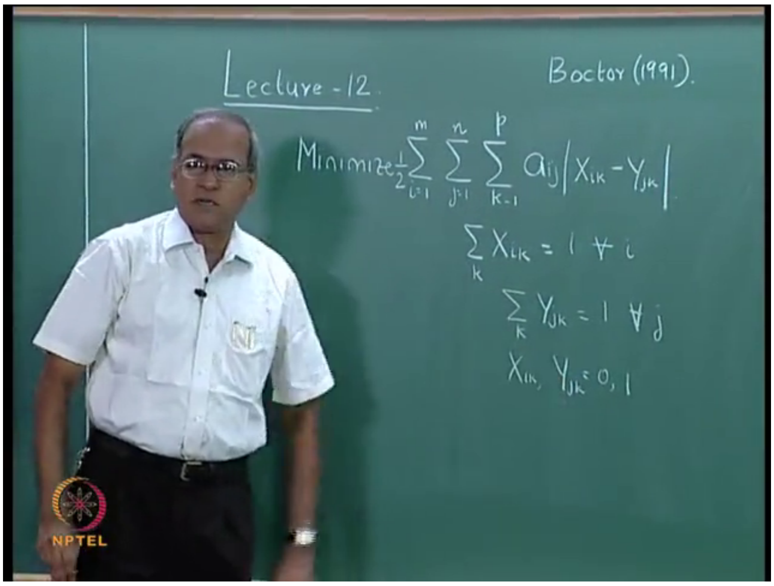
Manufacturing Systems Management
Prof. G. Srinivasan
Department of Management Studies
Indian Institute of Technology, Madras

Lecture – 12
Optimization based algorithms, Assignment based algorithm

In this lecture, we continue the discussion on optimization based approaches to cell formulation. In the previous lecture, we started a formulation that minimizes i intercell moves, and the formulation shown in the picture.

(Refer Slide Time: 00:29)



The objective function is to minimize. So, this is the objective function given there. i is equal to 1 to m represents a number of machines, j equal to 1 to N represents the number of parts or components. K equal to 1 to p would represent the number of groups. So, in this formulation p which is the number of groups is known and fixed; which is not a variable it is fixed.

Now, X i k, i equal to 1, if machine i goes to group k. And Y j k equal to 1 if part or component j goes to group k. So, each machine will have to go to only one group. So, X i k equal to 1 summed over k for every i. For every machine i it has to go to only one group. For example, if we are considering 2 groups, then we will have a const and if there are 6 machines, then we will have 6 constraints here one for each machine, for every i. The constraints will be of the type X 1 1 plus X 1 2 equal to 1 X 2 1 plus X 2 2 equal to 1 and so on.

Similarly, each part should go to only one cell. So, summed over k Y j k equal to 1 for every j. So, if there are 8 parts then, Y 8 1 plus Y 8 2 equal to 1 Y 1 1 plus Y 1 2 equal to 1 Y 2 1 plus Y 2 2 equal to 1 and so on.

So now X i k and Y j k are binary; which means either take value 0, or they take value 1. Now we also explained how this one would compute the intercell moves. Now an inteercell move occurs if part j visits machine 1, and part j and machine i are allocated to different groups. So far, an intercell move to happen there has to be a move which means a i j should be equal to 1. So, if part j does not visit machine 1, then there is no scope for an intercell move at all. So, this a i j should take a value 1. And the group to which machine i is assigned and the group to which part j is assigned have to be different. If there is a visit a i j equal to 1, and if machine i and part j are assigned to the same group k, then both these will be 1. And therefore, the absolute value of the difference will be 0. There would not be an intercell move. There will be an intercell move if either this is one and this is 0, or the other one is 0 and this is 1.

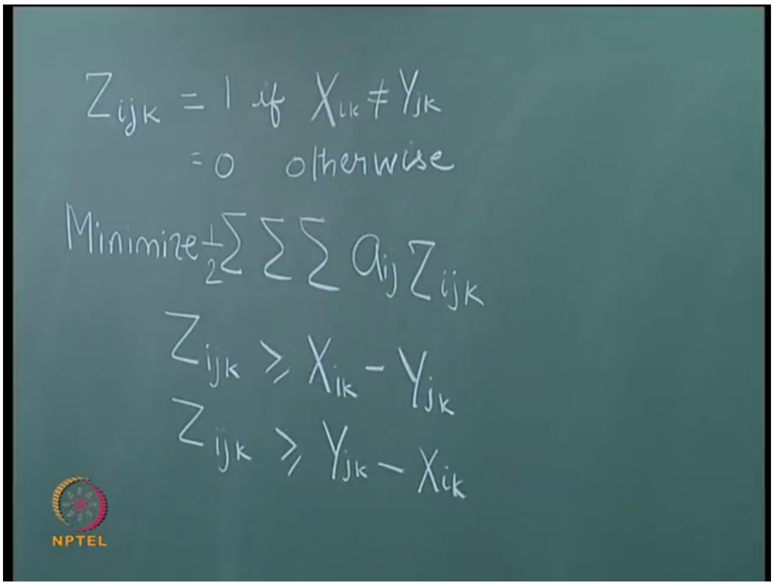
Therefore, this will compute the intercell moves. The only difference between what is shown here and what is shown in the slide is that there is a half in the objective function. The half comes because it is counted twice due to the summation i, i it will be counted twice therefore, we get a half. So, we could write minimize half into this. Now the 2 constraints are each machine goes to only one cell, and each part goes to only one cell. The important aspect is that p which is the number of groups is known and fixed it is not a variable. The other nice aspect of this formulation is that the objective function directly minimizes the intercell moves, or the number of intercell moves. Whereas p median minimize a distance. Distance computed using the formula that we have defined.

