast as possible or as minimal.

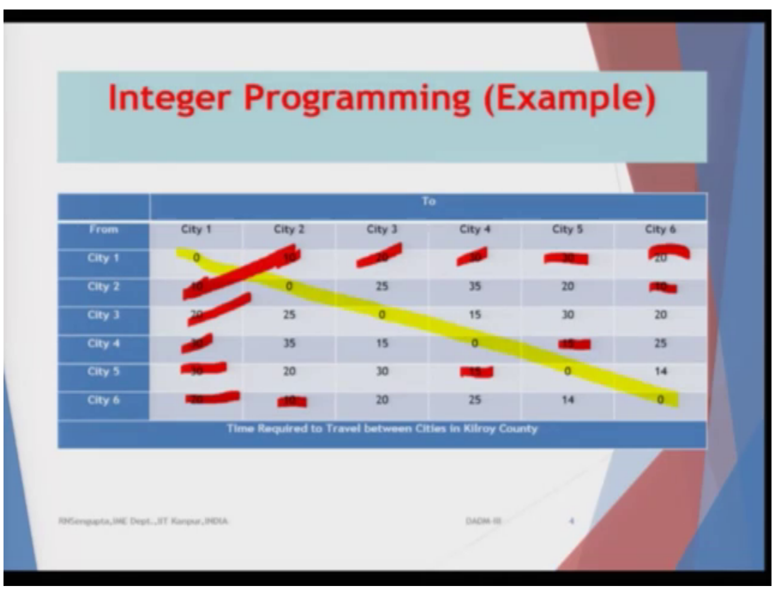
Obviously you cannot make it 0, you will basically try to reach those places for the emergency as soon as possible. So the district must determine where to build fire stations, it also knows the district authority knows or the district magistrate knows or the superintendent police or the SSP knows. They want to build the minimum number of fire station. So obviously the question is that why minimum? Because in the minimum number of fire stations if it is able to carted to all, then need would obviously reduce your total cost because trying to build up a building for the fire station have people employed there have different types of fire engines, they are obviously until costs.

What will the district authority wants basically minimize the cost and they would also ensure that at least one fire station is within 15 minutes driving distance. So if you remember when I mentioned they should reach the position or the place of emergency as fast as possible considering that the terrain of that region, the actually population of that district and all the constraints or all the actual practical constraints which are feasible.

The district administration thinks that any fire station should reach the emergency point within 15 minutes. So, driving time of each of the city so it can definitely be less than 15 minutes but it should not exceed 15 minutes. The time in minutes required to drive between the cities in the district are shown in one table which we will come to that. So, they can definitely their times can be 13.2 minutes, it be can be definite be 72.5 minutes, whatever it is those values are possible.

So we will try to utilize integer programming, we will tell the district authorities or we will basically tell the district authorities how many fire stations should be build and where they should be located such that this 2 important criteria's which has been put forward by the district authorizes are met. Number 1 the number of fire stations should be minimal point 1 and point number 2 maximum time taken for any fire engine to reach the emergency point is basically 15 minutes or less. So, here is the general lay out, so layout (of the) or matrix which will basically gives you the time if you remember I have mentioned the time.

(Refer Slide Time: 07:37)



So along the first column we have city 1, city 2, city 3, 4, 5, 6. So say for example if I am giving a very hypothetical example. So if it is in the district of say for or see this state of UP you may consider, so obviously travelling from one city or to other city for the fire engines in not possible but I am just giving an example, it can be the city of Allahabad, Banaras, Kanpur and so and so forth.

So the cities are marked as city 1, 2, 3, 4, 5, 6 along the first column and along the top most row obviously you have also the city, so this is city 1, 2, 3, 4, 5 till 6. Now, as it is given and we will okay, another important thing we will consider that city as points only such that any movement within the city would basically take 0 number of minutes or 0 number of seconds (we) which actually practically is not possible but we will still consider for our sake as an assumption which is true.

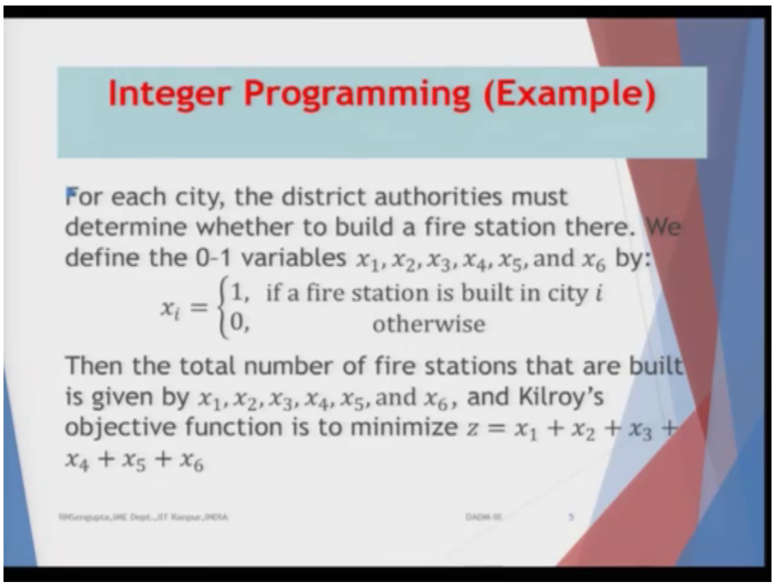
So if you are travelling from city 1 to city 1, so obviously you will see the timing is 0, so if I follow the concept that if you are travelling from city 2 to city 2 again time is 0, city 3 to city 3 time 0, 4 to 4 0, 5 to 5 0, 6 to 6 is 0. Which means the principle diagonal is 0, that means you do not take any time if considering a fire station in that city would not take any time for the fire engine to reach that emergency area point1. Point number 2, we are considering the time so the times obviously are given, so if you are considering that I am trying to travel like it can be hours, minutes whatever it is. But I am just writing the values so from city 1 to city 2 is 10, city 1 to city 3 is 20, city 1 to city 4 is 30, 1 to 5 is 30 and 1 to 6 is 20.

