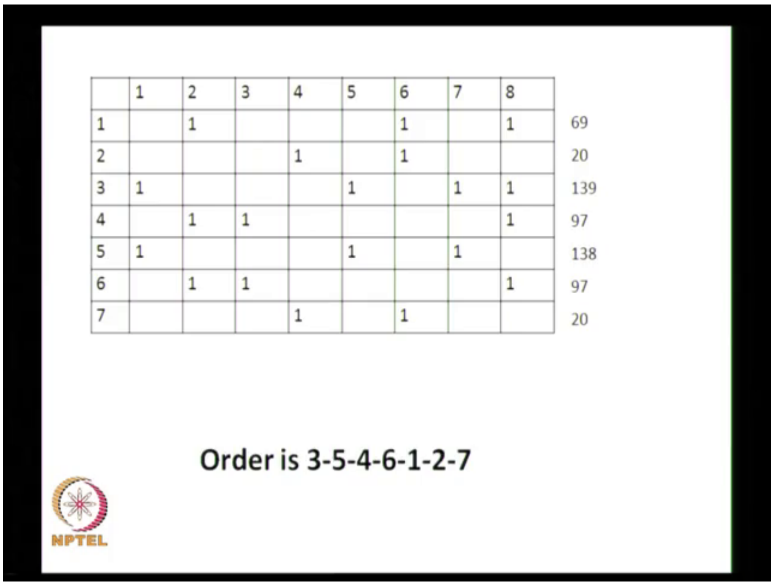
Manufacturing Systems Management
Prof. G. Srinivasan
Department if Management Studies
Indian Institute of Technology, Madras

Lecture - 08
Rank Order Clustering, Similarity Coefficient based algorithm

In today's lecture we look at Rank Order Clustering. Rank Order Clustering is another method to create part families and machine cells in the context of cellular manufacturing. We explained rank order clustering using an example.

(Refer Slide Time: 00:30)



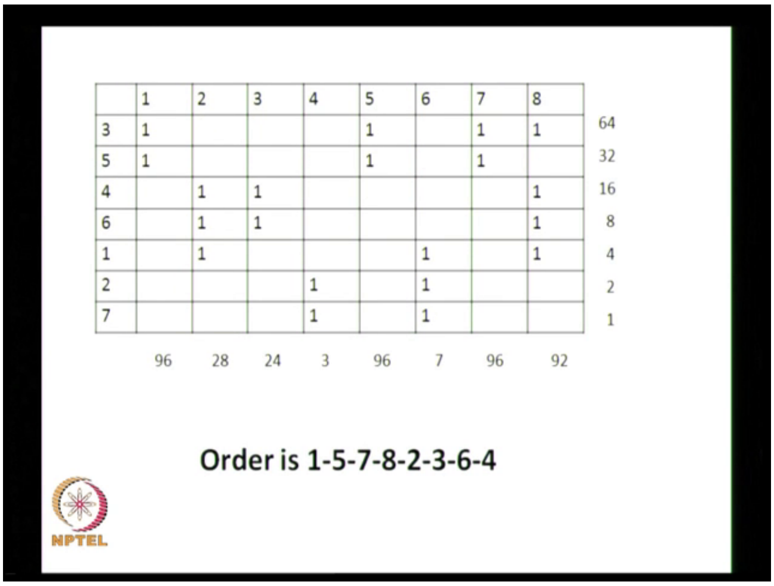
	1	2	3	4	5	6	7	8	
1		1				1		1	69
2				1		1			20
3	1				1		1	1	139
4		1	1					1	97
5	1				1		1		138
6		1	1					1	97
7				1		1			20

We consider this machine component incidence matrix, in this machine component incidence matrix there are 7 machines represented by the 7 rows and 8 parts or components that are represented by the 8 columns, what the sake of illustration we are looking at the small sized incidents matrix, whereas in practice the incidence matrix will be much larger than the size that we are looking at the we are considering a 7 by 8 matrix only for the purpose of illustrating the algorithm understanding fully that real life matrices are much larger.

We follow the basic notation where a 1 represents that part number 2 visits machine number 1 and blank or a 0 indicates that part number 3 does not visit machine number 1. There should be a 0 here indicating there is does not visit just for the, is of representation we replace the zeros by blanks or blanks basis and a blank space represents that the part does not visit the machine. A rank order clustering is a simple algorithm that is being used extensively based on very simple principles what we do now is we try and create a certain wait for each position for example, there are we first look at clustering the rows or rearranging the order of appearance of the rows based on a certain criterion. Now the criterion is as follows each rows has 8 elements representing the 8 positions corresponding to the 8 columns for example, row 1 can be seen as 0, 1, 0, 0, 0, 1, 0, 1.

We try and give the weight to each of gives 8 positions and this takes a weight of 0 which is this takes a weight of one which is 2 power 0, this takes a weight of 2 power 1, this takes a weight of 2 power 2 which is 4, this takes a weight of 2 power 3 which is 8, this takes a weight of 2 power 4 which is 16, this takes a weight of 32, 64 and 128, these are the weights that are given to these positions .