Indirectly measures the intercell move. 2 machines will be assigned to the same group in the p median if the distance is less. The distance will be less if parts required those 2 machines, or the row vectors corresponding to those 2 machines, if they are identical then the distance is 0. If they are not identical then there could be a distance. And lesser the distance, lesser the dissimilarity and lesser the intercell moves. So, minimizing distance or maximizing similarity, which where the objectives of p median, indirectly try to reduce the number of intercell moves.

So, one of the main objectives of cellular manufacturing at the cell formation stage, which is to minimize intercell moves; this modeled explicitly in this formulation as an objective function. Where as in the p median it is modeled indirectly as the objective function. So, the advantage is that the very specific objective of minimizing intercell moves at the cell formation stage is modeled directly and explicitly as the objective function, which is big strength of this formulation. As mentioned earlier this formulation is given by Boctor in the year 1991. Now this is a binary formulation, because variable state value 0 1.

Now, one of the disadvantages of this formulation is that, it has an absolute value in the objective function. And therefore, we cannot directly put it into a solver and try and get the solution. So, we need to convert this absolute value terms into a form where it can go directly into a solver. So, to do that we now we introduce some more variables and some more constraints.

(Refer Slide Time: 07:42)



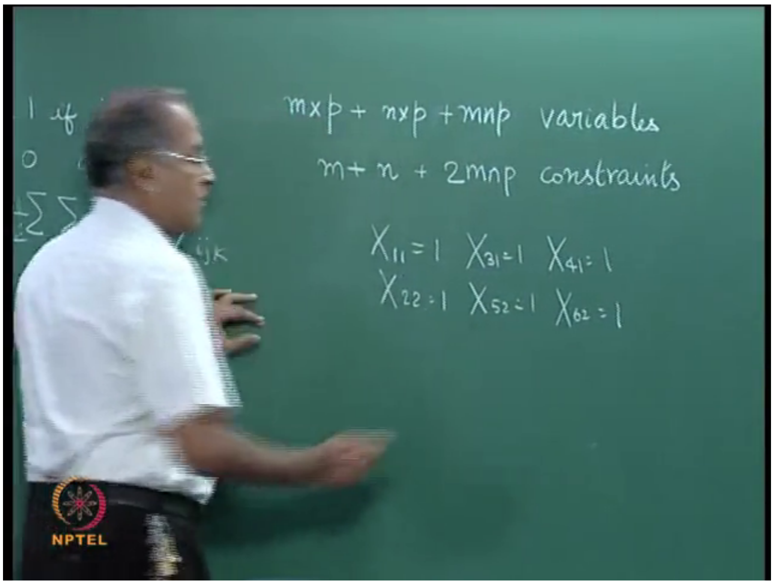
So, we will introduce a Z i j k, a Z i j k which is the binary variable. And this Z i j k will be equal to 1. If X i k is not equal to Y j k, and it is equal to 0 otherwise. So, X Z i j k will effectively substitute the role of the absolute value function of X i k minus Y j k.

Because if X i k equal to 1, and Y j k equal to 1 for a particular i j k, then the absolute value will be 0. So, here this here X i k will be equal to Y j k therefore, this will get 0 value. The intercell move will be created only if there is an a i j. So, write now we take out that a i j assuming that a i j equal to 1; which means the part visits the machine. And the part and machine are in different groups. So, X i k and Y j k are different; which means one of them is 0, the other one is 1. So, if X i k is not equal to Y j k and if a i j equal to 1 there is an intercell move. So, Z i j k is defined as equal to 1 if X i k is not equal to Y j k. If both are 1, then this will be 0, if both are 0 then this will be 0. If one is 0 and the other is 1, then it will be 1.

So, the objective function now will become minimize a i j Z i j k. Now we have change the objective function, but then we have to relate this Z i j k to X i k and Y j k. Now that is given by these 2 sets of equations which are given here. So, Z i j k is greater than or equal to X i k minus Y j k. Z i j k is greater than or equal to Y j k minus X i k. Now what happens? We can easily check that, if both are them are equal if both are 0 this will be 0 this will also be 0. So, Z greater Z i j k greater than or equal to 0, Z i j k greater than or equal to 0, minimization. So, it will take a value 0. If one of them is 1; for example, if this is 1 and this is 0, then this Z i j k greater than or equal to 1, Z i j k greater than or equal to minus 1, and since Z i j k is binary it will take a value plus 1.

If this is 0 and this is 1, then Z i j k greater than equal to minus 1, Z i j k greater than equal to plus 1, and since Z i j k is binary it will take a value plus 1. So, Z i j k will effectively replace, this term and the relationships are given here. So, the objective the actual formulation this is the objective function, subject to this this this and this. So, we have to check the number of variables, and the number of constraints to get a feel of the size of the formulation. At the same time, we are aware that X i k Y j k and Z i j k are binaries.

(Refer Slide Time: 12:25)



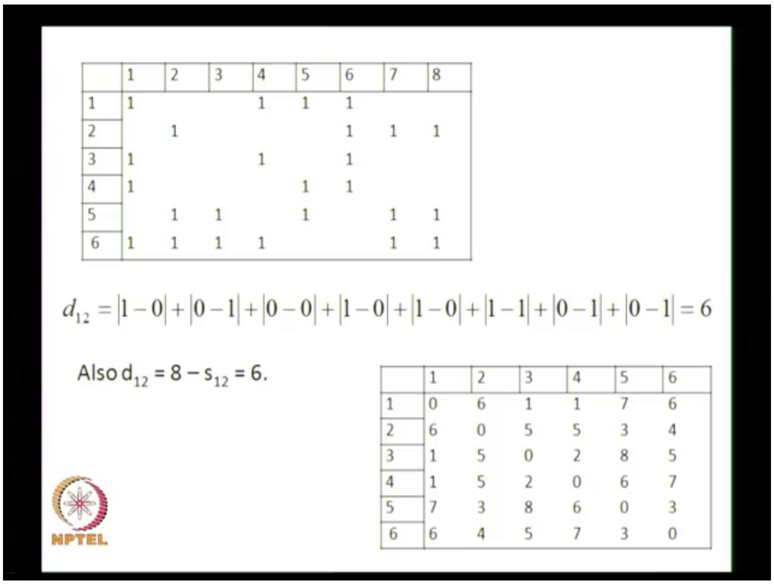
So, it is a all 0 1 problem with now how many X i k there are m into p variables. So, X i k variables are m into p, Y j k N into p, j equal to 1 to N n parts, p equal to k equal to 1 to p p groups, plus Z i j k is m N p. So, we have that many binary variables. The number of constraints there are m constraints there are N constraints. So, m plus N, m plus n. And for each of every X i k and Y j k, for each of this there is an absolute value function. And there are 2 constraints for each one of these.