Similarly if I have the values it is, 2 to 1 is 10, now the important fact is that the time taken to travel from point A to B or from point B to A is the same. That means we have the same roads, same roads condition, the direction of the traffic is such that there is no in hindrance if it is traveling from A to B or B to A. It may be possible say for example during emergency that if you are going from city 1 to city 2 you have to take route number1 and if you are travelling from city 2 to city 1 you have to take route number 2.

In that case the distance travelled, the time taken to travel would be different but here we are not going to consider that. Which means the matrix which you have for our consideration considering the time frame or the time taken to travel between cities is a symmetric matrix, the principle diagonal is 0 and all the of the diagonal elements are equal. So let us check 20, 20 that means from 3 to 1 or 1 to 3. This is 6 to 2 or 2 to 6 is 10 okay here it is there are some difference right I just missed it sorry for that.

So here the timing given is, let me consider this is 25 okay-okay this is same. So this is 10-10, this is 20-20, this is 30-30 my mistake it is same, it is a symmetric matrix. Because it was placed in such a longitudal manner it was difficult to understand. So you have 30-30, you have 20-20, that means travelling from 1 to 6, 6 to 1 is same is 20. Then travelling from 6 to 2 is 10 and travelling from 2 to 6 is basically 10. Travelling from 4 to, say for example 4 to 5 is 15 and travelling from 5 to 4 is 15, so it is a basically symmetric matrix and all the times are given so they are basically in minutes or seconds whatever it is and the values are given.

(Refer Slide Time: 11:43)



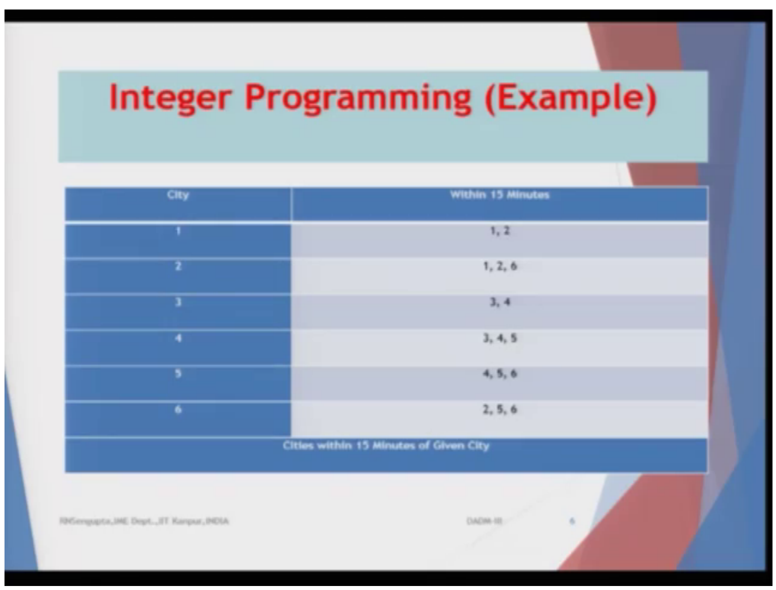
Now, for each city the district authorities must determine whether to build a fire station there or not. So, obviously if a fire station is built you invest money, you try to basically build up the building, employ people, if the city has to employ or the district administration or the authorities have to employ and they would be fire engines in that particular city and the station and if you do not, obviously you do not invest any money. So obviously it is a yes or a no decision.

So let me continue reading it for each city the district authority is must determine whether to build a fire station there, we define that as 0 1 variable, so if you remember in the case of an integer programming we said the values of the decision variables for integers that means they can be 0, 1, 2, 3, 4 whatever. So in this case they are 0 and 1 actually it is known as 01 programming but we will consider that they are integers, so it is absolutely possible to handle that in the realm of integer programming.

Now we define the variables which is 0 and 1 as x1, x2, x3, x4 and x5. So as there are 6 cities, it is there are 6 decision variables so x1 is 1. If a fire station is built in the city 1 otherwise 0, similarly x2 is 1 if a fire station is built in city 2 or else 0. Similarly x3 is 1 or 0, x4 is 1 or 0, x5 is 1 or 0 and finally x6 is 1 and 0. So in this case say for example there as fire station in all the cities. So all the decision variables starting from x1 to x5 are all 1. Consider if say for example you have fire stations in only the odd cities that means x1 or city 1, city 3, city 5 so x1, x3 and x5 are 1 while x2, x4, and x6 will be 0.

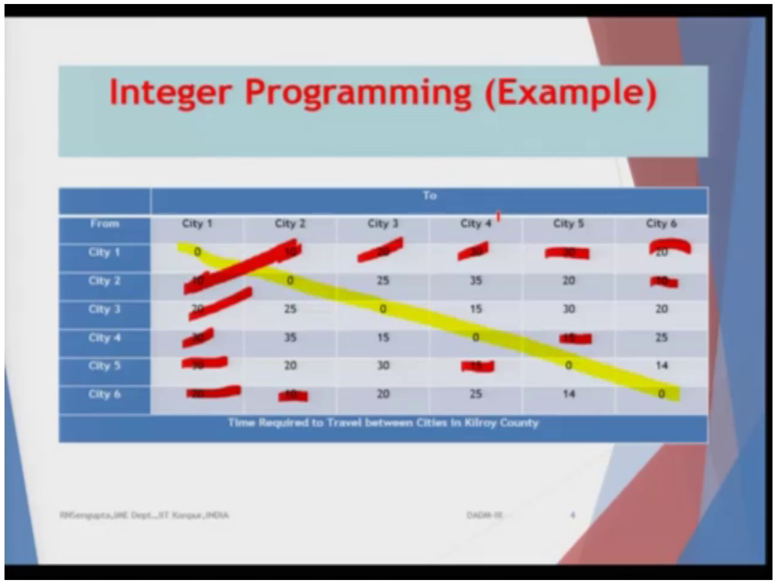
Then the total number of fire stations in case you have they are being built, so obviously you will have the decision variables as x1 plus x2 plus x3 plus x4 plus x5 and x6 is 0. So hence the total number of fire station is sum of all them which should be technically 6 in case if you are considering the 0 1 variable concept. Now, considering the matrix, so we had already the matrix where the times were given.