(Refer Slide Time: 03:33)



The weights corresponding to this positions are considered from the right and the right most position has a weight of 2 power 0 which is 1, 2 power 1, 2 power 2, 2 power 3, 2 power 4, 2 power 5, 2 power 6, and 2 power 7. The weight corresponding to the first row is this has a weight of 1, this has a weight of 4, this has a weight of 64 because this is 2 power 7, the weight is 64 for this, the weight is 4 for this and the weight is 1 for this giving as a total weight of 64 plus 4 plus 1 which is 69.

This row this has a weight of 4 and this has a weight of 16, this is 2 power 2, this is 2 power 4, if you go back this is 2 power 0, 2 power 1, 2 power 2, 2 power 3, 2 power 4, 2 power 5, 2 power 6, and 2 power 7 . So, the actual weight is 0 into 2 power 7, plus 0 into 2 power 6, plus 0 into 2 power 5, plus 1 into 2 power 4, plus 0 into 2 cube, plus 1 into 2 square, plus 0 into 2 power 1, which is 2 plus 0 into 2 power 0 which is 1, 16 plus 4 gives as a weight of 20.

Now, here the weight is 128 plus 836 plus 238 plus 139 here the weight is 64 plus 32, 96 96 plus 197, here it is 128 plus 836 plus 238 here it is 64 plus 32, 96 plus 197 and here it is 16 plus 4 which is 20. So, the weights corresponding to each row are computed by giving weights to column positions, the weights are 69, 20, 139, 97, 138, 97 and 20. Now the rows are rearranged according to the decreasing order or non increasing order of the weights. So, 139 comes first, 138 comes second, there are 2 with 97 we can use either 4 first or 6 or the other way then there is one with 69 and then there are 2 with 20.

The order will be 3 comes first followed by 5, 3, 5, 4, 6, 1, 2 and 7, the order will be 3, 5, 4, 6, 1, 2 and 7 that is the order in which the rows are arranged. So, now, go back the rows are arranged row 3 has 1, 0, 0, 0, 1, 0, 1, 1; now 1, 0, 0, 0, 1, 0, 1, 1 just the rows are rearranged the columns are kept intact. So, from this matrix by giving weights of 1, 2, 4, 8, 16, 32, 64 and 128 we find out to the weight corresponding to each row and rearrange the rows in the decreasing or non increasing order of the weights, from this incident matrix we can now get this incident matrix.

We repeat the same thing to rearrange the columns by giving weights to the row positions therefore, the weights for the row positions will be 1, 2, 4, 8, 16, 32 and 64 these are the weights, now using these weights once again they are of the order of 2 power 0, 2 power 1, 2 power 2, 2 power 3, 2 power 4 etcetera.

The weight corresponding to this column will be 64 plus 32 now for better understanding the weights are already shown here, 64 plus 32 will give us 96, here it is 16 plus 8 24, plus 4 28, here it is 16 plus 8 24, 2 plus 1 3, 64 plus 32 96, 4 plus 2 plus 1 7, 64 plus 32 96, 64 plus 16 80, 88 plus 4 92. So, the weights corresponding to the 8 columns are now calculated by assigning weights to row positions the assigned weights are fixed weight to row positions are shown here using these the compute the column weights for each of the column as 96, 28, 24, 3, 96, 7, 96 and 92. Now the columns are sorted or rearranged in the decreasing or non increasing order of the weights, we could start with 1, 96, 96, 96 then 92 then 28, 24, 7 and 3, the order will be 1, 5, 7, 8, 2, 3, 6, 4.

(Refer Slide Time: 10:32)

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

It is rearrange according to this order 1, 5, 7, 8, 2, 3, 6 and 4.

Now, you see that the order is 1, 5, 7, 8, 2, 3, 6 and 4 now this is the rearranged order after rank order clustering. Now when we look at this the observed that we will get a structure like this now from this structure now you seen that the rows are in the order 3, 5, 4, 6, 1, 2, 7 columns are in the order 1, 5, 7, 8, 2, 3, 6, 7, 4 now when we look at this structure this is the block diagonal structure where all the ones in the form of diagonal blocks are kept closer to the diagonal the zeros or off diagonal and they are far away from the diagonal, this is block diagonalising the given 0 1 matrix.

Then we from this we can identify 3 groups.

(Refer Slide Time: 11:42)

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

Where 3 and 5 will from one group machine group with parts 1, 5 and 7, machines 4, 6 and 1 will from another cell which will make another parts 8, 2 and 3 and machines 2 and 7 will from third cell which will make parts 6 and 4. There are 2 inter cell moves cell that are shown in red color and this. These inter cell moves involve component 8 which is attached to group number 2 will visit machine number 3 which is in group number 1 similarly component number 6 which is attached to group number 3 will visit machine number 1 which is attach to group number 2. There are 2 inter cell moves in this solution, rank clustering is a very simple way to try and get a block diagonal structure out of a given 0 1 incidence matrix therefore, rank order clustering can be used to create machine cells and part families exactly the same way as we have described.

This algorithm called rank order clustering comes from contribution of king in the year 1980 and subsequently made better through another paper by king and (Refer Time: 13:15) which came in 1982.

