Project and Production Management

Prof. Arun Kanda
Department of Mechanical Engineering
Indian Institute of Technology, Delhi

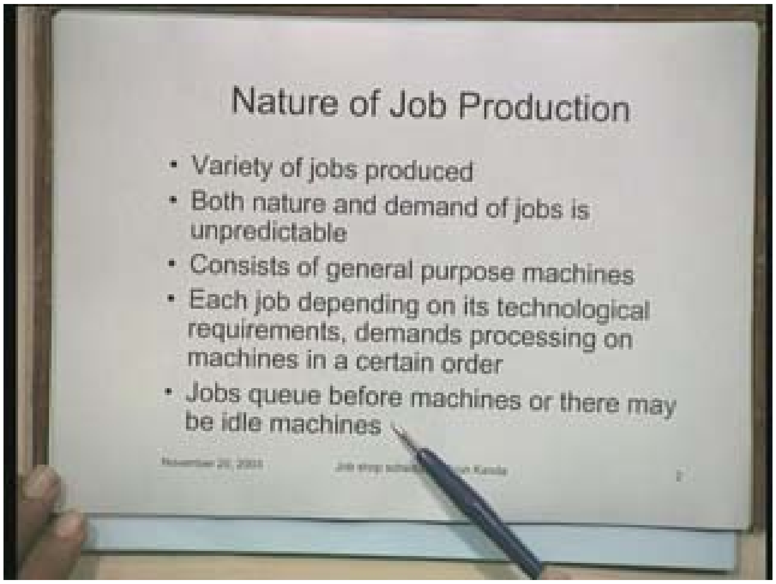
Lecture - 41

Scheduling of Job Shops

Today we are going to be talking about scheduling of job shops and when you are talking about scheduling of job shops, I think you must appreciate what we mean or what is the nature of job production and what kinds of problems exist in the job shop and what is it that we are trying to address through job shop production. In a typical job shop what is happening is that a variety of jobs are being produced. That is one thing and both the nature and the demand of jobs are unpredictable in a typical job shop situation. You can contrast this with batch manufacturing where in batch manufacturing you are sure about what you have to produce and you are typically producing those very quantities at predetermined intervals. But the nature of the job shop is that depending upon the kind of jobs which emerge the there is therefore a much greater degree of unpredictability in the job shop. A job shop typically consists of a large number of general purpose machines as opposed to special purpose machines which would happen in an assembly line.

So once you have general purpose machines that means you can handle a variety of jobs on those machines and therefore the job shop is equipped to handle this kind of variety in the with a variety of with a number of machines of different kinds. Each job depending on its technological requirements demands processing on machines in a certain order.
What we are meaning here is that each job by virtue of its special nature, the number of sequence would be different for each job and therefore what we are saying is that each job is essentially unique in a typical job shop and jobs will tend to queue before machines. That is one situation, or there may be idle machines and there are no jobs for that particular machine.

(Refer Slide Time: 03:42)



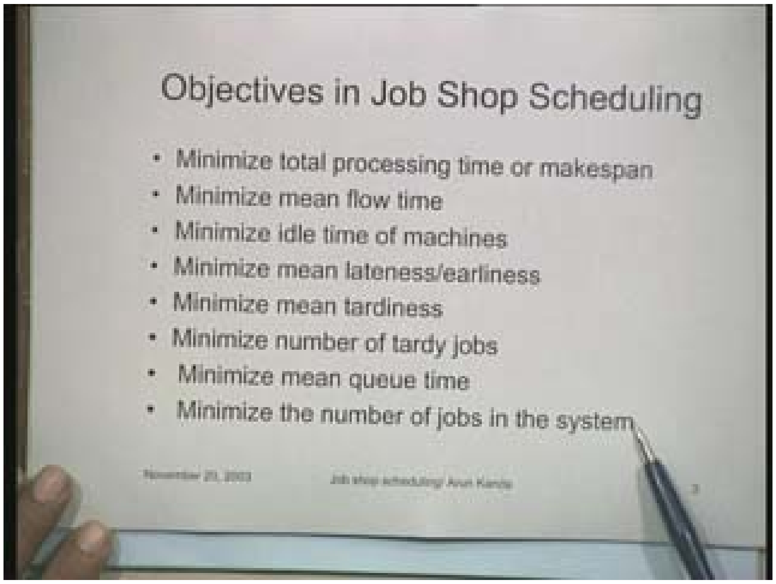
Nature of Job Production

- Variety of jobs produced
- Both nature and demand of jobs is unpredictable
- Consists of general purpose machines
- Each job depending on its technological requirements, demands processing on machines in a certain order
- Jobs queue before machines or there may be idle machines

These kinds of situations might occur temporarily in a job shop, then what we do is let us try to see the objective in job shop scheduling. In fact there could be a very large number of objectives. We have listed here some of the common ones. For instance minimize the total processing time or the makespan. You know what a makespan is. Makespan is the total processing time or the time required to complete all the jobs that you are required to produce. So minimizing the total processing time or makespan is one common objective to minimize the mean flow time. The mean flow time actually talks about flow time. Does anyone have an idea what flow time means? We will define it more formally in a minute in the next slide but what it means is that if a job comes on the scene at a certain point of time then it has to wait for processing up to a certain period and once the waiting is over then it is processed. The total time that a job spends on the shop floor, including the waiting and processing is generally known as the mean flow time. That is the time for which the job is flowing around on the shop floor. That is it. So minimization of the mean flow time is also a very major objective.

Minimizing the idle time of machine is because you do not want to unnecessarily keep the machines waiting for no rhyme or reason, and then you want to minimize the mean lateness or earliness of the job. Quite often on the parlance of job shop scheduling we can talk about mean lateness and mean earliness though in common parlance we do not tend to use the word earliness. It is supposed to be grammatically incorrect but here you are talking about lateness. So if the job is due at time ten and you actually make it at the time fifteen that means it is actually late. It is late by five days or five minutes. So that is what you mean by mean lateness and if it is earlier than what it is scheduled to be then it is earliness. In fact that means the lateness of a job can be either positive or negative. If it is positive that means you are actually late. That is what we normally refer to in the context of job shop scheduling as tardiness. It can be produced early. We can however just be talking about lateness and earliness. So any criterion based on lateness and earliness could be a criterion. What can happen is that you might want to do the job early if there is an incentive given to you. For every one hour completed earlier you get a certain benefit. You get a certain value. You earn a certain amount of money. So it is like doing it, completing it early and you get a certain amount of benefit and doing it late you might incur certain penalty. So when you are having this kind of situation, then this would be appropriate. Minimizing the mean tardiness, minimizing the number of tardy jobs, if you are making let us say five jobs you would like to minimize the number of jobs which are actually tardy. This is the situation that can be done, you can try to minimize the mean queue time. That is the amount of time that a job spends in the queue. So one job spends ten minutes, the other spends half an hour and so on. So the average time that the job spends is the mean queue time and you might want to minimize the mean queue time. You want to minimize the number of jobs in the system.