(Refer Slide Time: 14:15)



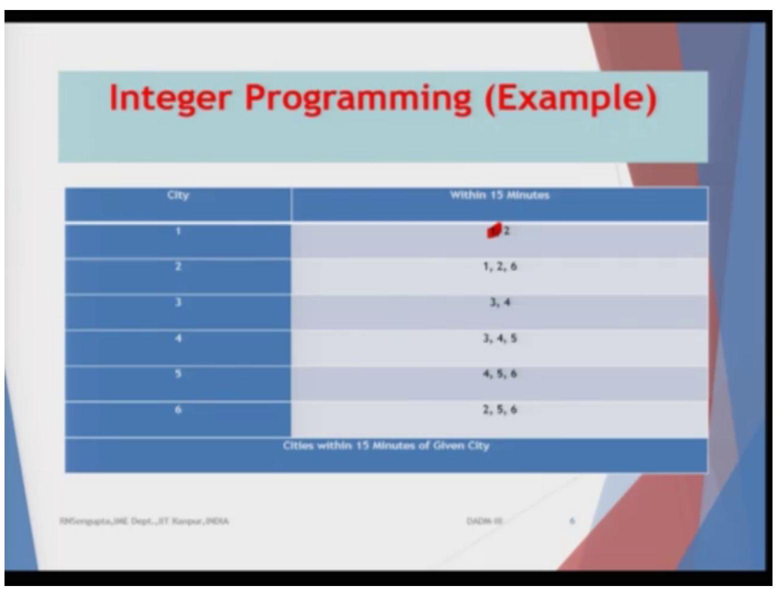
Now, I am considering that how many of them are within 15 minutes for each city, so obviously we will go in this direction. From city 1 the cities obviously 1 is within 15 minutes to itself so because the value at the time taken to travel from city 1 to city 1, or 2 to 2, and within the Jth city to Jth city were all 0, so obviously they would be included. So city 1 to city 1 is included hence 1 is there. Now if you see the table I will just go back to the table once.

(Refer Slide Time: 14:44)



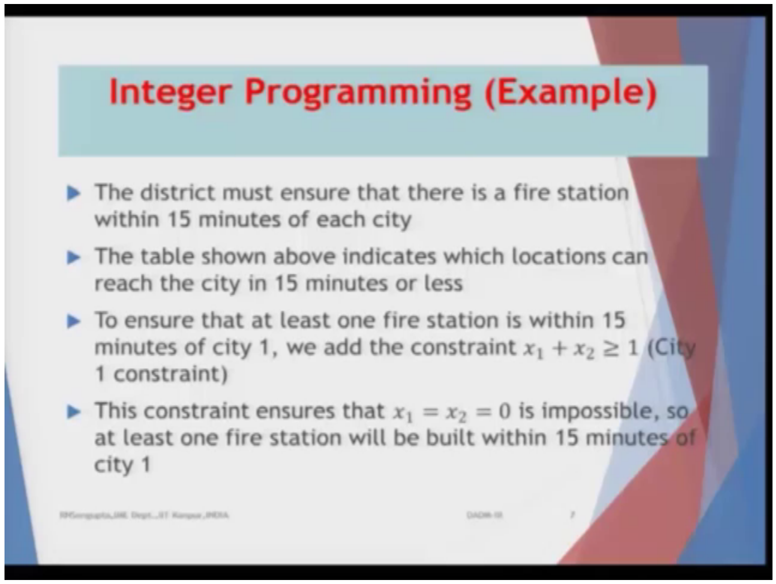
So consider city1, 1 is there, 3 would not be there, 4 would not be there, 5 would not be there, 6 would not be there only city which is plausible from 1 is 2. So obviously in the set where you can reach from 1 is 1 and 2, similarly when I go to 2 as it is a symmetric matrix, 1 is also included, 2 is also included, 3 no, 4 no , 5 no, 6 yes. So let us see for 2 it is 1, 2 and 6.

(Refer Slide Time: 15:15)



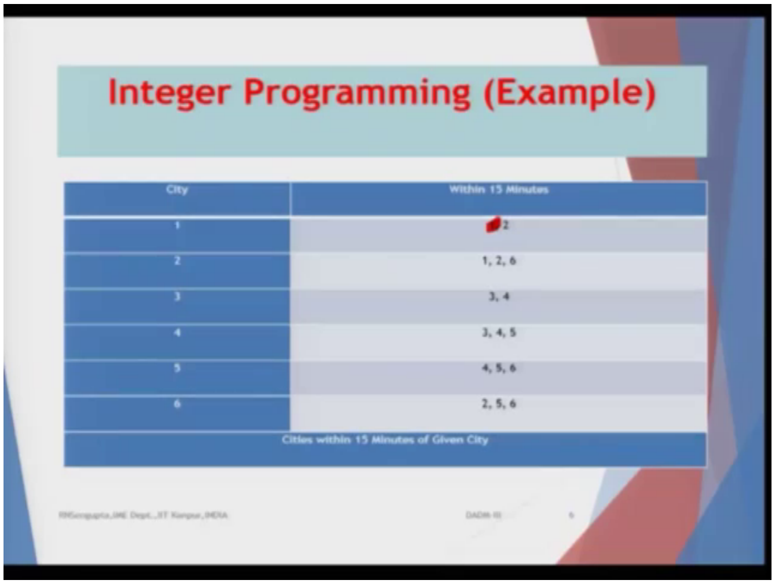
So we see as it is rightly we found out within 15 minutes you have basically city 1 to 6 from city 2. Similarly when I go and consider from cities 3 perspective, 3 would be there, 4 is there from cities 4 perspective, 4 is obviously there , 3 and 5 is also included. From cities 5 perspective 5 is obviously there 4 and 6.And finally from city 6 you have basically 2, 5 and 6 included in that set. Set means that consideration set.