(Refer Slide Time: 13:19)

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

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

Let us try and apply the rank order clustering to another matrix which is given here, in fact, before we go to this example let us revisit the previous one where we have a we restarted with this incidence matrix restarted with the this incidence matrix and then we finally, got a solution which is this. Now when the initial rank order clustering algorithm was first proposed we were giving weights like this at indicated that the weight corresponding to these 8 columns would be 1, 2, 4, 8, 16, 32, 64 and 128 similarly the weights corresponding to the rows where 1, 2, 4, 8, 16, 32 and 64.

If we one made to write a small computer program for rank order clustering one has to do the multiplication with the weights and then identify a score for each row or each column. In this case the scores for the columns are computed based on the weights here, but then we observed that is weights are exponentially increasing because the weights are 1, 2, 4, 8 etcetera. We 7th place here has a weight of 2 to the power 6 because this is 2 to the power 0, 2 to the power 1 2 etcetera. So, the M-th place or if there are M machines here then the top most machine will have a weight of 2 to the power M 2 to the power M minus 1.

Similarly when we are computing these course the weights where 2 power 0, 2 power 1, 2 power 2, 2 power 3, and so on. If there are M columns then the first column weight will be 2 to the power N minus 1. When we apply this technique to a real life problem the may encounter matrices of the size say 50 by 100 or 50 by 200, which means we have to store somewhere a weight or a number of 2 power 199 if we had 200 columns.

This was seen as some kind of a limitation of the rank order clustering algorithm when it was first proposed because as that time it was not possible to store a number as largest 2 to the power 199 and so on as an integer. Subsequently that was overcome by saying that we need not compute a score for each of these rows if we go back to the algorithm to gave a weight to each position which is 0 which is 1, 2, 4, 8, 16, 32, 64 and 128 and based on these weights we computed a score and then we sorted them on the decreasing or non increasing order of the scores.

It was quickly found out that it is not necessary to give to weight to each position and then computer score it is or it is all right to treat this entire thing as a string of 0s and 1s which means this row will be a string of 0, 1, 0, 1, 0, 1, 0, 1 and by simply comparing the 1s 0s in the various positions it is possible to sort them in the order that we want, for example, this string that has a 1 in the first place will come a head of this, this, this, this and this.

The only thing is to check whether 3 comes first or 5 comes first look at the 5 position both are 0s next position both are 0s once again both are 0s both are 1 once, again both are 0s both are 1 there is a 1 and 0 there for 3 will come I head of 5. So, by simply sorting them and treating each row as a binary string we can do a simple sort algorithm and we need not compute a weight associated with each row based on column position weights and then sort that or alternately not necessary to compute a weight for each column as we did here using row position weights and so on.

So, very quickly rank order clustering algorithm was being implemented in the form that was described just now that is by considering each row and each column as a string of zeros and ones and not by associating a weight which was exponential in nature. So, that limitation of rank order clustering was quickly over come and people could use rank order clustering for a very large sized matrix also.

Now after this let us come to another example of using rank order clustering or trying to understand are there other issues with rank order cluster. Let us consider this as a incidence matrix with 6 rows and 8 columns, 6 machines and 8 parts once again a small sized matrix more for the purpose of illustration rather than duplicating a real life scenario, real life matrices are much larger than the size that we are right.

Let us assume that this is the initial matrix and closer look will tell us that this is already sorted in the after the first sort. So, we do let us say 2 phases of rank order clustering and after 2 iterations of rank order clustering one for each row and one for the each column we would get a final block diagonal form like this. So, the final block diagonal form will give us 6, 5, 2, 1, 3, 4 and 2, 7, 8, 3, 1, 4, 5, 6, this will be the final block diagonal structure. Now we get in to the issue of how many groups do we want to create out of this, which machines go to which group and which parts go to which group in the earlier example when we had this again the same question remain this is the block diagonal structure that we got after 2 iterations of rank order clustering one for each row and one for the each one for the rows and one for the columns.

We said that on closer look at this matrix we find that there are 3 groups, now here we have to do this little carefully now the best block diagonal structure that we can think of is a structure like this.

(Refer Slide Time: 21:12)

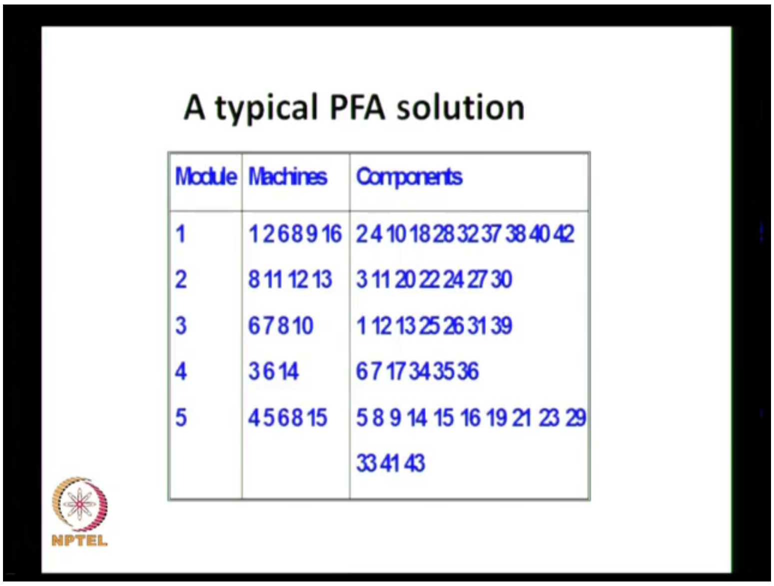
	1	2	3	4	5	6	7	8
6	1	1	1	1			1	1
1	1			1	1	1		
3	1			1		1		
4	1				1	1		
5		1	1		1		1	1
2		1				1	1	1