(Refer Slide Time: 08:43)

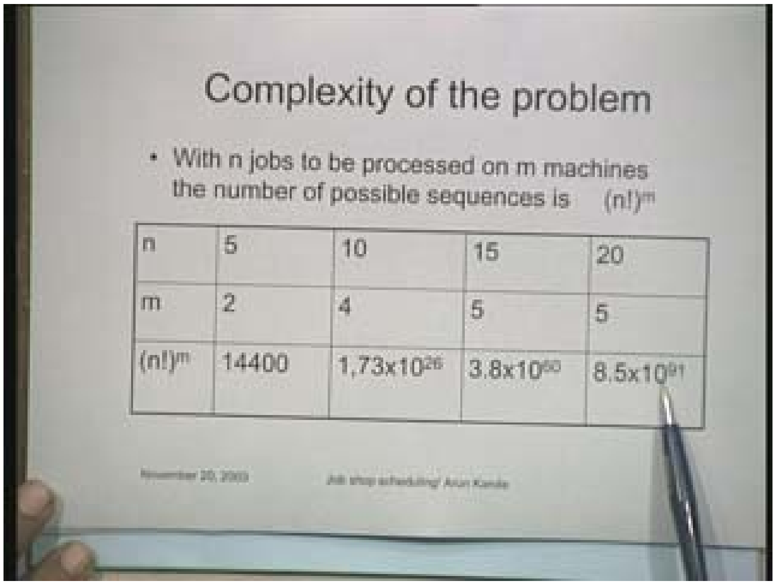


Objectives in Job Shop Scheduling

- Minimize total processing time or makespan
- Minimize mean flow time
- Minimize idle time of machines
- Minimize mean lateness/earliness
- Minimize mean tardiness
- Minimize number of tardy jobs
- Minimize mean queue time
- Minimize the number of jobs in the system

The lesser the number of jobs in the system the more efficient you are. Actually if there is no job in the system you have done all the jobs and therefore it a good, it would be a desirable thing to do. There could be a variety of objectives in the typical job shop scheduling situation. The job shop scheduling problem is supposed to be a very complex problem. Mathematically to give you an idea of how complex the problem could be, we have a general result which says that if there are n jobs to be processed on m machines, the number of possible sequences is n factorial to the power of m. This is the maximum number of sequences that are possible with n jobs and m machines. Just to give you an idea of how big the number can be, you can see for instance if the number of jobs is 5, 10 and 15, let us say it goes up to 20 and the number of machines is 2, 4, 5 only in that sense. This particular value of the number of possible sequences for this particular situation is 14,400 jobs possibility combinations. With 4 machines and 10 jobs this number will go to 1.73 into ten to power 26 number of possibilities. With 15 jobs this value will go up to 3.8 into 10 to power 60. It is a very large number and in fact if you are using even the fastest computer to evaluate one possibility, you can take years for the computer to actually evaluate all possible combinations and you have a situation here where you are dealing with 20 jobs and 5 machines. In that case the number of combinations that you can land up with 8.5 into 10 to power 91, very large number again.

(Refer Slide Time: 11:06)



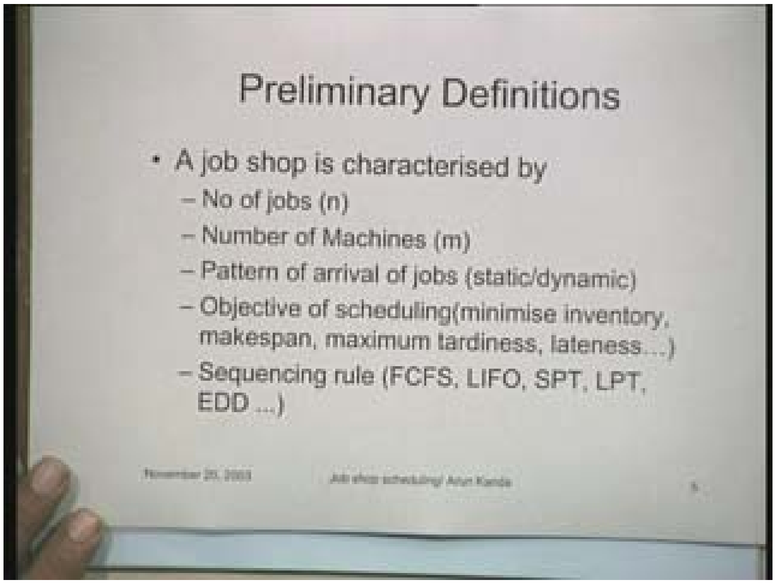
Complexity of the problem

- With n jobs to be processed on m machines the number of possible sequences is $(n!)^m$

n	5	10	15	20
m	2	4	5	5
$(n!)^m$	14400	$1{,}73 \times 10^{26}$	3.8×10^{60}	8.5×10^{91}

It is actually because of this combinatorial nature of the problem that it is difficult to find the solution to the general n m job shop problem. The only way you can find practical solutions to such cases is through the use of heuristic rules and it is very difficult to find out an optimum solution except in certain very trivial or simple cases. What we will try to look at in this particular lecture is to see some of the situations for which exact solutions are available and we will see the kinds of algorithms which are required for solving this particular problem. Some definitions are in order here. How do you characterize a job shop? A job shop is characterized by the number of jobs. That is one thing. The number of machine, m is the second thing. The number of jobs, the number of machines, then you can talk about the pattern of arrival of jobs. Is it a static job shop or is it a dynamic job shop? The difference is that in a static job shop all the jobs are actually available at time zero and you will know the situation and no further jobs will arrive will be arrived till you have solved the problem. But in a dynamic situation what will happen is you have jobs coming, you have more jobs coming and you therefore do not know when a next job will come and therefore the pattern of arrival of jobs is actually dynamic in character. The objective of scheduling is to minimize the inventory, minimize the makespan, minimize the maximum tardiness, and minimize the lateness and so on. These are some typical objectives in this case and the scheduling rule that is normally used for solving such problem is first come first served last in first out shortest processing time, longest processing time, and earliest due date and so on. You have a number of priority dispatching rules.

(Refer Slide Time: 13:36)

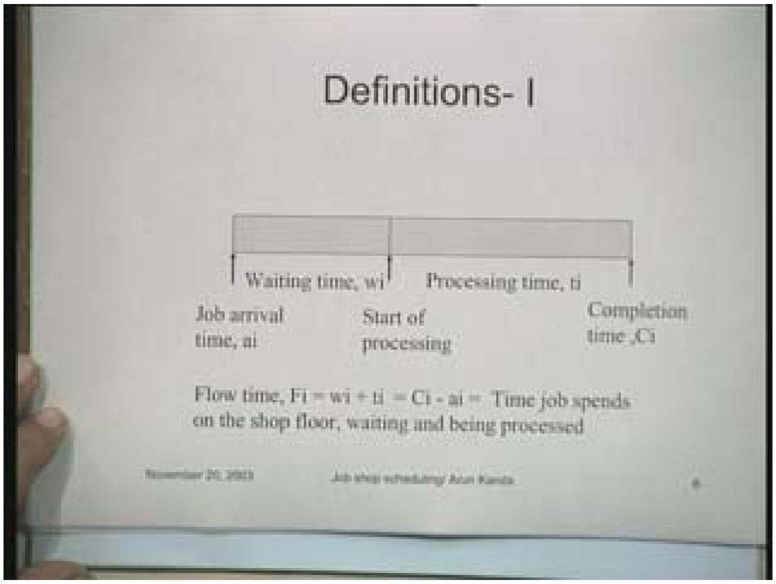


Preliminary Definitions

- A job shop is characterised by
  - No of jobs (n)
  - Number of Machines (m)
  - Pattern of arrival of jobs (static/dynamic)
  - Objective of scheduling(minimise inventory, makespan, maximum tardiness, lateness…)
  - Sequencing rule (FCFS, LIFO, SPT, LPT, EDD …)

We will in fact comment upon the various types of priority dispatching rules that are therein situations where you have the general job shop, that means you do not have results very well defined and therefore for such cases quite often we are content with using or choosing kind of priority dispatching rule. Let us define some terms, so let us say the job arrives on the scene at time ai or ai is the arrival time of the job in general, so this is the time that the job arrives. The job then waits typically and this is the waiting time of the job is up to this particular point of time. So at this point of time there is a start of processing and once the start of processing takes place, the total amount of time required will be the processing time required for that particular job. This ti will be the time required and in this particular time, this would be the completion time of the job and so you have it here. What we are simply defining is that the flow time of a job fi is nothing but the waiting time of the job plus the processing time of the job. Flow time is nothing but the waiting time plus the processing time of the job and another way of looking at it is that this is normally equal to C_i which is the completion time of the job – the arrival time of the job.