(Refer Slide Time: 15:48)



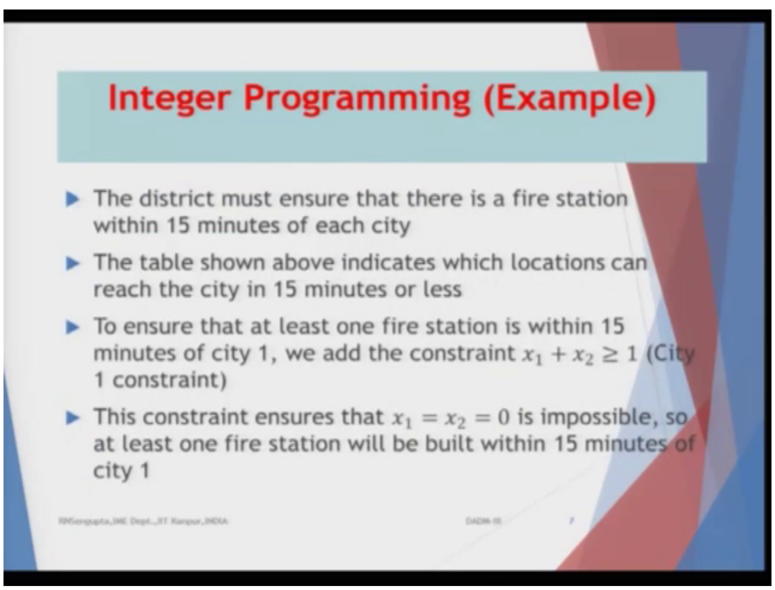
Now the district authorities ensure that there is a fire station within 15 minutes of each city that was one of the main assumptions. The table shown above which we had just indicated which locations can be reached to that particular city if you are basically starting from city 1 so the only cities which you can be reached is 1 and 2.

(Refer Slide Time: 16:12)



So let us see if you have a fire station in city1 it can cater to the needs of the emergency in city 1 plus city 2. If a fire station is in city 5 it can cater to the need of 4, 5 obviously and 6. Similarly if it is in city 6 it can cater to the needs of emergency for city 2,5 and 6.

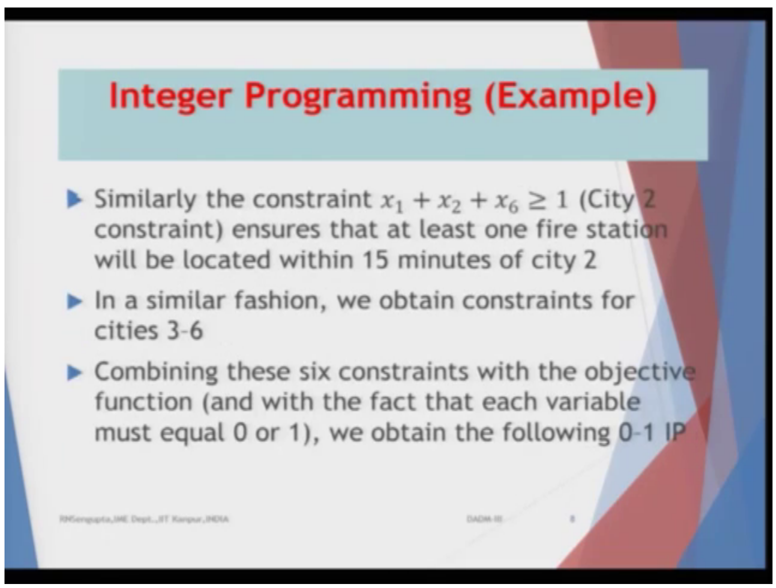
(Refer Slide Time: 16:28)



So the table shows that about indicates which locations can be reached from that particular city in 15 minutes or less. Now we will basically bring in the constraints such that we try to basically minimize the number of such fire stations in totality. So we will go from city 1 to 2 to 3, 4 and 5, 6 so and so forth , it can be done in any order but once we basically included 1 city it would not be considered in the next phase, I will come that the point which I just mentioned. To ensure that at least 1 fire station is within 15 minutes of city 1 we basically add that constrain that for city 1 the constrain is x1 plus x2 is greater than 2. So obviously it would mean that there is 1 fire station within 15 minutes of city 1.

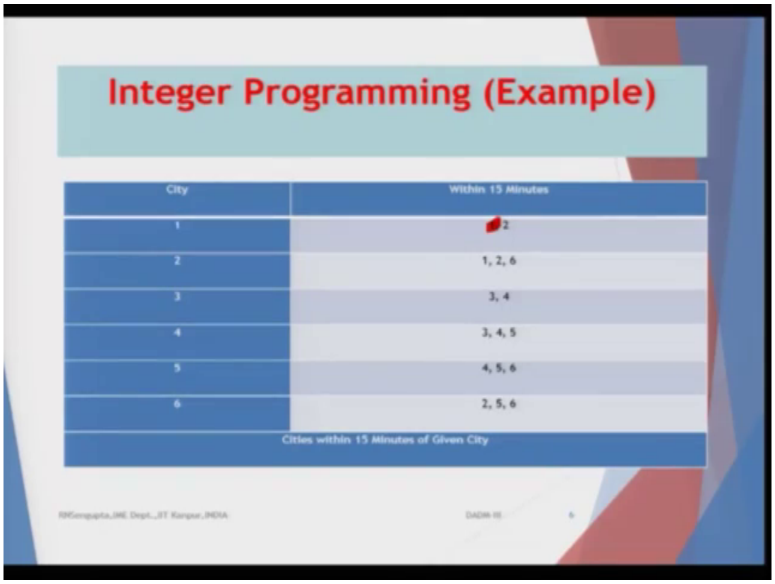
And we are able to basically cater to that with that constraint I will explain that in more details later on. This constrain which is I have just discussed ensures that if x1 is 0 or x2 is 0 it is not possible because 0 plus 0 cannot be greater than 1. So at least 1 fire station has to be built which is within 15 minutes of city 1 because basically it caters to 1 and 2.