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

There are 2 groups with machine 6, 5, and 2 forming the first machine cell and machines 1, 3, and 4 forming the second machine cell parts 2, 7, 8, 3 visit the first machine cell parts 1, 4, 5, 6 are attach to a second machine cell with 4 inter cell moves. So, the point that I making here is that rank order clustering per say or explicitly does not give the machine groups and part families rank order clustering gives only a rank ordered representation of the given matrix.

So, essentially there are 2 stages to the algorithm the first stage is to get the rank order matrix which will look like this, the second stage is to create machine cells and part families using a matrix like this. Many times this is evident and can be seen through the naked eye and we could make machine groups and part families. Sometimes different people looking at the block diagonal structure may create different solutions which means different configurations of machine cells and part families or we need another algorithm which will identify machine cells and part families from a given block diagonal structure that comes out at the end of the algorithm. When we have something like this as they final structure and we have a structure like this now we realize that this is the final solution.

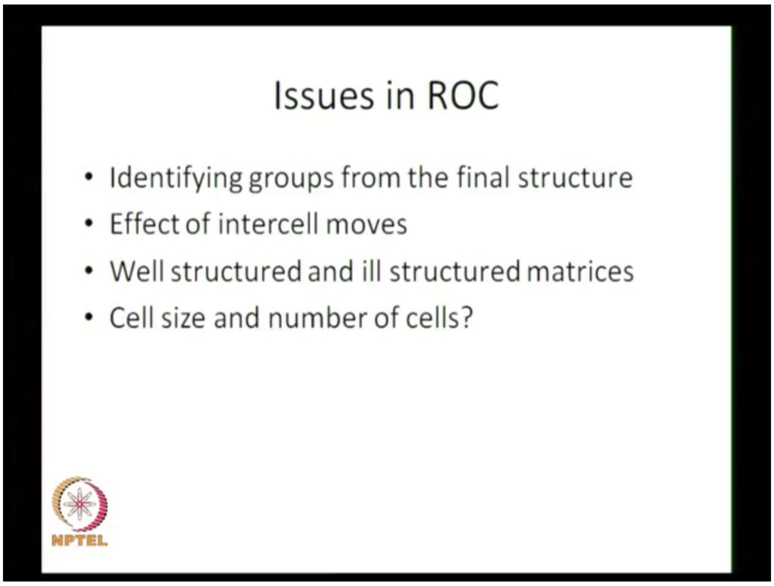
(Refer Slide Time: 23:07)



A typical PFA solution we have seen slide earlier in our example for the 16 machine forty 3 part example a typical PFA solution will be like this it will give the machines belonging to each group and components belonging to each group. So, the end of the production flow analysis algorithm we would get exactly how many groups we have what are the machines go to each group and what are the part are go to each group.

Now, from this table we could create this matrix where as in rank order clustering we actually have this matrix and then from this matrix we draw this 2 lines it may look simple that we are just drawing tool lines are we are portioning it, but after the portioning we can get the tabular representation where we would say 6, 5, 2 belong to the first group 1, 3, 4 belong to the second group. Now this by itself can become some subjective at some point and need additional effort to do that and they may get different solutions depending on who is doing it now what are the issues in rank order clustering the.

(Refer Slide Time: 24:27)

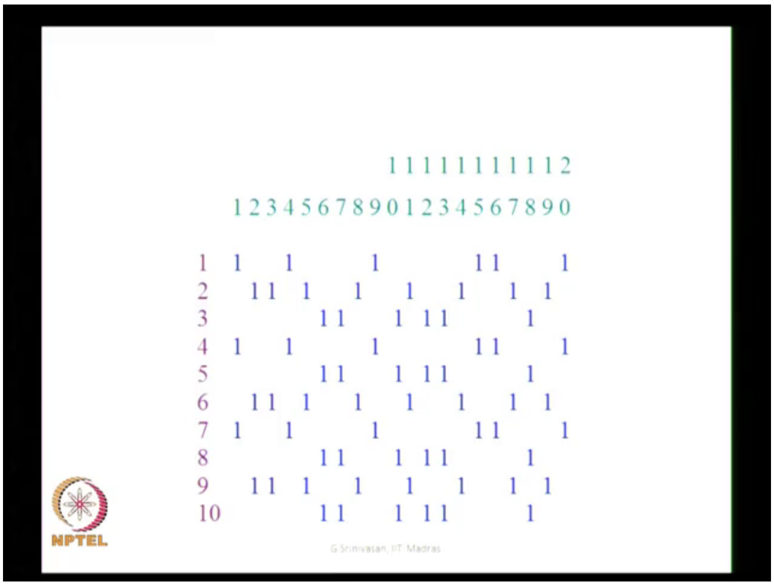


First one is identifying groups from the final structure which is what I describe right now, whereas, PFA we get the groups and then we create the matrix in rank order clustering the matrix is given first from which have to create the groups. Then the effect of inter cell moves this aspect is very similar to that in production flow analysis whether we use production analysis or rank order clustering we finally, get machine groups and part families we get inter cell moves these inter cell moves will have to be eliminated and we have to fall back on the 4 ways that that we saw in the earlier lecture to eliminate or minimize inter cell moves.