(Refer Slide Time: 15:22)



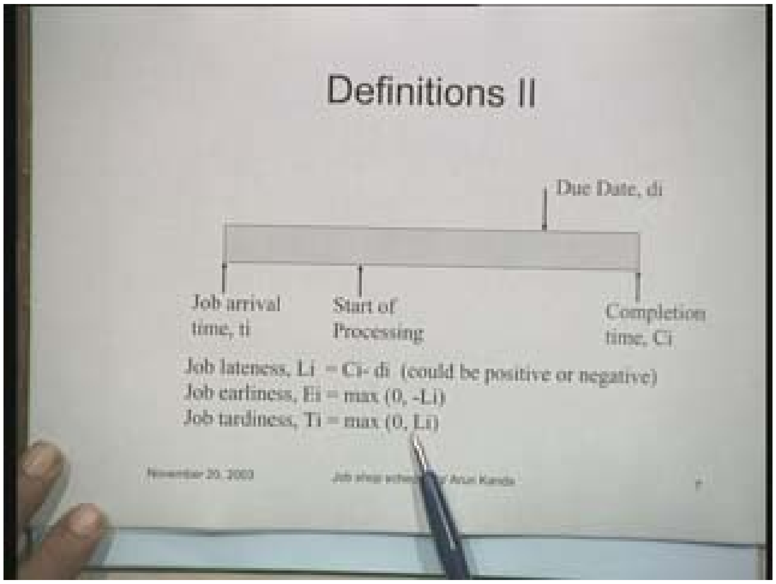
Definitions- I

Waiting time, wi
Processing time, ti
Job arrival time, ai
Start of processing
Completion time ,Ci

Flow time, Fi = wi + ti = Ci - ai = Time job spends on the shop floor, waiting and being processed

So Fi = W_i T_i + T_i – C_i – A_i and the physical interpretation of this is the time, the job spent on the shop floor waiting and being processed. You have this particular definition for the flow time of any job. We can also talk about due date related criteria and again what you find is that this is the arrival time of the job which is T_i. This is the start of processing of the job here. The job comes here. It waits for so much time, then it starts processing and then subsequently you have the completion time of the job which is C_i and when you are talking about the completion time as C_i, the due date for the job might be something like this. D_i could be earlier or later than the due date. We can define job lateness L_i as nothing as anything but C_i – D_i. This could be positive or negative as the situation is, if it is positive it is lateness, if it is negative it is earliness. Mathematically we can say that job earliness is E_i which is the maximum of zero and – L_i and job tardiness is nothing but T_i which is the maximum of zero and L_i.

(Refer Slide Time: 17:02)



Definitions II

Due Date, di

Job arrival time, ti

Start of Processing

Completion time, Ci

Job lateness, Li = Ci- di (could be positive or negative)
Job earliness, Ei = max (0, -Li)
Job tardiness, Ti = max (0, Li)

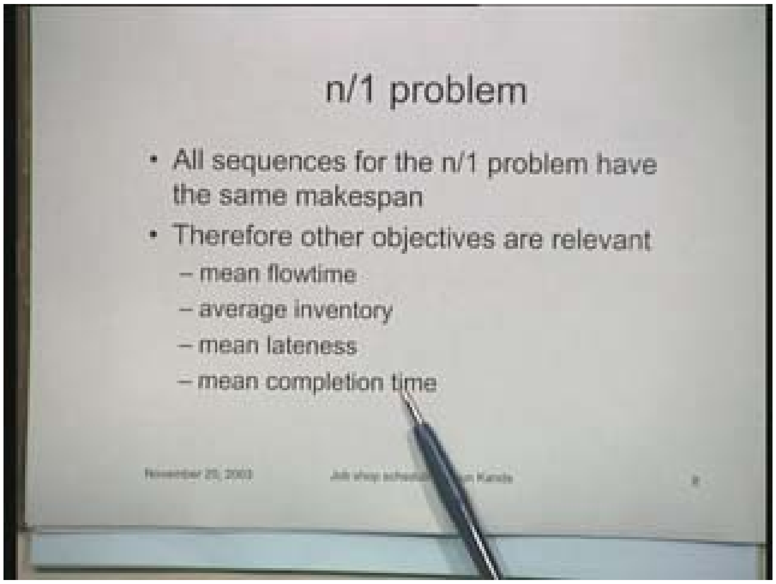
November 20, 2003

Job shop sche Arun Kanda

7

In that sense we can define job lateness, job earliness and job tardiness because these are also terms which you need to be clear about. Let us now consider the simplest problem and the problem for which some results are available. That is why we are looking at this problem. It is a n1 problem. That is there are n jobs to be processed on one machine. So, all sequences for the n one problem have the same makespan. Do you agree with this? What it simply means is that if you have n jobs to be performed and each job has a certain fixed time, no matter what sequence you choose the total amount of time required for doing the jobs will be the same. Really speaking therefore the makespan will be the same for all sequences. So that is not really a criterion for choosing the best value here and therefore other objectives are relevant in that situation. What are these objectives that you can talk about in minimizing the mean flow time. You can talk about minimizing the average inventory. Then you can talk about minimizing the mean lateness and then you can talk about minimizing the mean completion time.

(Refer Slide Time: 18:44)

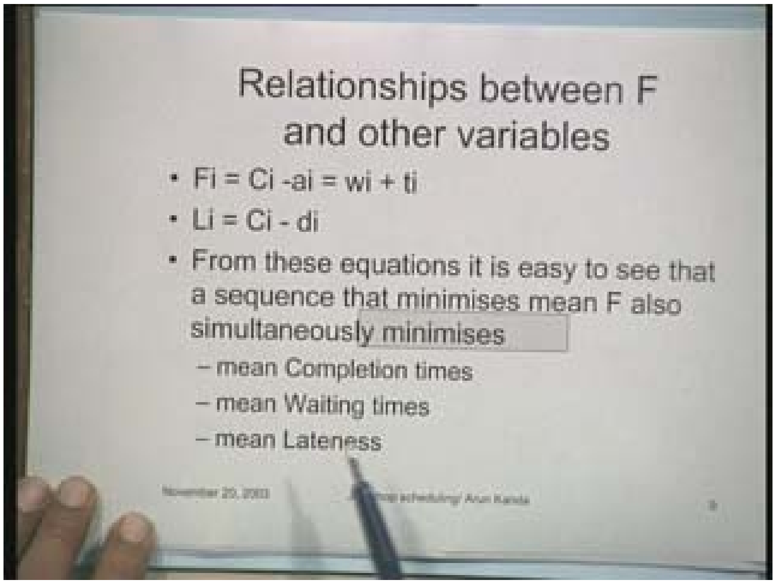


n/1 problem

- All sequences for the n/1 problem have the same makespan
- Therefore other objectives are relevant
  - – mean flowtime
  - – average inventory
  - – mean lateness
  - – mean completion time

It so happens that we can prove that the rule which tries to minimize anyone of these also minimizes all the others. That is a significant result. It is very much like saying that the rule which does this we will see which rule does it. We will actually minimize these various objectives simultaneously. Let us try to establish a relationship between the flow time f and other variables. What happens is that we had proved that the flow time is nothing but the completion time – the arrival time of a job and this is equal to nothing but the waiting time plus the processing time of the job and the lateness is nothing but $C_i – D_i$. That is the completion time – the due date of the job. From these equations, it is easy to see that a sequence that minimizes mean flow time also simultaneously minimizes a variety of other things. How? Mean flow time is equal to $C_i – a_i$, so this average of mean flow time is equal to average of completion time – the average of arrival times. Now the arrival times average will be a constant. The time for arrivals will be a constant because that is the time that the jobs came on the scene, so this is a constant. What it means is that any particular rule which minimizes the mean flow time will also minimize the mean completion time one. Secondly if you look at this particular equation here summation t_i would actually be a constant for any sequence for any sequence. What will happen is that a rule which minimizes f_i bar or mean flow time will also minimize the mean waiting time. That is a second result and here what you find is that if you are trying to minimize the mean completion time of a job because the due dates are constant in terms of their total values the rule will also minimize the mean lateness. Really what you have seen is that a rule which tries to minimize the mean flow time will simultaneously try to minimize the mean completion time, the mean waiting time and the mean lateness.