(Refer Slide Time: 17:54)



Now, let us go to city 2 okay, now you may be thinking why I am considering x1 an x2?

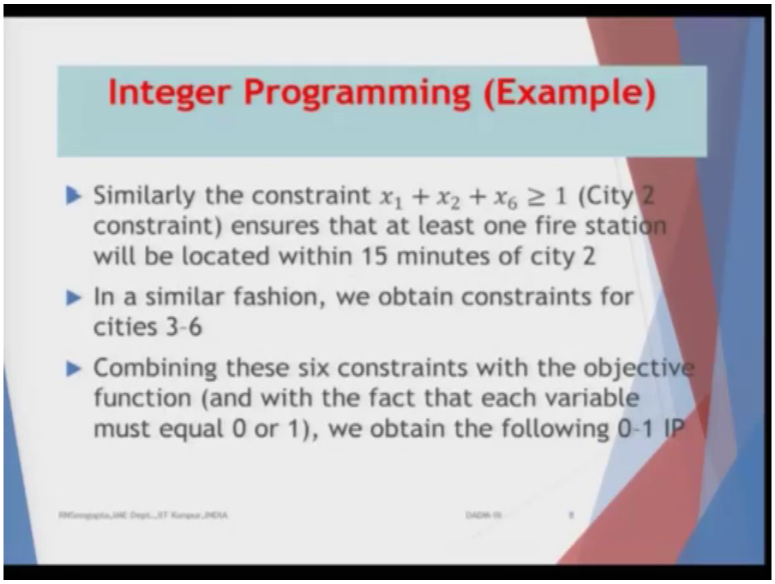
(Refer Slide Time: 18:01)



Let us go back, so if you see the first row it basically means city 1 is only able to cater 1 and 2, so obviously x1 plus x2 is greater than 1.Because either it its build in 1 so obviously 1 would basically be able to cater both the cities.

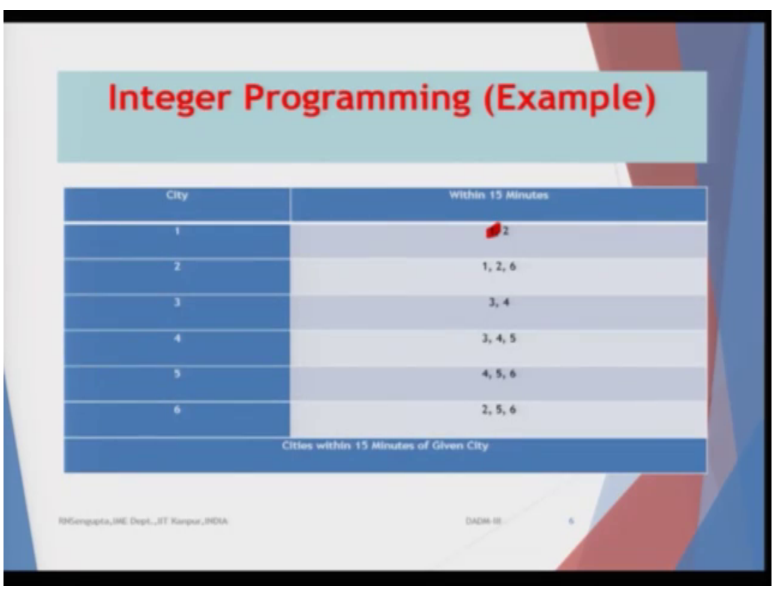
Technically, if no fire station is built in 2, still 2 is basically catered by the fire station which is built in city1. Now consider the second constraint, corresponding to the second constraint. It is given that 2 is able to cater to 1, 2 and 6 which means x1 plus x2 plus x3 would basically have a constraint such that if only 1 of the fire station is build in city2 it is able to cater 1, 2 definitely and 6 also. So let us see that constraint.

(Refer Slide Time: 18:53)



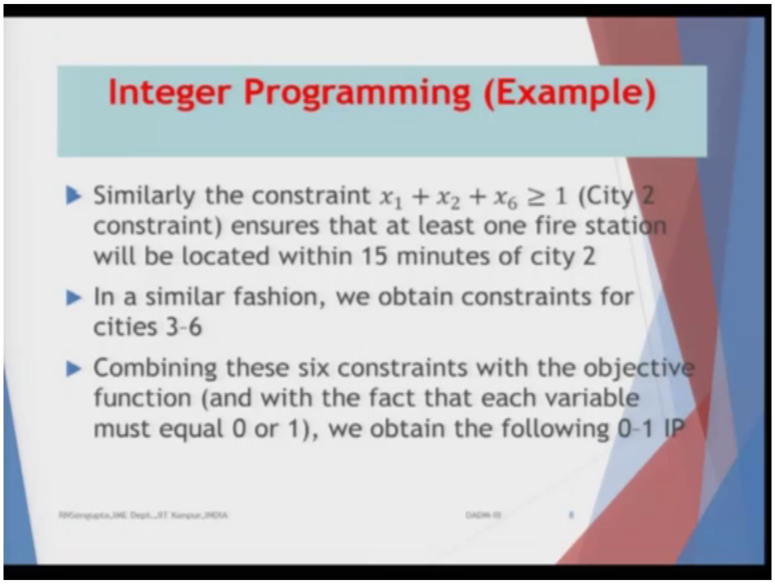
Similarly the constraint corresponding to city 2 would be x1 plus x2 plus x6 is greater than 1, it ensure that at least 1 fire station will be located within 15 minutes from city2 so as that Is able to cater to 1, 2 and 6 those are the cities. In similar fashion we can obtain the constraints for all the rest cities 3, 4, 5, 6.