These 4 are can I carry out with then existing machine can I alter the production plane process planes in such a manner that it is possible to do it here or can I make some small engineering changes and executed should I duplicate the machine or by a slightly less expensive machine and put it there or do I carry out part sub contracting or operations sub contracting these are the ways by which one can minimize inter cell moves.

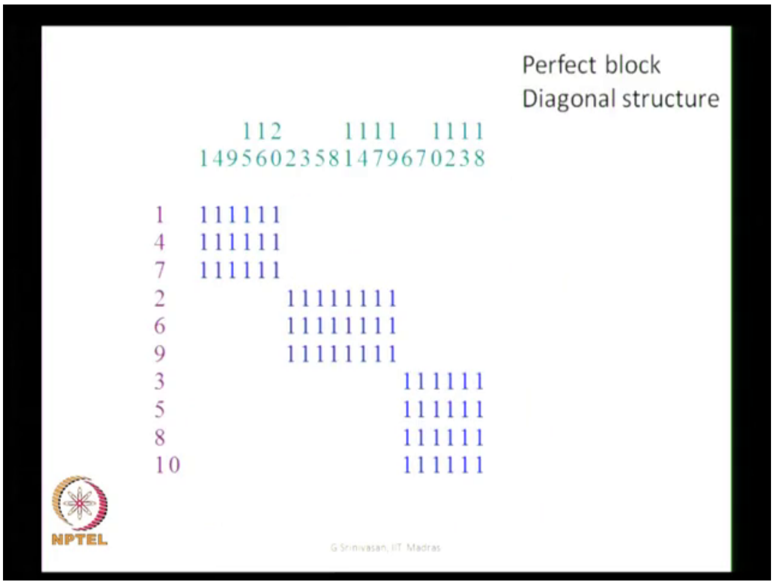
Next one is to understand are their problems are situations where rank order clustering may not do well for example, if we consider a matrix like this.

(Refer Slide Time: 26:01)



This is another example of a machine component incidence matrix this example has 10 rows and 20 columns which means 10 machines and 20 parts and components, if we carry out rank order clustering by doing it first for the rows and then for the columns we will get a structure like this.

(Refer Slide Time: 26:23)

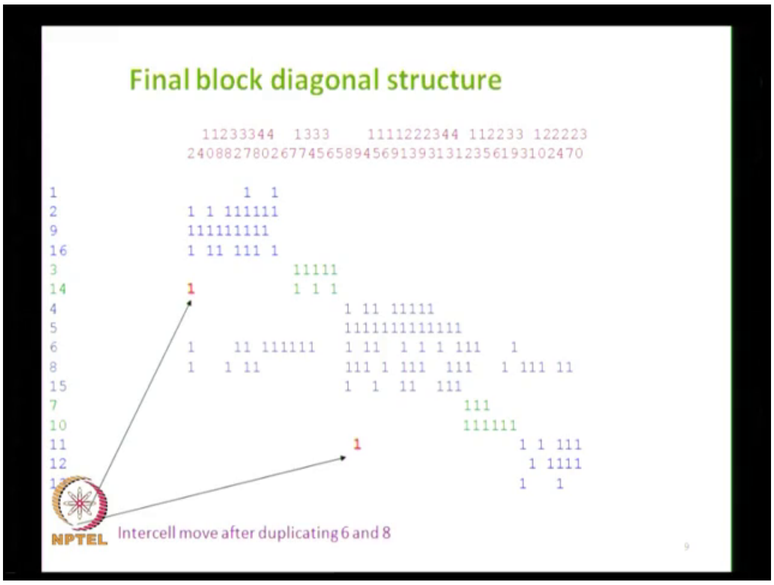


Now, this matrix has been specifically created so, that we get a very nice structure here, the second part of the algorithm which is actually identifying machine groups and part families from I a given block diagonal structure is very simple in this particular structure because which very evident that there are 3 groups now we see all the ones are clustered beautifully together there are 3 distinct blocks all the zeros are outside of the blocks.

The second part of identifying part families and machine cells is easy, this type of a matrix is called a perfectly block diagonalizable matrix or well structured matrix. Sometimes we have what are called badly structured matrix which are occasionally called as ill structured matrix and in such instances rank order clustering becomes a little difficult in the sense the algorithm will give you a solution like this, but then the task of identifying part families and machine cells becomes difficult when we use a method like rank order clustering.

There are ways to overcome that now we can for example, we have this and let us say we do this classification and for some reason we understand that this classification has 4 inter cell moves, sometimes by closely looking at a matrix we would know that there can be some rows are some columns which actually have an unusually large number of ones. If we know that such columns are such rows are such machines are such parts will create inter cell moves, sometimes we can remove a particular part or we can remove a particular one from the matrix and then carry out rank order clustering. And the whole order of the final block diagonal form will become very different. Now those things come out of experience after that we can put back the one in the corresponding position after we have block diagonalize. In fact, when we go back to production flow analysis barbet also gives certain interesting thumb rules when we do the group analysis if you remember this is a solution that comes from our 16 machine 43 part matrix.