So, this will contain 2 times for each Z i j k there are 2 constraints. So, there are m into N into p Z i j k. So, 2 times m and p constraints.

Because each Z i j k results in 2 constraints now earlier. Without the Z i j k, it had a very small number of variables and constraints, but more importantly because we have to convert the absolute value term into linear constraints. We have a large number of constraints.

Now, in this form with this objective function, with this this this and this. We can solve this problem to try and get the optimal solution for a fixed number of groups to minimize intercell moves.

(Refer Slide Time: 14:48)



	1	2	3	4	5	6	7	8
1	1			1	1	1		
2		1				1	1	1
3	1			1		1		
4	1				1	1		
5		1	1		1		1	1
6	1	1	1	1			1	1

$$d_{12} = |1-0| + |0-1| + |0-0| + |1-0| + |1-0| + |1-1| + |0-1| + |0-1| = 6$$

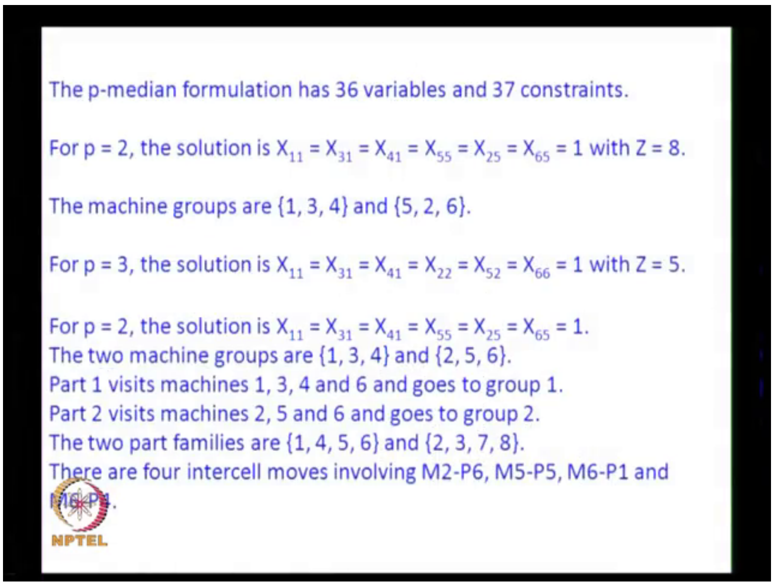
Also $d_{12} = 8 - s_{12} = 6$.

	1	2	3	4	5	6
1	0	6	1	1	7	6
2	6	0	5	5	3	4
3	1	5	0	2	8	5
4	1	5	2	0	6	7
5	7	3	8	6	0	3
6	6	4	5	7	3	0

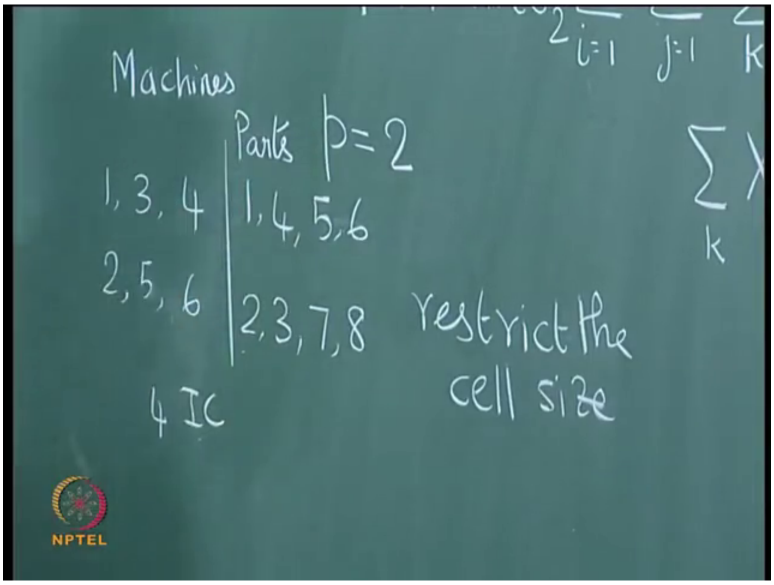
So, if we go back to our example that contains 6 rows and 8 columns which is shown on the first figure in the slide, now we can solve this to try and for p equal to 2 or p equal to 3 and we can solve this to get the optimal solution.

Now, let us look at what happens if we solve for let us say p equal to 2, if we solve for p equal to 2. Let us assume for a moment that we know a solution for p equal to 2 which we know for this matrix there are 4 intercell moves with. For the same matrix the machine groups are 1 3 4 and 2 5 6 with 4 intercell moves.