(Refer Slide Time: 21:57)

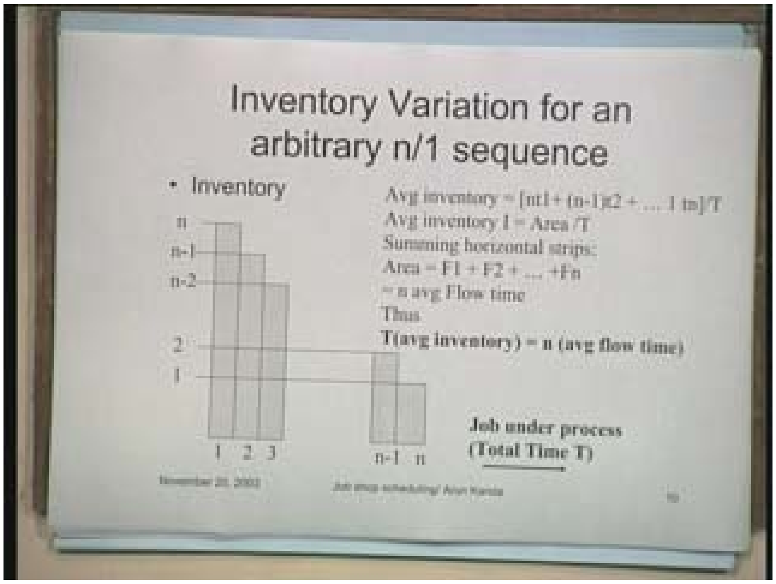


Relationships between F and other variables

- Fi = Ci -ai = wi + ti
- Li = Ci - di
- From these equations it is easy to see that a sequence that minimises mean F also simultaneously minimises
  - mean Completion times
  - mean Waiting times
  - mean Lateness

That is what we have proved here. Let us look at the inventory variation for an arbitrary n1 sequence that you have n jobs and one machine. What will happen? If we have the first job, the second job, the third job and so on up to the n th being completed, what is the average inventory? When the first jobs are being processed the number of jobs in inventory is n. Then once the first job is completed the number of jobs in inventory is only n – 1 because one job has gone out and then subsequently for the third job is going to be n – 1 and so on. Ultimately when the final inventory is just one job in that particular case, there is only the last job, i.e., n th job is in progress and that is the one which is actually waiting.

(Refer Slide Time: 23:08)



Inventory Variation for an arbitrary n/1 sequence

- Inventory

Avg inventory = [nt1 + (n-1)t2 + … 1 tn]/T

Avg inventory I = Area /T

Summing horizontal strips:

Area = F1 + F2 + … +Fn

= n avg Flow time

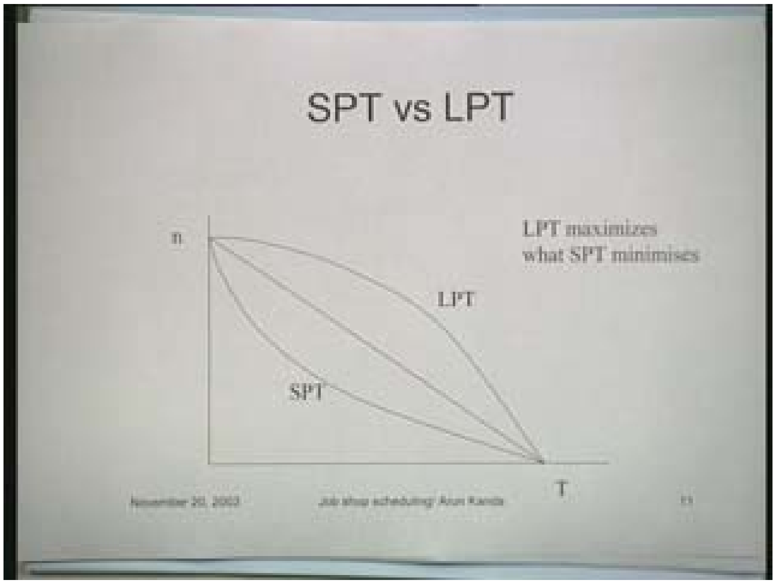
Thus

T(avg inventory) = n (avg flow time)

What you can see here is if you were to calculate the average inventory you can see the vertical strips here. Each of these is vertical ones. So time required in processing the first job n T_1, then n – 1 into T_2 which is the processing time for the second job and so on. This is 1 to the power T n an average inventory. This is the total inventory divided by the time period t which is the total time. So you have this, so the average inventory I is nothing but the area the total area under the strip divided by t that is what it is. It comes out here, and then if you sum up the horizontal strips that are these strips, the total area should work out to be the same. What does summing up the horizontal strip show? This is actually the flow time of the first jobs in the sequence. This is the flow time of the second job in the sequence, so this flow time means this is the waiting time and then this is the processing time and so on. So the total area is $F_1 + F_2$ and so on up to F_n. This is nothing but n times the average flow time and thus t into average inventory is equal to n into average flow time.

What is quite interesting from here is that the average flow time and the average inventories are nothing but directly related to the area under the curve. So if you want to minimize the average inventory or if you want to minimize the average flow, time how can you do it? The best way to do it is try to minimize the total area under the curve that is what it is. How do we try to minimize the total time for i mean, the total average inventory? It is basically something like this; we plotted a graph between the average inventory on side. So the maximum value is n and here the total time is t. We want to minimize the area. How would you minimize the area? Minimization of the area would mean that you should have some such curve here and this is what the SPT rule will do. The shortest processing time will do, that means if we use the SPT rule we are actually trying to minimize the area under the curve as shown here and this would actually ensure that you use the LPT. LPT will maximize whatever the SPT minimizes.

(Refer Slide Time: 26:13)



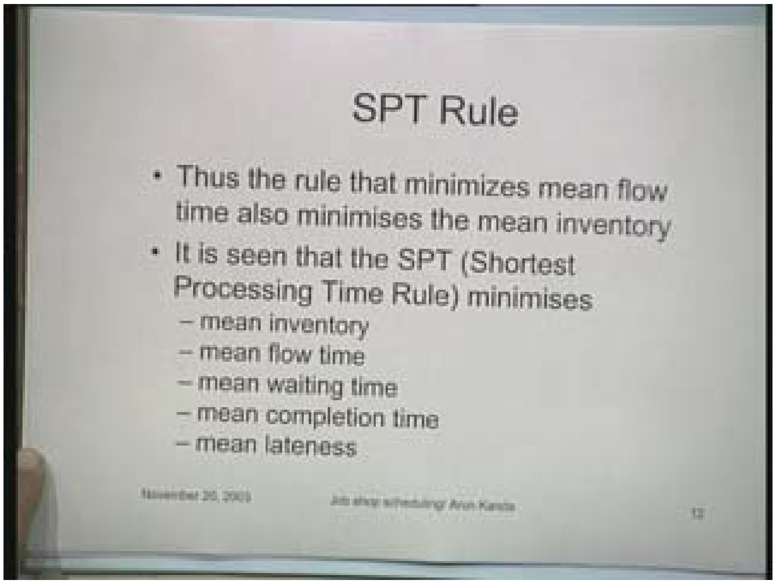
SPT vs LPT

LPT maximizes
what SPT minimises

So the SPT rule now is the golden rule. What is a golden rule? It minimizes the average inventory. It minimizes the mean flow time, it minimizes the completion time, it minimizes the average lateness and the average completion time. All the 4 criteria that we are talking about, is something like a 4 in one ice cream. One particular rule is the SPT rule, for this kind of situation will simultaneously accomplish those 4 objectives.