(Refer Slide Time: 19:19)



Let us see, so 3 would be x3 plus x4 is greater than 1. For the fourth city it would be x3 plus x4 plus x5 is greater than equal to 1. For the fifth city is x4 plus x5 plus x6 is greater than equal to 1 and for the last city which is the sixth city it will be x2 plus x5 plus x6 is greater than equal to 1.

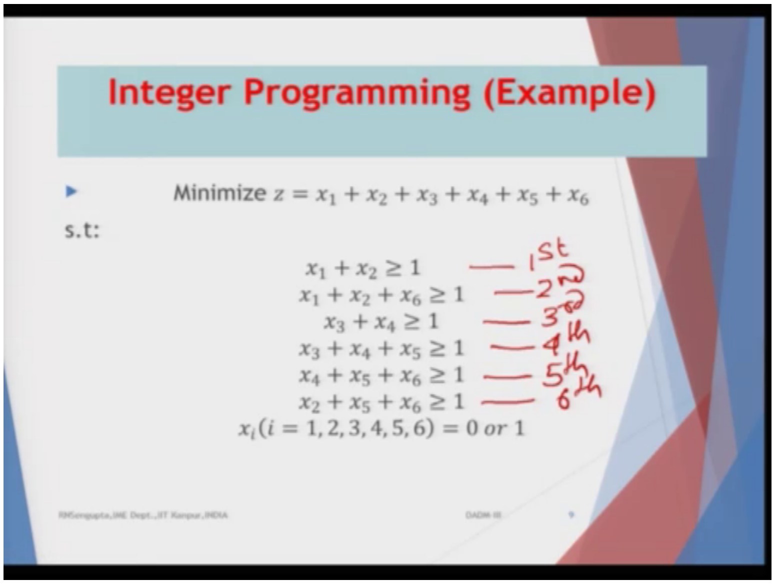
(Refer Slide Time: 19:42)



Now we need to combine, so we will combine all this constrains but what, if you remember what I am sure we have not forgotten the objective, objective was basically to ensure that z which is equal to the summation of x1, x2, x3, x4 and x5 was minimum. That means if only one, if it is possible that we have only one city which caters to all, if it was possible then obviously we would have basically minimize the cost. So combining this 6 constraints with the objective function and with the factor that each variables must be 0 and 1.

So, obviously x1 can be 0 1 either a fire station build or not build, similarly for x2 or x3 or x4 or x5 and x6. Combining these 6 constraints with the objective function and with the fact that each variables must equal to 0 and 1, we obtain the 01 integer programming as follows.

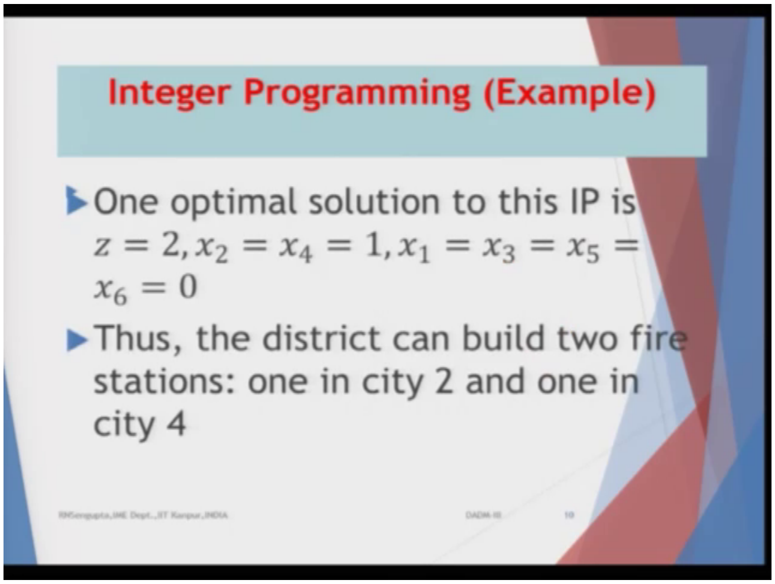
(Refer Slide Time: 20:37)



Now it so once I have mentioned that let me basically state the objective function we know. We want to minimize z which is the sum of x1 plus x2 plus x3 plus x4 plus x5 plus x6. Such that constraints are corresponding to the fact that the minimum time taken to reach any one city from the other is 15 minutes , so obviously I will just mark this is corresponding to the 1st city, this is to the 2nd city, first city was x1 plus x2 is greater than equal to 1.

Similarly x1 plus x2 plus x6 is greater than equal to 1 is for the second, this is the third, the fourth, this is too bold the marker I will just remove it and then come to the point in marker. So, this was the first city, this is the second city, this is the third city, this is fourth, fifth and sixth, these are the constraints and obviously xi, where i is equal to 1 to 6 would be either 0 and 1. So, because either you build a station or do not build a station.