(Refer Slide Time: 29:02)



If you see this very carefully we know that 6 and 8 are 2 machines which have a very large number of components visit. In fact, barbet says that if there is a machine there all the components visit now actually take it of the analysis and after the analysis duplicated in every cell if there is a machine where only one part visits. Once again taking out of the matrix and after production flow analysis put that machine back into the cell where that part visits. Similarly if there is a part which visits all the machines take it out of the analysis.

And then see what more can be done because such a part will create inter cell moves which will also in a way if you have a part that visits all the machines if you see the way the production flow analysis works because we create complete cells for each part the cell where such a part is assign will have all the machines and because it has all the machines every other module will be a subset of that, all of them will be merge to give you a single cell with all machines and all parts which you do not want. So, take out such parts from the matrix and then proceed.

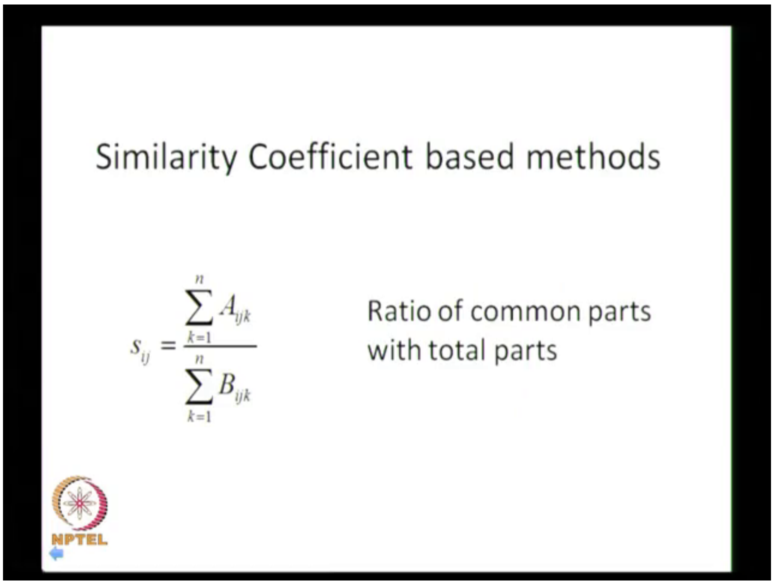
So, one of the best ways overcoming this limitation where you are enable to get the nice block diagonal structure is to do the rank order clustering once find out which are the positions that will definitely be inter cell moves take those ones outside then apply rank order clustering again. So, that in the end you get a very nice structure. And then you can put back all those ones that have been taken away from the matrix in the right positions and then see where the inter cell moves exits.

The last one of courses in many practical situations the want to restrict the number of cells as well as to restrict the cell size once again if you go back to one of the slides that we have seen in an earlier lecture were I had spoken about applications at said that the large cell size examples did exist there where cells where that went up to 15 machines and so on. So, many times as the cell size increases, we understand that the control on the cell comes down, smaller the size of the cell the better it is many times a reasonably large number of small sized cells are desirable rather than having a very small number of very large sized cells.

We need to look at these 2 aspects of cell size and number of cells both production flow analysis and rank order clustering explicitly does not address this, but leaves it to the decision maker to incorporate these aspects because when we look at the PFA solution or when we look at a rank order clustering solution there could be times when you have one slightly large cell we have seen that in an earlier example where a particular cell had 11 machines.

Now, do we want to create a cell with 11 machines are do we want create a cell with 8 machines, we saw in the previous example there the one that had a cell size of 11 involved 7 extra machines, whereas the other alternate solution that had 8 machines in the cell involved 8 extra machines. So, cell size and the number of size it is also a factor which is not explicitly addressed in the algorithm, but the decision maker is given the flexibility and the freedom to choose these 2 parameters and adjust the cell configuration after the solution from these 2 algorithms are obtained.

(Refer Slide Time: 33:31)



Now, having seen rank order clustering; let us move to some similarity based methods now before we understand what similarity based methods are let us once again re visit this simple example we have seen this example this example is the one that gives as a perfectly block diagonal structure.

When we did rank order clustering for this very quickly we observed that if we take weights and then do it or if we treat each one as a string now 1, 4 and 7 are together you see 1, 4 and 7 they are identical that is why they are together here 2, 6 and 9 are identical. So, if you keep 1 and 4 next to each other you realize those 2 strings are perfectly identical 1 and 7 are identical 4 and 7 are identical and so on. Now we can say that 1 and 4 are also similar, they are the same they are similar now how did it happen, if we know translate this information that we have here the block diagonal structure into a real life scenario we now say that there are machines 1, 4 and 7 which process parts 1, 4, 9, 15, 16 and 20.