That is what we have tried to prove and so this is actually a summary of the final result that we have. The SPT rule is the rule that minimizes the mean flow time and also minimizes the mean in process mean inventory. It is seen that the shortest processing time rule minimizes the mean inventory the mean flow time the mean waiting time the mean completion time and the mean lateness simultaneously.

(Refer Slide Time: 27:26)

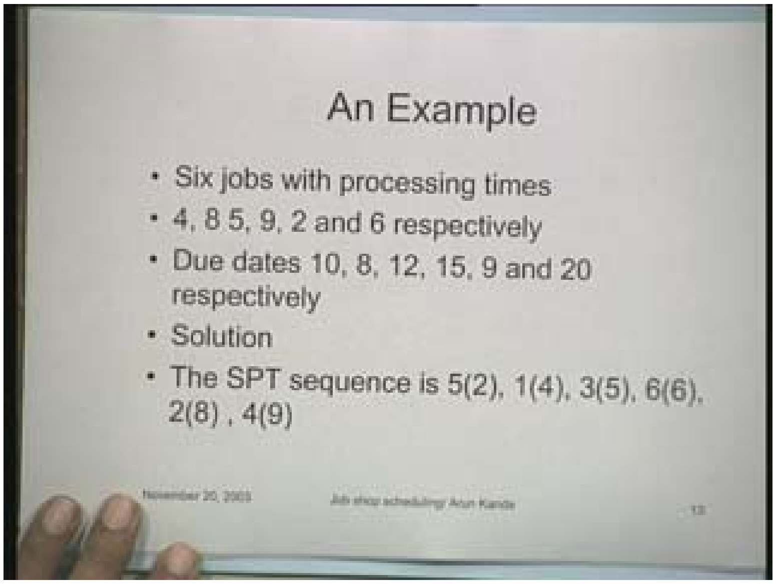


SPT Rule

- Thus the rule that minimizes mean flow time also minimises the mean inventory
- It is seen that the SPT (Shortest Processing Time Rule) minimises
  - mean inventory
  - mean flow time
  - mean waiting time
  - mean completion time
  - mean lateness

This is the advantage of using the SPT rule. Let us take an example and try to see what would be the implication of choosing a SPT rule. Let us say that there are six jobs with processing times 4, 8, 5, 9, 6 and 2 respectively. So six jobs have to be processed and these are the processing times available for the six jobs and the due fates for these jobs are 10, 8, 12, 15, 9 and 20 respectively. The final solution is therefore very simple. You will follow the SPT sequence and the SPT sequence means that you take the job which is the smallest processing time. So you take this job which is job number 5, so job number 5 has a processing time of 2. Then job number one has a processing time of 4, job number three has a processing time of 5, job number six has a processing time of jobs 6 and that is the answer. This particular job is job 8. It has the processing time of 8 and job number 4 has a processing time of nine. So job number 4 has a processing time of nine. So we have this particular situation here, this is the sequence in which the jobs would be done under a SPT rule.

(Refer Slide Time: 29:19)

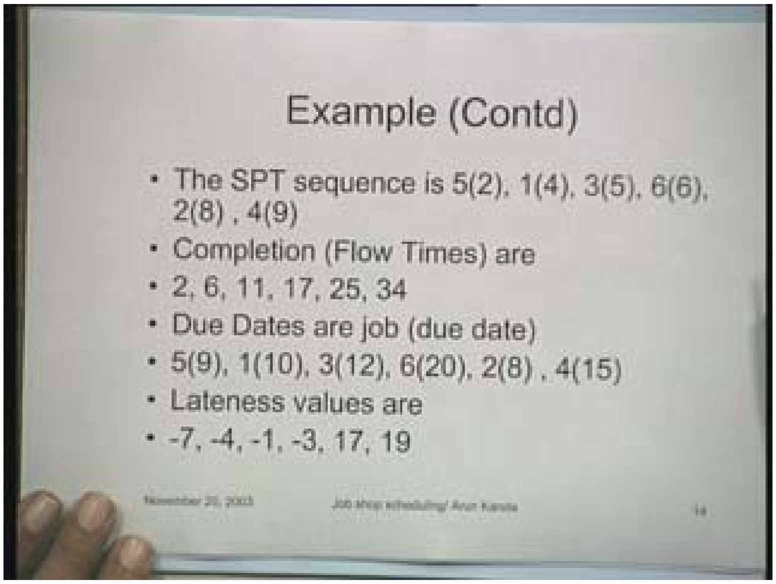


An Example

- Six jobs with processing times
- 4, 8 5, 9, 2 and 6 respectively
- Due dates 10, 8, 12, 15, 9 and 20 respectively
- Solution
- The SPT sequence is 5(2), 1(4), 3(5), 6(6), 2(8) , 4(9)

We say that the SPT sequence is this. The completion of the flow times is simple.
This job ends after 2. So you take the cumulative times, this is 2, then this is 4, which is 6, then after 6 it is 11, 11 + 6 = 17. 17 + 8 = 25; 25 + 9 = 34; so these are the completion times or the flow times of these particular jobs. Due dates are like given like this, for each job. So 5 1 3 6 2 5 and you are given these particular due dates and knowing the difference between C_i – D_i you get the lateness values. The lateness values for this particular example are – 7 – 4 – 1– 3. So these jobs are typically early and these 2 jobs are late.

(Refer Slide Time: 30:31)



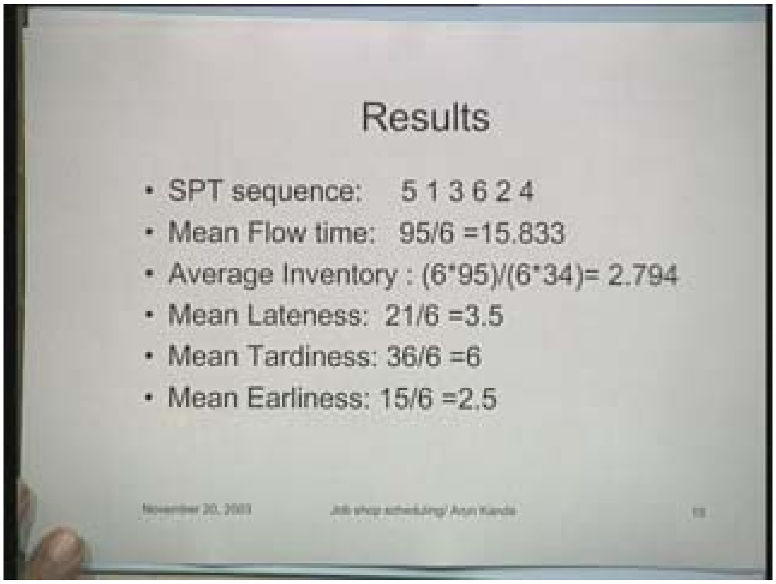
Example (Contd)

- The SPT sequence is 5(2), 1(4), 3(5), 6(6), 2(8) , 4(9)
- Completion (Flow Times) are
- 2, 6, 11, 17, 25, 34
- Due Dates are job (due date)
- 5(9), 1(10), 3(12), 6(20), 2(8) , 4(15)
- Lateness values are
- -7, -4, -1, -3, 17, 19

November 20, 2003 Job shop scheduling/ Arun Kanda 14

What we do in this example is we can look at some of these values and try to summarize the results for this particular case. The SPT sequence is 5 1 3 6 2 4. The mean flow time is 95 divided by 6 which is 15.833. The average inventory is 6 into 95 divided by 6 into 34. This actually comes directly from the formula that we derived about the total area under the curve for inventory. The average inventory is 2.794 jobs. The mean lateness is 3.5. The mean tardiness is six and the mean earliness is 2.5.