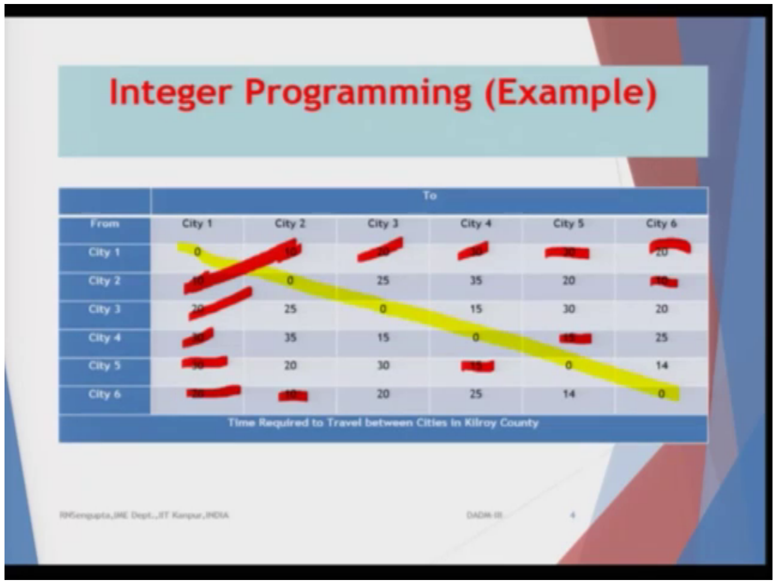
(Refer Slide Time: 21:54)



Now, once you solve it, I am not going to the coming to the solution because the solution methods have already been discussed considering the Gomory cut algorithm and the branch and bound, branch and bound I will come to that later please have patience. So, once your optimum solution is solved you find out that z is 2 and what are the solutions you have, you do not have any fire station in city 1, you do not have any fire stations in city 3 or city 5.

Only the fire stations are to be built is at city 2 and 4. So hence x2 is equal to x4 is equal to 1, while x1, x3, x5 and x6 are 0. Thus the district can build 2 fire station, 1 in city 2 and 1 in city 4.

(Refer Slide Time: 22:42)



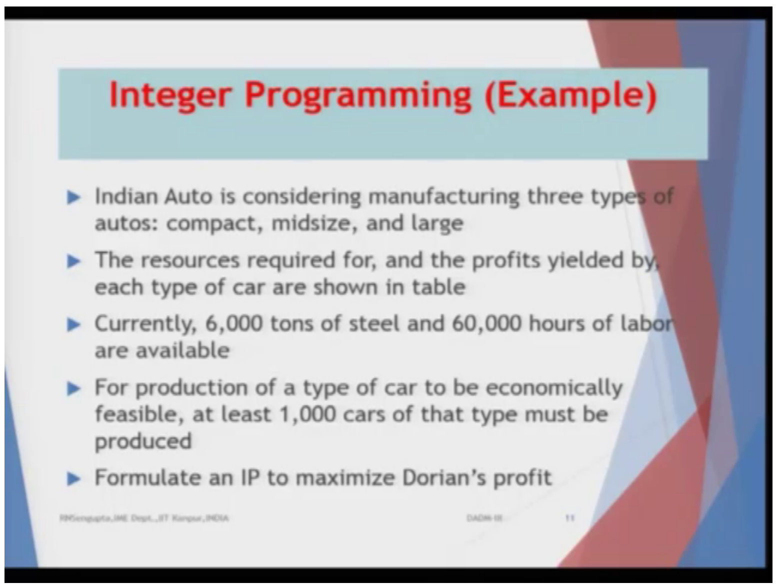
Now, if I go back to the problem, say for example now we make the constraint that a city has to be reached within 10 minutes, so in that case the first, the set of cities which are minimum 10 minutes it cannot exceed 10 minutes. So in that case, the number of cities in case of city 1 would be city 1 and 2 obviously the same because 15 and 10 would not make a difference, in case of city 2, now you will basically have 1, 2 and 10. In case of city 3 you will basically have only 3 because this 15 is now ruled out because you have to reach that particular city within 10 minutes.

For city4 you have only city4, city 5 you have only city 5 because these 15 are ruled out and city 6 you have only city 2 and city 6. So obviously the constraints will change incase if you are trying to basically build up that the fire station has to reach that particular city within 20 minutes and less. Then again the constraints change, constraints changing would be basically the first one will be if you have 20 or less it will be x1 plus x2 plus x3 plus x6 is greater than equal to 1. because 4 and 5 are ruled out,

For city 2 the constraint, the second constraints will be x1 plus x2 plus x6 is greater than equal to 1. For the third city it will be as I mentioned is 20 minutes, so obviously it will become x1 plus x3 plus x4 plus x6 is greater than equal to 1. For 4th city, I am just repeating it please bear with me, for the 4th city it will be x3 plus x4 plus x5 is greater than equal to 1. For the 5th city it will be x2 plus x4 plus x5 plus x6 is greater than equal to 1. And for the 6th city it is basically x1 plus x2 plus x3 plus x4 plus x5 plus x6 is greater than equal to 1.

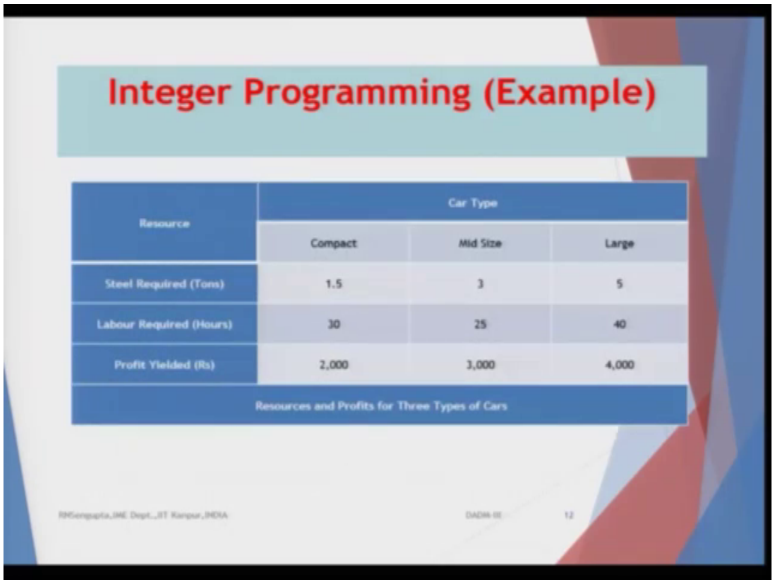
Obviously we will have the same objective function, sum of x1 till x6 is to be minimized and then when you solve the problem you get a different answer. So this is the only the constraint will change so do not worry on about all this things okay. And coming back to there, this are greater than 1 obviously we will use the same techniques I am sorry to repeat that you will use the same techniques like add the slag and surplus , solve the linear programming add the Gomory cuts or do the concept of branch and bound and then proceed accordingly.