These 3 machines 1, 4 and 7 will not the similar machines in terms of it is configuration for example, if you say that there is a part number one that visits 1, 4 and 7 it is quite likely that 1, 4 and 7 are different machines one could be a length one could be some grading and so on, but they are similar in the sense that all of them are processing parts 1, 4, 9, 15, 16 and 20 because of which the rows where the same. So, we could go back to the rank order clustering and say that if 2 rows are identical then those 2 will come one after another, which is also true because if 2 rows are identical they will get the same score and therefore, will come one after another, example we have seen here in the earlier once we have seen here that these 3 columns are identical because they have the same weight same 1 2, 1 2, 1 2 identical columns they have the same weight and they come together 1, 5 and 7.

There is these 2 they look very similar 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0 they look very similar, but they are not identical, they will they can come next to each other they may come next to each other 2 3 and so on. So, what we can do now there is to try and find out some kind of a similarity index or a similarity coefficient between the rows and then start grouping them based on these similarity coefficient, those methods are called similarity coefficient based methods. There are several ways of defining the similarity coefficient; we are going to see one way of defining a similarity coefficient.

Now, let us look at this matrix and let us try and compute some similarity coefficients.

(Refer Slide Time: 37:44)

	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

	1	2	3	4	5	6
1	--	1/7	¾	¾	1/8	¼
2		--	1/6	1/6	½	3/7
3			---	½	0	2/7
4				--	1/7	1/8
5					--	4/7
6						--

This is our incidence matrix once again very small sized incidence matrix now there are 6 rows representing 6 machines. We construct a similarity coefficient matrix among the machines or among the rows. This is the similarity coefficient matrix among the machines or among the rows. Now what is the similarity between 1 and 2, now the similarity between 1 and 2 is computed to be 1 by 7 which is based on this formula which we have to explain now. So, if we take rows 1 and 2, 1 has component numbers 1, 4, 5 and 6 visiting this has component numbers 2, 6, 7 and 8 that are visiting.

The first thing we do is which are the common components are how many common components are visiting both 1 and 2, if you see that rows carefully there is only 1 common component visiting which is component number 6, which also means there is only 1 1 pair all other pairs are either 1 0 pairs or 0 1 pairs or 0 0 pairs. If we take these 2 rows for this component is a 1 0 pair and for this it is a 0 1 pair, for this it is a 0 0 pair and for this it is a 1 1 pair, 1 1 pair tells you that this component is visiting both the machines, the number of components visiting both the machines is one which is the numerator.

And if you bring these 2 machine together how many components will visit either of them which means we should look at 1 0 pair, 0 1 pair and 1 1 pair, they leave out only the 0 0 pair if we do that we realize that this is a 1 0 pair. So, component 1 will come, component 2 will come, 3 will not come, 4 will come, 5 will come, 6 will come, 7 will come, and 8 will come. The union set is 7, the union set is the sum of the 0 1 pairs, 1 0 pairs and 1 1 pairs, the union or the number of components that will visit either of the machines or both the machines is the denominator, the numerator is 1 the denominator is 7 and the ratio is 1 by 7 for 1 and 2.

Let us look at 1 and 4 if you look at 1 and 4 we have 1, 0, 0, 1, 1, 1, 0, 0 and for 4 it is 1, 0, 0, 0, 1, 1, 0, 0, which are the common ones the numerator will have 1, 5 and six. So, there are 3 parts are components that are visiting both 1 and 4, the numerator is 3, the denominator is 1 has 1, 2, 3 and 4, this has 1, 2 and 3. So, denominator will have 1 visits both 2 is not counted, 3 not counted, 4, 5 and 6 come, 7 and 8 are not counted, only 4 components 1, 2, 1, 4, 5 and 6 will visit both 1 and 4 or either 1 or 4 or both if they are combine together, denominator is 4 numerator is 3 and there for the similarity confident is 3 by 4 like this we can contracts the similarity coefficient matrix.

This matrix is actually a symmetric matrix, but only one half of the matrix is shown the diagonals are not explicitly computed, because similarity between a machine and itself is always one and we do not require the number is in the diagonal because our idea is to bring them machine together group them. So, a number here in the diagonal is not going to add value to the purpose for which we are computing this similarity coefficient.

Similarity coefficient matrix is always shown either as a upper triangular and lower triangular with only the diagonal not shown and only one part of the triangle a shown here we show this matrix as an upper triangular matrix which is a 6 by 6 matrix because we have 6 rows here. So, we compute this similarity coefficient now look at the 0, 3 and 5, 3 has 1, 4 and 6, 5 has 2, 3, 5, 7 and 8 there is no part that visits 3 and 5, numerator is 0 denominator is 8 therefore, it get 0. Similarity coefficient is always between 0 and 1 it is not negative.

Now using the similarity coefficient matrix we now try and group the machines and how do you do that now we look at the first one and we realize that the maximum similarity is 3 by 4, 3 by 4 is 0.75 the next one is 4 by 7 which is less than that, the maximum similarity is 3 by 4. So, we can either group 1 and 3 first or 1 and 4 first, we could group that and we start the first grouping by saying that machines 1 and 3 are grouped first then the rest of them are grouped. The next matrix that you will have is a 5 by 5 matrix where 1 and 3 are grouped, first 2, 4, 5 and 6 remain. Similarly 1 and 3 are grouped 2, 4, 5 and 6 remain.