(Refer Slide Time: 31:42)



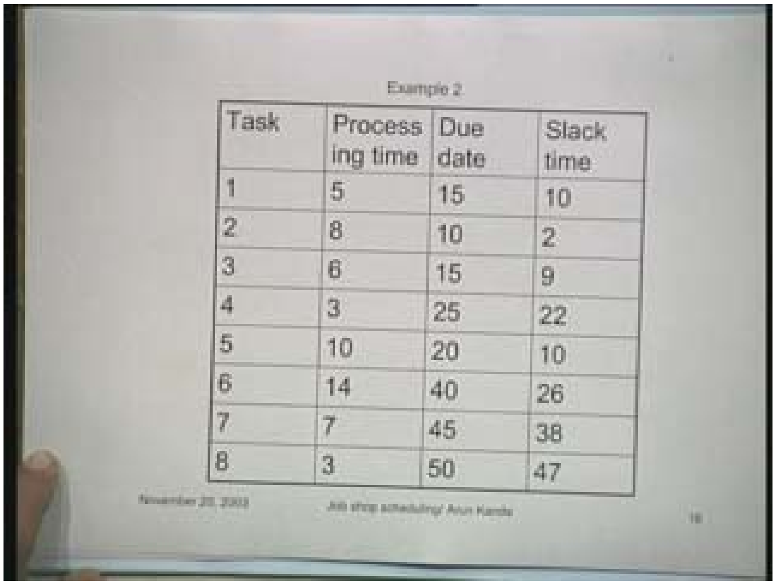
Results

- SPT sequence: 5 1 3 6 2 4
- Mean Flow time: 95/6 =15.833
- Average Inventory : (6*95)/(6*34)= 2.794
- Mean Lateness: 21/6 =3.5
- Mean Tardiness: 36/6 =6
- Mean Earliness: 15/6 =2.5

November 20, 2003 Job shop scheduling/ Arun Kanda 15

This is how you could summarize the performance of a particular rule. In this case of SPT rule and what we will do now is we will now take another example and in this example if we have let us say 8 jobs to be done, the processing times for each of these jobs is given and the due dates are given and the corresponding slack time is nothing but the difference between the due date and the processing time. So 15 – 5 = 10 and that is how we define slack time. 10 is the due date minus 8 which is the processing time. This is 2 and so on.

(Refer Slide Time: 32:32)



Example 2

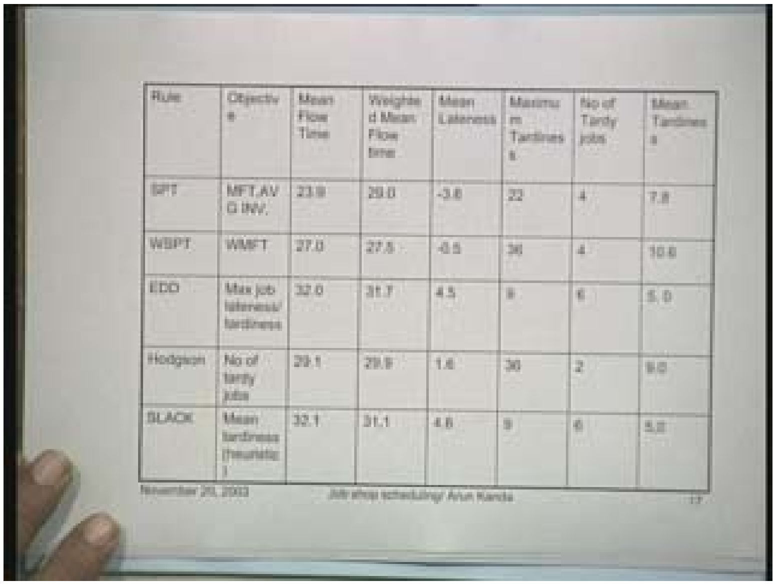
Task	Processing time	Due date	Slack time
1	5	15	10
2	8	10	2
3	6	15	9
4	3	25	22
5	10	20	10
6	14	40	26
7	7	45	38
8	3	50	47

November 20, 2003 Job shop scheduling/ Arun Kanda

The important thing is that we will now state some other results without actually going into proofs. But what we will do is that to the same problem if we apply different kinds of rules we might apply for instance the SPT rule and the SPT rule tries to minimize apart from other criteria like mean flow time, the average inventory and so on. So the corresponding value of the mean flow time, the w8ed mean flow time, the mean lateness, the maximum tardiness, the number of jobs and so on is like this. Another possible variant of this job would be the WSPT rule. That is the w8ed SPT rule. In the SPT rule we do not assign any w8ages to the times. We have w8ed mean flow time if that is the objective. This would happen if you take certain w8ages to be applied to different rules. What you do is you take T_1/W_1. T_2/ W_2, T_3/ W_3 and so on, so if you are taking higher w8age you are actually reducing the value of T by W. So it gets a higher priority in that sense. That is the w8ed SPT rule and for this, this is the performance measure and the implication of the w8ed SPT is that it tries to minimize the w8ed mean flow time rather than the mean value. The third rule that is commonly talked about is the earliest due date rule, EDD rule.

In the EDD rule what happens is this particular rule will try to actually maximize the job lateness or it will try to actually minimize the job lateness and the minimum job tardiness in that sense of the term. The corresponding values of the mean flow time the w8ed mean flow time, the mean lateness and the tardiness and the number of tardy jobs are actually shown here. Then the 4th kind of rule which is important is what we call Hodgson's rule. In hodgson's rule the objective is to minimize the number of tardy jobs. Try to minimize the number of tardy jobs, so corresponding to this rule you have the mean flow time the w8ed mean flow time, the mean lateness, the maximum tardiness, the number of tardy jobs and the mean tardiness values which are shown here and then finally the slack. Slack is nothing but the mean tardiness which is a heuristic. This tries to minimize the mean tardiness and these are the corresponding values.

(Refer Slide Time: 35:44)

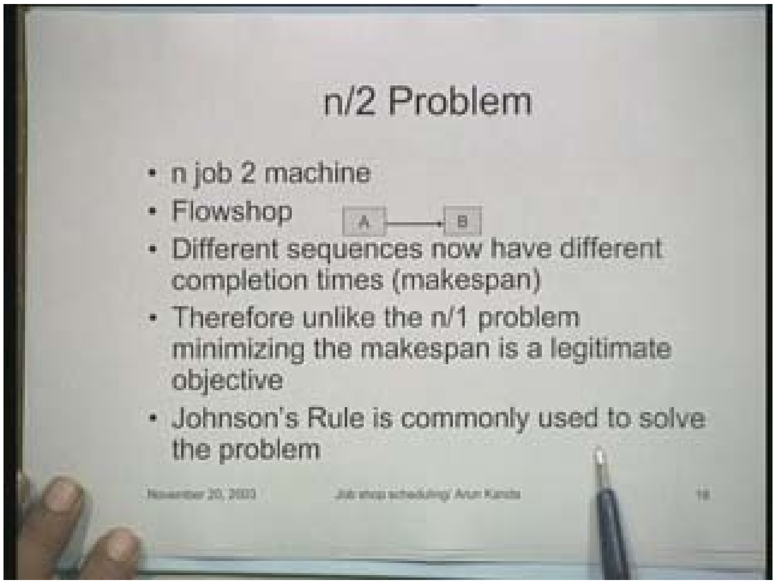


Rule	Objective	Mean Flow Time	Weighted Mean Flow time	Mean Lateness	Maximum Tardiness	No of Tardy jobs	Mean Tardiness
SPT	MFT,AVG INV.	23.9	29.0	-3.6	22	4	7.8
WSPT	WMFT	27.0	27.5	-0.5	36	4	10.6
EDD	Max job lateness/ tardiness	32.0	31.7	4.5	9	6	5.0
Hodgson	No of tardy jobs	29.1	29.9	1.6	36	2	9.0
SLACK	Mean tardiness (heuristic)	32.1	31.1	4.6	9	6	5.0