(Refer Slide Time: 15:45)



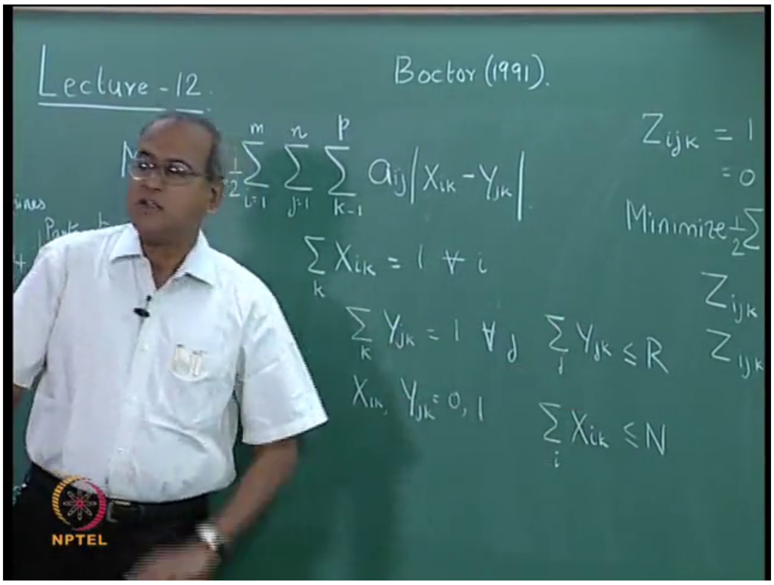
(Refer Slide Time: 15:55)



Let us assume that we know the solution at the moment. We do not know whether this is optimal or not, but we know the solution. Now let us consider we are solving for p equal to 2, let us put all the 6 machines into one group even though we are solving for even though we are solving for p equal to 2, if we put if we solve for p equal to 1, the solution would be all the 6 machines are in one group all the 8 parts will be in the in one group, and there will be 0 interscell moves. So, when we solve for p equal to 2, it will still do the same thing, by putting all the machines into one group, all parts into one group.

The second group will contain no machines and no parts, and it will give us a solution with 0 intercell moves. So, even if we increase p for a large sized problem, it will always put all the machines into one group, all the parts into one group. Rest of the groups will be null in the sense there would not be any machine are part side. So, it will give some kind of a trivial solution, the way this formulation is written. Therefore, we need to do something. So, what we do is we restrict the cell size we restrict the cell size. This important to restrict the cell size. Otherwise, it will put all the machines into one group.

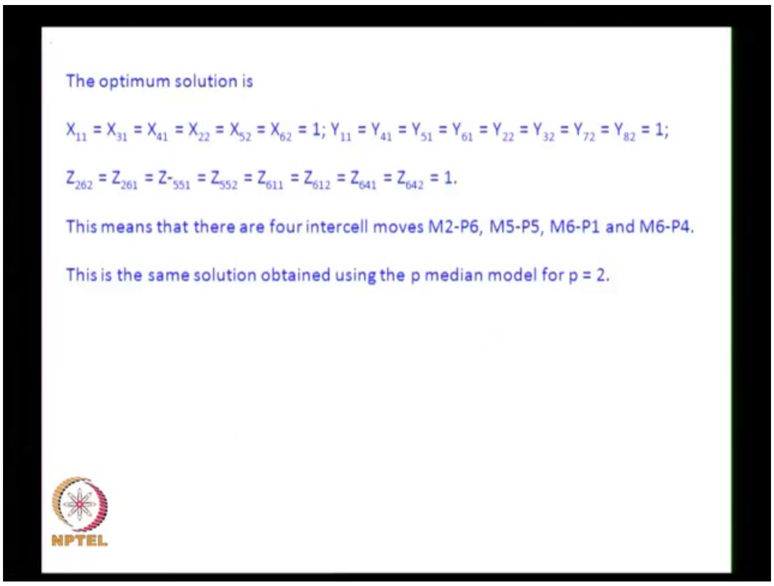
(Refer Slide Time: 18:00)



So, we should add constraints of the type sigma X i k summed over i will tell me the number of machines that are assigned to group k. So, this will be less than or equal to m. Now when I solve this problem with this additional constraint. Now I am restricting each machine cell to have a size of 3 capital N is the number of machines in a particular cell, suppose we restrict it to 3. So, it has to create more than one machine cell.

In that scenario, it will give a solution 1 3 4 2 5 6. So, the optimum solution with the addition of this constraint is X 1 1 equal to 1 X 3 1 equal to 1.

(Refer Slide Time: 18:56)



So, X 1 1 equal to 1. This means that machines 1 3 and 4 belong to the first group. 2 5 and 6 belong to the second group. And if we look at the part family solution, we get Y 1 1 Y 4 1 Y 5 1 and Y 6 1 belong to the first group. So, 1 4 5 6 will come here, and 2 3 7 8 will come here. 2 3 7 and 8 parts will come here, these are machines, these are parts. They will come to the second group and we will get the solution with 4 intercell moves. So, this formulation can be used provided we understand that we have to restrict the cell size.

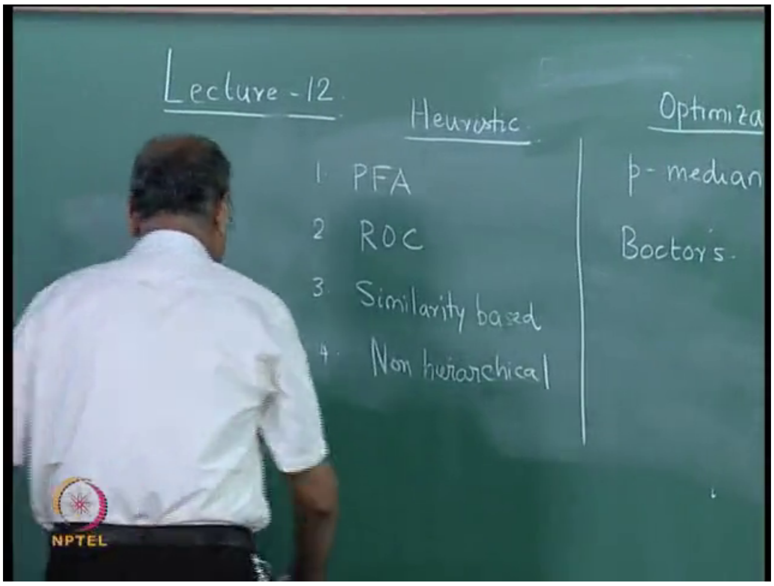
Now, the cell size can be restricted in terms of number of machines, or it could also be restricted in terms of number of parts that visit the cell. So, we could add constraints of this type which is sigma Y j k summed over j is less than or equal to sum r; where R is the maximum number of parts that you want in a cell. So, in in for this to be effective, 2 things have to be done. The number of groups has to be fixed, but when we fixed a number of groups we are essentially saying even though we sum it up to p we say that the number of groups will not exceed. P we does not say it is exactly p. And in this formulation, you do not have a way to say that it should be exactly p because p appears in the summation also.