We know that given 1 and 3 we can compute the similarity coefficient using the formula that we used they could compute the numerator and the denominator and we could do that, but once we form 1 and 3 together how do I calculate the similarity between this 1 and 3 and 2. So, to do that we go back to this matrix the earlier matrix, similarity between 2 and 1 and 3 is 2, 2, 1 the similarity is 1 by 7, 2, 2, 3 the similarity is 1 by 6.

(Refer Slide Time: 46:01)

	1,3	2	4	5	6
1,3	--	1/6	¾	1/8	2/7
2		--	1/6	½	3/7
4			--	1/7	1/8
5				--	4/7
6					--

{1,3,4}, [2}, [5}, {6}. The next highest similarity coefficient is between 5 and 6 and we have a solution {1,3,4}, {2}, [5,6}. We then have the solution {1,3,4} and {2,5,6} with 2 groups and the solution {1,2,3,4,5,6} as a single group.

We go back here, 1 and 2 by 3 is seen as 1 by 6 which is the smaller of the similarity 2 to 1 similarity is 1 by 7, 2 to 3 similarity is 1 by 6, the bigger of the 2, 1 by 7 and 1 by 6 the bigger of the 2 is taken as a 2 similarity is between 2 and 1 and 3, similarly between 4 and 1 and 3, 1 2 4 is 3 by 4, 3 to 4 is half, 3 by 4 is taken 1, 3, 4 is 3 by 4, similarly 1 to 5 is 1 by 8, 3 to 5 is 0. So, between 5 and the group 1 3 the maximum of them which is 1 by 8 is taken similarly 1, 2, 6 is 1 by 4, 3 to 6 is 2 by 7 and 1 by 4 which is bigger than 2 by 7 is taken.

The rest of them are easy 2 to 4 is 1 by 6 that comes from the same matrix, we now we have a 5 by 5 matrix and then we look at which is the maximum similarity, now we realize the 3 by 4 is the maximum similarity, 4 goes to 1 3 now the next set will be 1, 3, 4 together as one group, 2 as another group, 5 as another group, 6 as another group. So, now, we generate the 4 by 4 matrix from this where the first row will be 1, 3, 4, 2, 5 and 6, column is again 1, 3, 4, 2, 5 and 6.

Now, we follow the same approach from this matrix we can create the next matrix by taking the maximum of the similarities and if we continued by this process finally, we will get we will the next iteration we will get 4 groups with 1, 3, 4, 2, 5 and 6. Then we will get 1, 3, 4, 2 separately 5 and 6 will go together then we will have 1, 3, 4 together 2, 5, 6 together and then we will have all of them coming together. So, we would started with a solution where each of these 6 machines are 6 rows where 6 different groups from there we came to the solution with 5 groups where 1 and 3 was group together, then we came to a solution with 4 groups were 1, 2, 4 was group together, then we come to the solution with 3 groups there 1, 2 and 3, then we come to a solution with a groups and then we come to a solution with a single group. If we say that we want a solution with 2 groups we would now take this solution with 1, 3, 4 and 2, 5, 6 as 2 groups.

We would go to the solution with 1, 3, 4, 2, 5, 6 with 2 groups which is the solution that we have here say if we want a solution with 2 groups we will have this 1, 3, 4 and 2, 5, 6 as 2 groups with 4 inter cell moves, this is one of the similarity coefficient based methods. The method that we have seen called a single linkage clustering algorithm, it is a clustering algorithm, because it uses clustering methods to form groups, it is called single linkage because at this stage when we computed the similarity between 1, 3 and 2 we took the maximum out of 1, 2 and 3, 2 for example, we took the maximum out of 1, 2 and 3, 2 when we take the maximum it is called single linkage cluster analysis.

Now once we identify 2 groups and the solution from here which is 1, 3, 4 and 2, 5, 6 we have to now get the part families. One of the ways of getting the part families this to do the same similarity coefficient now for the parts instead of the machines, here we have done similarity coefficient for the machines we can do similarity coefficient for the parts. So, that will be an 8 by 8 matrix and from 8 we go to 7, 6, 5, 4, 3, 2, and 1.

And then find out the solution where there are 2 part groups and we will realize that the 2 part group solution will be this with 1, 4, 5, 6, and 2, 3, 7, 8. Then we create the final block diagonal structure and get machine groups and part families. Now machine groups and part families can be formed separately for machine groups as well as separately for part families.

Another way of doing it is once we form machine groups we say we have 2 machine groups with 2, 5, 6 and 1, 3, 4, now part families can be easily formed by looking at this, look at part number 1 part number 1 requires 6, 1, 3, 4, which of these groups it will go, it will go to 1, 3, 4 because there are more machines. Part one will go to the group with 1, 3, 4, part 2 requires 2, 5 and 6, which group will it go it will automatically go to the machine group with 2, 5 and six. If we follow this simple rule of allocating parts based on an existing machine configuration most of the times we get similar results and we could get the apart families automatically from this.

So, other issues related to single linkage cluster analysis we will see in the next lecture, where I will also try and explain what exactly is this single linkage. And at the same time I will also introduce you to some basic principles of cluster analysis and how this algorithm comes out of cluster analysis.

We will see all those aspects in the next lecture.