November 20, 2003 Job shop scheduling/ Arun Kanda 17

What we have basically tried to show here is that just in the case of the SPT we considered the proofs. In the other cases we are simply stating the results without proofs that WSPT will minimize the mean w8ed mean float time. The earliest due date will try to minimize the maximum job tardiness or the maximum job lateness. Hodgson's rule will try to minimize the number of tardy jobs. Actually hodgson's rule is something derived from the EDD rule. What is the EDD rule? The earliest due date rule which means, once you know that the jobs are arranged according to the earliest due date you will find out the job which is first late. The job which is first late is then discarded from the scene and kept in the set of jobs called the set of late jobs and we then keep on doing this exercise till this process ends, when you have a minimum number of late jobs. After the n 1 problem that means you have n jobs and 1 machine, we are now considering a situation where there are n 2 jobs, 2 jobs and 2 machines and it is a flow shop situation. If it is a job shop situation we do not have an exact solution procedure for that particular case. As I indicated to you we are discussing only those situations for which exact solutions are available. What is a flowshop? Flowshop means that if this is machine A and this is machine B then all jobs would be processed first on machine A. Then on machine B, they will not have any arbitrary sequences. That is what you mean by a flowshop and different sequences now have different completion times or different makespans and therefore unlike the n 1 problem minimizing the makespan is a legitimate objective. You see now for the n 2 case, any arbitrary sequence will not actually minimize the total makespan.

(Refer Slide Time: 38:41)



n/2 Problem

- n job 2 machine
- Flowshop

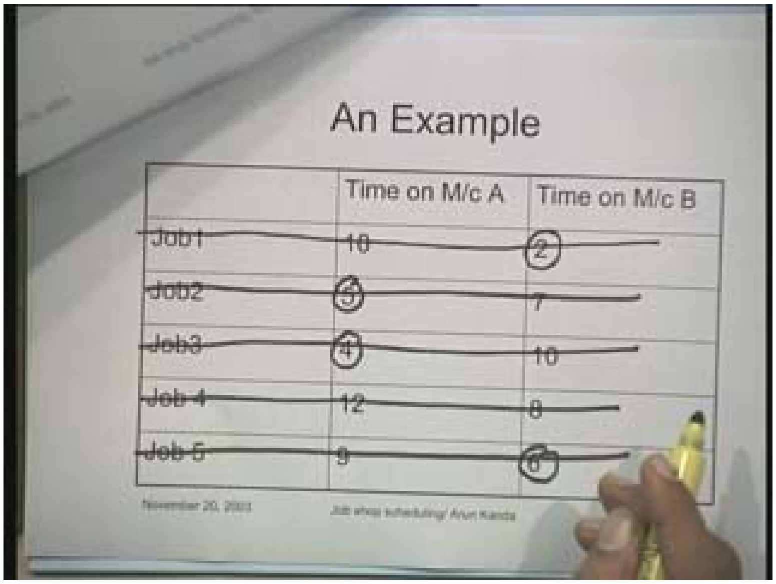
- Different sequences now have different completion times (makespan)
- Therefore unlike the n/1 problem minimizing the makespan is a legitimate objective
- Johnson's Rule is commonly used to solve the problem

So the rule which is commonly used for solving this type of problem, the n 2 problem is actually the one which is Johnson's rule and it is easily used to solve the problem. Have you had an exposure to Johnson's rule earlier? How we actually solve the Johnson's rule is actually suppose we have this problem and the problem is that we have let us try to solve this problem. Job one requires 10 units of time on machine A and 2 units of time on machine B in that sequence because it a floor shop. Similarly job 2 requires 5 on a 7 on B, job 3 requires 4 on a 10 on b. Job 4 requires 12 on a 8 on b and job 5 requires 9 on a and 6 on B. So how do you solve a problem like this? How would you look at the problem? Let us try to solve this particular problem because what we can do is suppose that we have a problem like this, we are saying essentially job 1, so you try to find out the minimum processing time. What you notice is a minimum processing time occurs on job one. This is the minimum here. This is also the minimum here. So this figure is a global minimum here. What happens is that if this particular time happens to be on B, we will schedule this job last. If it happens to be on machine A then we will schedule this job first.

What we will do is we have a table in which we can directly say job 1 should be last. You put job one last. We do that here and having done that now after this job 1 has been processed, what we can do is we can cross out job one from the list because it has been done and then from the remaining times, we try to identify which is the smallest value. So the smallest value is now 4. This is the smallest value. This means that job 3 will now be done first. So you do job 3 here. This is what we do and then of course this particular job has been scheduled. So you can cut out this job like we did and from the remaining times again try to find out which is the lowest. The lowest value now is 5 which means since this occurs job 2 for job 2, so job number 2 is scheduled here. Once we have scheduled job number 2, you can cross out job number 2 because these are the 3 jobs which have already been placed and looking at this particular situation here, what is the minimum now. The minimum is 6, so job number 5 should be scheduled last in the available slot.

So job number 5 should be scheduled last here and once it has been scheduled then we can cut out job number 5 from the list and then of course the minimum is 8. Job number 4 should turn out last; in fact there is only one slot. So it goes here and what have we been able to find out and then of course all the jobs are now assigned.

(Refer Slide Time: 43:16)



An Example

	Time on M/c A	Time on M/c B
~~Job1~~	~~10~~	~~2~~
~~Job2~~	~~5~~	~~7~~
~~Job3~~	~~4~~	~~10~~
~~Job 4~~	~~12~~	~~8~~
~~Job 5~~	~~9~~	~~6~~

November 20, 2003

Job shop scheduling/ Arun Kanda

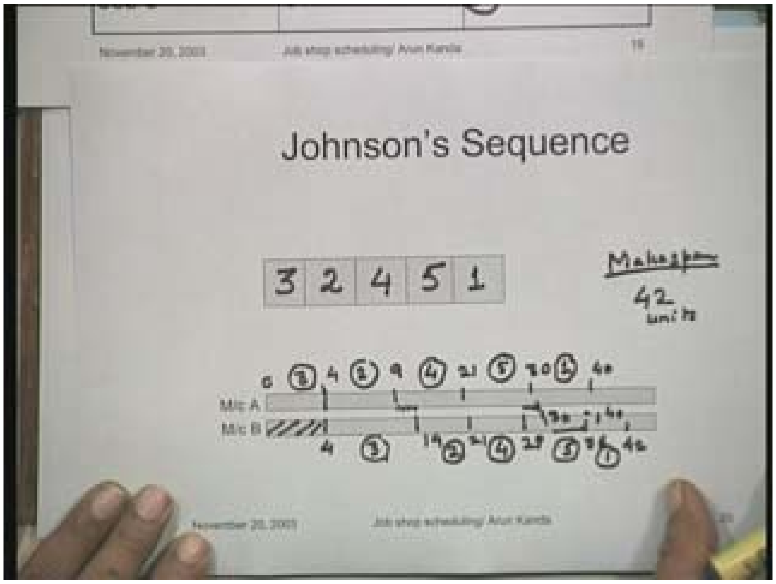
This particular procedure that we have just followed is called Johnson's sequence and what you find is that by applying Johnson's rule, we have been able to determine a sequence, 3 2 4 5 1 and this particular sequence ensures that the total makespan or the time required for doing all the jobs is actually minimum.

So you can find out exactly how the time would be a minimum. What we need to do is simply we need to draw a gantt chart to be able to determine this particular value. How is it done? In fact it is done in this manner. You know 3 2 4 5 1, so job number 3 has a processing time of 4. We can go from time 0 to 4 and this is the time required for job number 3. Job number 3 on machine 1 will go from time 0 to 4 in that sense of the term, then you have job number 2. Job number 2 has a processing time of 5. So 5 + 4 = 9, this is time 9 and this is processing sequence 2. So after three we process job number 2, so we get 3. Then for 4, a processing time for job number 4 is 12. So 12 + 9 = 21 and this is actually job number 4 and then subsequently when job number 5 is concerned, job number 5 would require a total time of 9.