So, we would not be would not be able to do that. Therefore, this will if it is advantages because of the nature of the objective function. Unless this restriction comes it will try and put everything into one cell, and give a solution with 0 intercell. So, cell size restriction has to be given. So, it is like a p comma N, or p comma N comma R, but normally p comma N only this restriction is used this can also be used this restriction is used. So, both represent some kind of an upper limit on the number of groups and the cell size. The other thing that can happen sometimes is that; if you are working on large sized matrices, and if you are working on matrices that that that have lots of intercell moves happening. Sometimes we can get into a situation where, if let us say we are solving for some p equal to 5 or 6 and some restriction on the cell size. So, let us assume there could be 6 machine groups, but there could be 5-part groups. That can happen. So, one-part group can become empty.

Sometimes some of these machine groups may have only one machine, and some of these part groups may have only one part. Then we get into a slightly different different kind of situation where do we want a cell that has only one machine. Or do we want a cell that processes only one part. If the answer is no then we also put restrictions of this type where we say each machine cell should have at least 2 machines, and each part family should have at least 2 parts. So, even though this is written as single constraint, there actually 2 constraints. So, this is the upper limit on the part family size, this is a lower limit on the part family size. Similarly, this is an upper limit on the machine cell size, and this is a lower limit on the machine cell size.

So, as we constrain the problem, the number of intercell moves will also keep increasing. So, this is a nice optimization based approach to solve this problem that minimizes the intercell moves directly, and places restrictions on the cell size. For p equal to 2, and the restriction that cell sizes less than or equal to 3, we get the same solution with 4 intercell moves. Now so far, we have seen a few methods for cell formation.

(Refer Slide Time: 25:06)



So, we have seen production flow analysis, we have seen rank order clustering. We have seen similarity based algorithm. We have seen a nonhierarchical clustering algorithm. And under the optimization, we have seen p median.

And we have seen Boctor's formulation. So, there are several other formulations, and several other algorithms that are also available for cell formation. In fact, the number of algorithms available is very, very large.

We can easily categorize this as optimization based algorithm. Because in both these we actually solve an optimization problem, an integer programming problem binary ip problem in both the cases. Where you have a clearly defined objective function, you have constraints and you have variables. Here we do not do that. Here we solve using algorithms which are essentially step by step procedures to get to a solution. While the optimality is guaranteed when you solve using these 2, the optimality is not guaranteed in this.

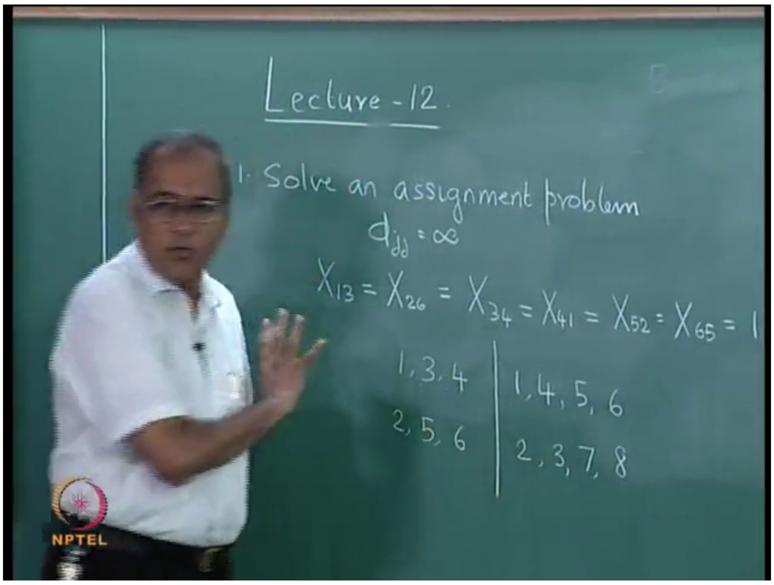
Many times, they are actually trying to solve the same problem of forming cells. But the objective functions are slightly different though they aim to represent the same, here also the way the alga each algorithm works is slightly different, but all the algorithms essentially try to do the same thing, which is either to minimize intercell moves or to put machines that are similar, similar in terms of part requirement into the same group. Now these can be called as optimization based algorithms, these could be called as heuristic algorithms. Algorithms that provide good solutions, algorithms that do not take too much of a computation time as the problem size increases.

We have not studied the computational requirements of these, but at the same time I would make the statement that these the computational time required the effort required, does not increase at a very alarming or exponential rate compared to the effort needed to solve these 2. So, heuristic algorithms provide good solutions. They do not guarantee optimality at every instance. At the same time, they provide algorithms that are based on some ideas which are very relevant to the problem. And provide good solutions, within a reasonable computational time and effort. Amongst or as perhaps one of the last algorithms for cell formation with binary matrices, we will look at one more algorithm, which is an algorithm based on an assignment problem.

Which draws some ideas from p median, and also draws some ideas from non-hierarchical clustering. It is still a heuristic algorithm, and not exactly an optimization based algorithm. Even though it solves an assignment problem. In order to be consistent, we use the same machine component incidence matrix to try and explain the algorithm. Now let us assume that we have the same 6 roweight column incidence matrix, where there are 6 machines and 8 parts. The dissimilarity matrix or distance matrix is given here below. Sample computation is also shown here d 1 2 distance between 1 and 2 is 6. We have already seen how that 6 is calculated.