(Refer Slide Time: 25:23)



Let us consider a secondhand example; Indian Auto is considering manufacturing three types of autos, compact, midsize and large. The resource required for, and the profits yielded by each type of car is shown in a table I am going to come to that. Currently 6000 tons of steel and 60000 hours of labors are available which is under the jurisdiction of Indian Auto. For production of a type of car to be economically feasible at least 1000 type cars of that type must be produced. We need to basically formulate the problem as an IP maximization problem for the profit.

(Refer Slide Time: 25:56)

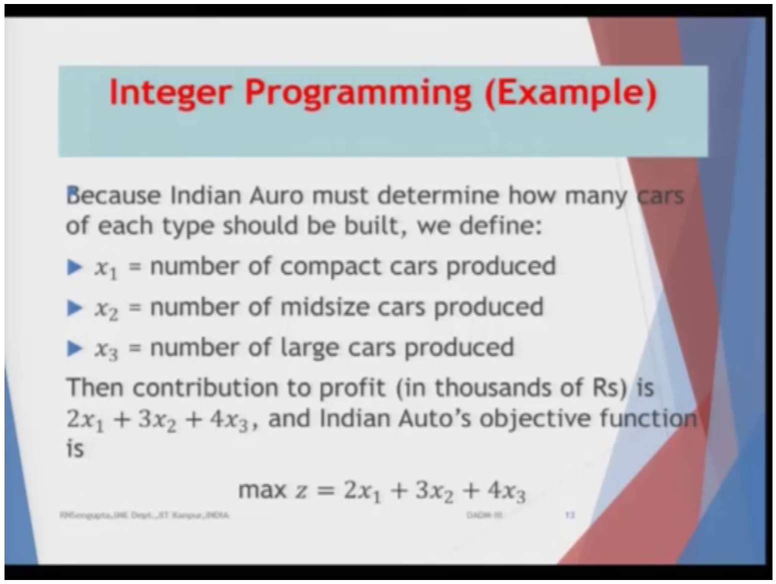


Resource	Car Type		
	Compact	Mid Size	Large
Steel Required (Tons)	1.5	3	5
Labour Required (Hours)	30	25	40
Profit Yielded (Rs)	2,000	3,000	4,000
Resources and Profits for Three Types of Cars			

So the resources are given as this, so the car types are given there are 3 car types compact, midsize and large. So it means the smaller cars, mid value car and the larger cars. Profit yielded in rupees I am considering I am only reading the last row now, profit yielded in rupees are given even though those values are practically not very feasible but still I will consider those values, it is 2000 for compact, 3000 for midsize and large are 4000. Labor cost are given for compact, midsize and large are 30 hours, 25 hours and 40 hours.

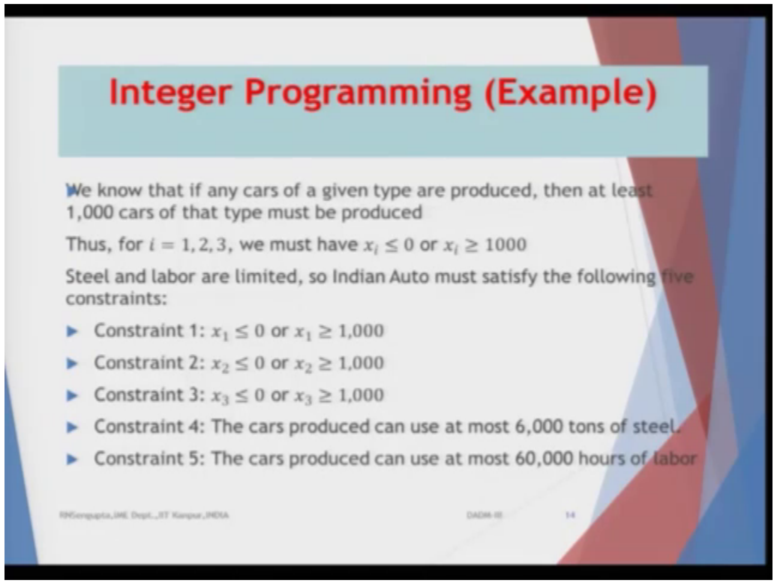
And the steel requirements are given as 1.5, 3 and 5 tons respectively for the compact, midsize and large cars. So the resource and the profit for the 3 cars are given.

(Refer Slide Time: 26:52)



Now because Indian Auto must determine how many cars of each type should be built, we define the following variables. X1 is the number of compact is the variable and obviously they would be integers they cannot be continuous, it cannot produce 2.57 cars or it cannot produce 100.792 cars. X2 is the number of midsize cars produced, X3 is the number of large cars produced. Then hence, the contribution you will basically have is basically, if you have 22000 from the small, 3000 from the middle, and 4000 from the big cars. The total maximization problem would be in thousands, it will be 2000 into x1, plus 3000 into x2, plus 4000 into x3.

(Refer Slide Time: 27:42)



Now, we know that if a car of a given types is produced then at least 1000 types are to be produced and we will consider that x1 is cannot be less than so we should basically have that the number of cars produced has to be greater than equal to minimum at least 1000. And the constraints if you consider that steel and labor are limited so Indian Auto must satisfy the following constraints.

So constraint 1 is either you produce or you do not produce, so obviously in that case x1 is greater than 1000, x2 is greater than 1000, x3 is greater than 1000. Constraints 4 is that car produced can use at most 6000 tons of steel totality and the car produced must in totality used 60000 a number of man hours. So with this I will stop 41st lecture and continue more discussion about the integer programming solution in the 42nd and lectures and so and so forth. Have a nice day and thank you very much.