So, we will assume that we have this distance matrix with us. Now the same distance matrix can be used as an input to solve a p median problem, which we have shown. Now let us take this distance matrix, and let us understand the basic idea of p median, particularly p median and a nonhierarchical clustering that we have already seen. P median and non-hierarchical clustering are the same except for the fact that p median chooses the best combination of seeds and allocates the rest of them to that seed. Non-hierarchical somewhere it assumed that the points themselves can be seed. So, the basic principle is we are assigning machines to machines. Because some of these machines act as seed points, or seed machines, and the non-seed points are assigned to the seed points. And the matrix the same input matrix is given to p median as well as to the assignment. Now the distance matrix or dissimilarity matrix is a square matrix. Because it is a it is computed for the machines, it is a square matrix, it is a 6 by 6 matrix. So, first thing we do is since it is a square matrix, and it is a distance matrix. We try to solve an assignment problem out of this.

(Refer Slide Time: 31:37)

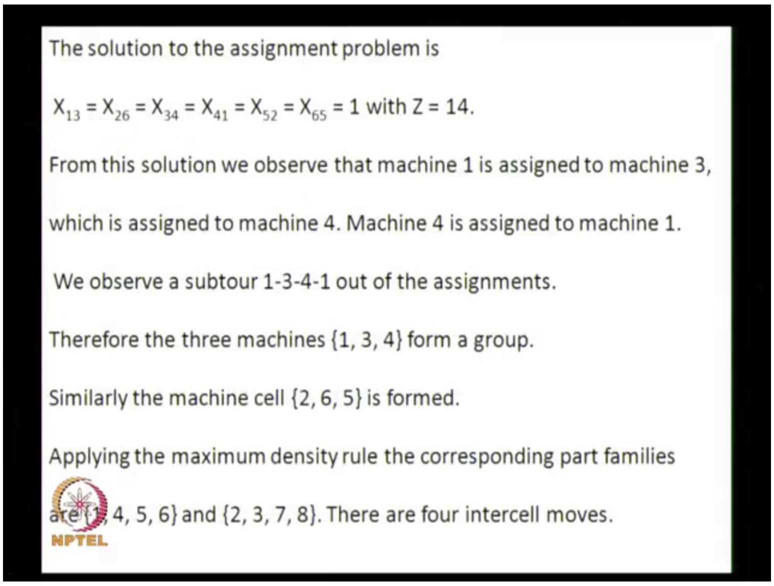


Solve an assignment problem out of this. Assignment problem is a well-known problem in operations research any first course in or we would have studied the assignment problem solve. Assignment problem generally assigns machines to people or machines to jobs or jobs to people and so on. Assignment problem input matrix is a square matrix.

So, we also have a square matrix here, it is also symmetric in our case. And we solve an assignment problem. That minimizes the total distance. Now the very structure of the matrix which is shown in that in that slide as a second table, has all the diagonals having 0. Because distance between any point and itself is 0. This distance matrix are dissimilarity matrix is constructed from a machine component incidence matrix. Which is also shown above. And I have already mentioned in an earlier lecture that each of these rows or machines can be thought always points in an 8 dimensional 0 1 space.

So, each of these 6 machines or rows represent 6 points. Therefore, distance between a point and itself is 0. So, all the diagonals are 0. So, when we solve an assignment problem with this as input. To minimize the total distance, it will give us a trivial solution with all diagonal assignments. It will say X 1 1 equal to 1, X 2 2 equal to 1, X 3 3 equal to 1, 6 6 equal to 1 with total distance equal to 0. Now we do not want that to happen. Therefore, we put all d j j to infinity. So, that we do not have diagonal assignments. So, an important thing is like a traveling salesman problem, we put diagonally equal to infinity. So, that we do not want diagonal assignments.

(Refer Slide Time: 34:13)



Now, let us change all of them to infinity, and let us solve an assignment problem to get a solution which is like this, X 1 3 equal to X 2 6 equal to X 3 4 equal to X 4 1 equal to X 5 2 equal to X 6 5 equal to 1, with minimum distance equal to 1 14.

Now, let us go back to the distance matrix, try to understand what we are trying to do through this solution. Now X 1 3 equal to 1 is here, and this tells us that we are assigning 1 2 3. So, this is assigning 1 2 3, and this distance matrix or dissimilarity matrix is between machine and machine. Unlike in a normal assignment problem where it is between machine and job or machine and people it is between machine and machine. So, X 1 3 equal to 1 means I am assigning machine one to machine 3. Now we go back and see where 3 is assigned machine, 3 is assigned to 4. Which means machine 3 is assigned to machine 4. So, machine 4.

Now, I go back and see what happens to 4. So, machine 4 is assigned back to machine 1. So, machine one is assigned to machine 3, machine 3 is assigned to machine 4, and machine 4 is assigned to machine 1. It is like 1 to 3, 3 to 4, 4 to 1. So, one is assigned to 3, 3 is assigned to 4, 4 assigned to 1 implies that; 1 3 and 4 belong to one group. Similarly, 2 is assigned to 6, 6 is assigned to 5, 5 is assigned to 2 means; 2 5 and 6 belong to the same group. So, 2 5 and 6 belong to the same group.

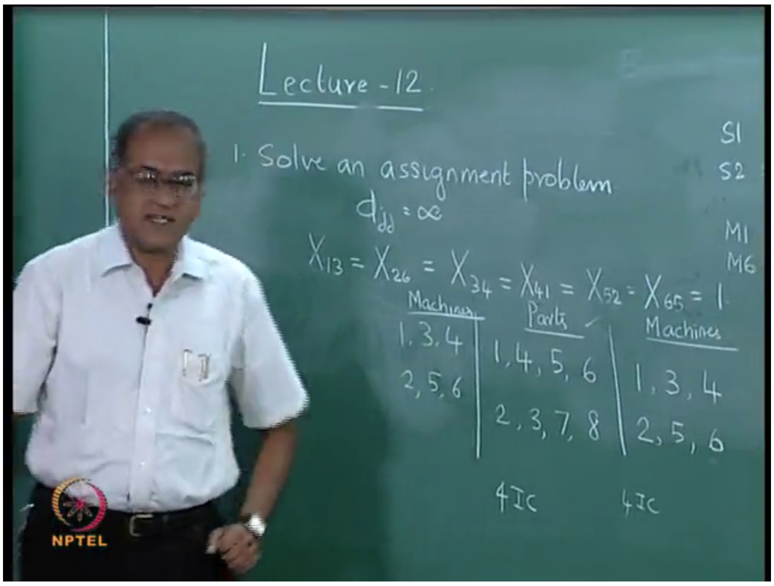
So, we have 2 groups now with 1 3 and 4 2 5 and 6 by know. We know that we have got the solution, but we have got a solution with 2 groups. Now once we have 2 groups, we can always use our part assignment idea, and assign parts to these 2 machine groups. So, when we do that, part one is 1 3 4 6. So, part one goes here. Part 2 is 2 5 6 part 2 goes here. 3 is 5 6. So, 3 goes here. 4 is 1 3 6 4 goes here. 5 is 1 4 5 5 goes here. 6 is 1 2 3 4 6 goes here. 7 and 8 go here. This we have already done, and we know this solution.

Now given a machine group, given a machine group the part assignment rule that we followed. This rule has 2 steps, take every part and assign the part to the machine group where it visits maximum number of machines so that intercell moves is minimized. If there is a tie assign the part to the machine group that has the smallest size. That was our part assignment rule. And we have already seen that the part assignment rule is essentially a way to cluster the parts around machine groups with each machine group acting like a seed. So, given a machine group, the part assignment rule assigns parts to the machine groups, such that the intercell moves created are minimized.

Now, let us do something else. Now let us assume that we do not know this. Let us assume that we know only the part groups. Earlier given the machine groups, we had a part assignment rule which minimized intercell moves. Now let us define given part groups, let us define a machine assignment rule which will minimize intercell moves; how do we do that. These are our part groups take each of these machines now. So, machine 1 has parts 1 4 5 and 6 look at it row wise, parts 1 4 5 and 6 visiting them.

Now, look at one now assume right now that you do not know this; that you do not know this. So, machine one requires 1 4 5 and 6 from the first row. Look at 1 4 5 and 6. So, machine one will come here.

(Refer Slide Time: 40:21)



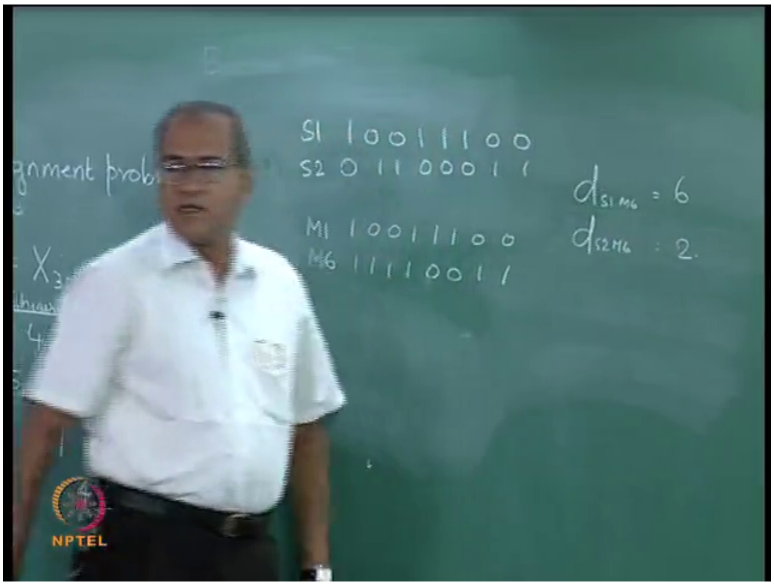
So, this is part, this is machines. Now look at machine number 2; 2 6 7 and 8, 2 6 7 and 8 therefore, machine 2 will come here and create one intercept. Machine 3 1 4 and 6 from the matrix. So, 1 4 and 6 machine 3 will come here. Machine 4 is 1 5 and 6. So, machine 4 is 1 5 and 6, machine 4 will come here. Machine 5 or row 5 is 2 3 5 7 and 8. So, 2 3 7 and 8 5 will come here. 6 is 1 2 3 4 7 8. So, 1 2 3 4 7 8. So, 2 3 7 8 is here, 1 4 is here. So, 6 has to come here with maximum number of parts are required. So, the machine assignment rule is the same as part assignment rule, it says; given part groups assign each machine to the part group where maximum number of parts visit that machine. If there is a tie, assign the machine to the part group with smallest size. So, the part assignment rule, and the machine assignment rules are similar; except that parts are replaced by machines.

So now applying the machine assignment rule, I have got these 2 which is the same as 1 3 4 and 2 5 6. So now, with this if I apply, the part assignment rule again, I will get only this because for this I have got this. So, if I continue to use the part assignment and machine assignment rules alternately, I will keep getting the same set of groups. Therefore, the moment I know that this one is repeating. I can terminate the algorithm, otherwise if they are not repeating I will keep doing this algorithm till they repeat.

So, we can try and use the part assignment rule and the machine assignment rule alternately to try and get this. The other way to do it is to say that I have machines first, this comes out of the assignment algorithm. Now I create parts. Now I find intercell moves, there are 4 intercell moves. Now for this I create machines using the machine assignment rule; I get again 4 intercell moves. So, once the number of intercell moves is the same I terminate. Otherwise I can keep doing this. If the number of intercell moves is reducing, I can keep doing it, till it is the same. Sometimes it could worsen, but it normally does it. So, till it is the same you can continue to do this. So, that is how this algorithm will work.

Now, let us try this. The machine assignment rule is also similar to clustering the machines with each part group providing a seed. For example, if this set is known.

(Refer Slide Time: 44:49)



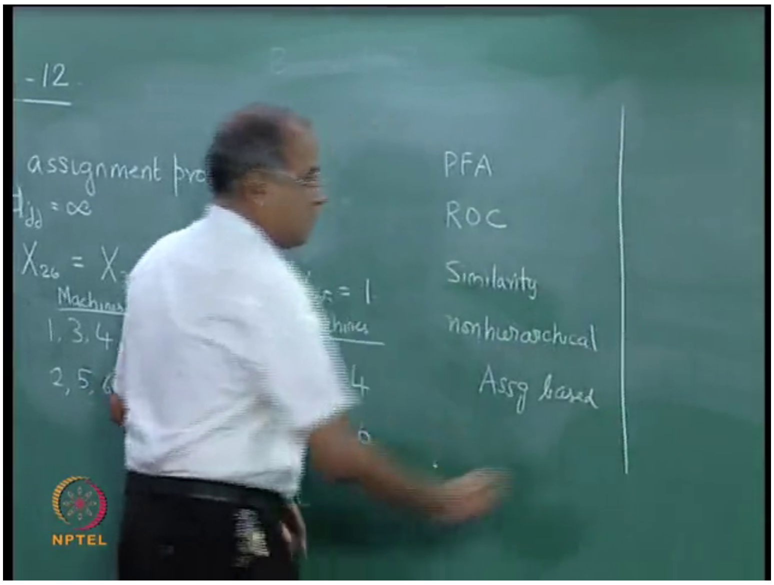
Now, there are 2 seeds, one seed is 1 0 0, 1 1 1 0 0 is 1 4 5 6, this seed is 0 1 1, 0 0 0 1 1. This is this. Now if I take the first machine which is this is seed S 1 this is seed S 2, if I take M 1, which is machine one given by row 1. It is 1 0 0 1, 1 1 0 0, 1 0 0 1, 1 1 0 0. Now this is identical to this. Distance will be 0 therefore, it will go to this. Take M 6, 1 1

1 1, 0 0 1 1, 1 1 1 1, 0 0 1 1, now what is the distance between this and S 1? 0 1 1 2, 0 3 4 5 and 6.

So, distance between S 1 and M 6 is 6. What is distance between S 2 and M 6? 1 0 0 1 plus 1 2 0 0 0 0 therefore, 6 will go to machine number 2. Where it visits maximum. Where maximum parts visited the distance will be less. So, exactly as the same way, one can show that the machine assignment heuristic is also a way to cluster the machines, around known part groups. So, one can use this part family mission group cell formation, one can solve the cell formation problem using an assignment algorithm to provide an initial solution or to provide with initial number of groups, and then apply the part assignment rule to get part families.

Now for the known part families, assign use the machine assignment rule to create machine cells. If the number of groups are the same compute intercell, moves keep repeating this, till the same solution comes. So, that the algorithm terminates. The only difference is that sometimes, we will have unequal number of groups here. For example, I may have 7 or 8 groups for a large matrix, and when I do the part assignment using the part allocation or part assignment rule, all these machine groups will not be able to attract parts. Such cases we have to repeat this, till we get a consistent solution which can be evaluated. So now, what happens is in our earlier classification we had said we will we have seen PFA, we have seen rank order clustering.

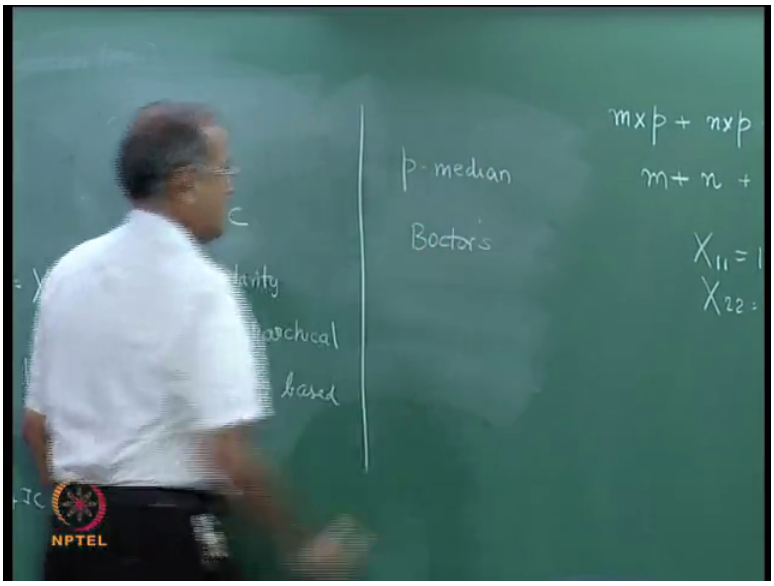
(Refer Slide Time: 48:46)



We have seen similarity based algorithm, we have seen a nonhierarchical clustering algorithm, and an assignment based algorithm. I would still put the assignment based algorithm under the heuristic category, because this portion is solved optimally. This is somewhat similar to the p median solution.

The difference is that p median is also a binary problem 0 1, assignment by definition is a 0 1, but assignment can be solved as a linear programming problem also. So, assignment algorithms take less time. We have algorithms which run with very little time non-exponential character. So, assignment algorithms are easy to solve relatively easier to solve than p median, take less time. So, we could use this and then alternately we could use parts and machines to do that.

(Refer Slide Time: 49:59)



So, under optimization we have seen p median. And we have seen the Boctor's formulation. Now if we go back to the assignment based algorithm; in this particular example, the 6 machines gave us only 2 groups. And both these groups were able to attract parts. So, we had 2-part families which in turn gave 2 machine cells, and the algorithm terminate. Now I mentioned that there could be instances where the number of groups obtained here and here will be unequal. And in such cases the algorithm has to be slightly modified. Now we will take another example; which is the familiar example through which we explain PFA, and then we will show how this algorithm works for that example. And that we will see in the next lecture.