This will be 30 and this will be 5. W have 3 2 4 5 3 2 5 and 1. This is what we are doing on machine A and what is the processing time for 1? Job number one requires ten units of time, so what you will have is 40 here and this is 1. All that we know through a gantt chart is the time that the job enters, which job enters i.e., job number 3 from 0 to 4, job number 2 from 4 to 9 and so on. So 3 2 4 5 1 and it comes out at time 40. However in order to tell us the complete time at which this job will be over you have to talk about processing on machine B. So processing on machine B is done in the same sequence. So what happens is really that as far as job number 3 is concerned it has a processing time of 10 on machine B and that means 4 +10 = 14, so you going to be some somewhere here. This is 14 and this is 4. Obviously up to this particular time from zero to 4 this machine B is going to be idle and 4 to 14 is the time required for job number 3, so this is job number 3 which is actually being done on. So it requires ten units of time here, then we are talking about job number 2. When you talk about job number 2 it has a processing time of 7, so what is going to happen is that this job which is actually over a time 9 and will have to wait up to this particular time because the machine is not available during this particular time. Then at time 14 you can start processing job number 2. Job number 2 will take time of 7. So exactly at time 21 you will have job number 2 which will be processed from 14 to 21 on this particular case. Then you take job number 4, job number 4 requires a processing time of 8 units. What happens is this is at 21 and the job does not have to wait now. What is going to happen is that job number 4 will require a processing time of 8. It will go up to 29. It will be over here and this is now job number 4 3 2 4 and at time 21, then what is going to happen? The next one, this machine is over at 29 but this job has not been completed yet. It is processing on the first machine. So this will be a situation where the machine is going to be idle.

This is the machine which is idle during this particular point of time and at that particular point of time at time 30. This job number 5 is going to be available and then it will take on that means. It is available at time 30 and from time 30 onwards until, that means we are now talking about job number 5. The total processing time is now 6. So up to time 36 we will go on up to here and this is now job number 5. Then we talk about number, job number 1 is being processed till time 40 whereas this machine will again be idle up to 40. As far as job one is concerned it would actually be over at 42 and this is therefore job number 1. You have the complete picture. What it shows is how the 2 machines are going to be loaded with different types of jobs according to Johnson’s rule. What we find is that the completion time or the makespan for this job is now 42 units and this is the makespan and the guarantee that we have from Johnson’s rule is that there is no other sequence which will reduce the time required for doing all the jobs in lesser than 42. That is the guarantee. This is how you can use Johnson’s sequence and the notion of gantt charts to actually identify the total time in which the jobs are going to be done.

(Refer Slide Time: 50:36)



Johnson's Sequence

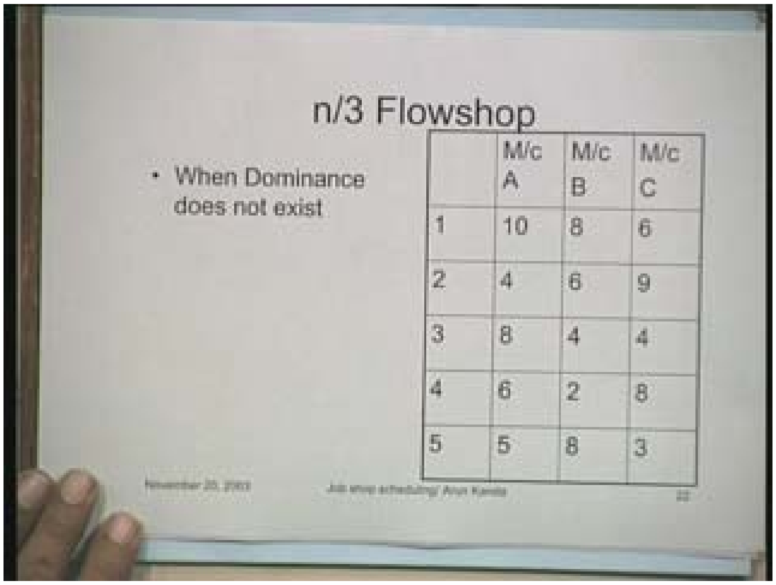
3	2	4	5	1

Makespan
42
units

We know that the makespan is now 42 units. What we can now try to explain is in fact this particular situation where we have the n 3 flowshop situation. That means we are now talking about n jobs, 3 machines to be processed on a flowshop. What does it mean? That there are three machines A, B, C the processing has to be done for the n jobs first on A then on, A then on C. So it is a flowshop situation and this is the kind of situation. So what would happen is something very similar. Actually this rule that we are talking about now is called Jackson's extension of Johnson's rule. Very interestingly it is like 2 brothers who have solved the problem. So what you have is that the first job requires a processing on a processing on B, processing on C, similarly 2 requires processing on and so on. Now what happens is that in case there is dominance, meaning that let us say the minimum of the time on A, the minimum of the time on A is 8. This is the minimum time on A and the maximum time on B which is the middle machine. The maximum time is also 8. When the minimum on A is greater than equal or equal to the maximum on B this is said to be dominance.

What can happen is that machine B could be dominated either by A or by C and if that happens like this is a dominant situation. Then you can convert this problem into an equivalent Johnson's rule problem and solve. How? You can take A + B together and B + C together. That means you can take 2 fictitious machines A +B and B + C, then once you have done that you can look at that means what we are trying to say is that we can combine A + B and B + C, we will have 2 columns and we can solve the same problem by using Johnson's rule and get the same sequence and then try to apply the same sequence now to three machines. It will be a gantt chart which will have three machines and it will give you the minimum makespan for this problem. However it has been seen that even when dominance does not exist, that means here what you find is that the minimum is 4 and the maximum is something else which is 8. There is no dominance here and no dominance here, even in such a situation you can combine A and Band B and C, you still have 2 fictitious machines, solve it and the general conclusion is that the solution is generally found to be very close to optimum even when dominance does not exist although it is not optimum in general.

(Refer Slide Time: 54:02)



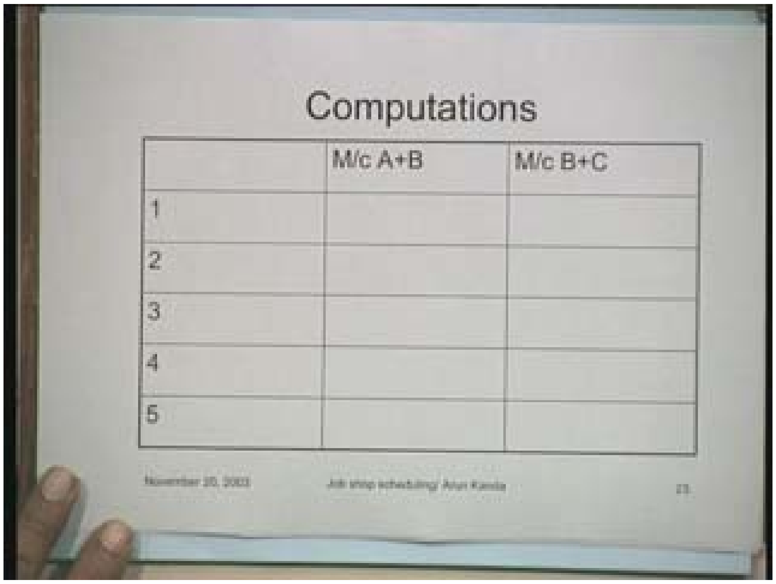
n/3 Flowshop

- When Dominance does not exist

	M/c A	M/c B	M/c C
1	10	8	6
2	4	6	9
3	8	4	4
4	6	2	8
5	5	8	3

That is what you tend to find in this kind of situation here. As far as computations are concerned all that you need to do is the time required for A and B, you put it down here. The time required for B and C, put it down here and do complete this table. Once you have this table completed, solve it exactly like a Johnson's rule to with 2 fictitious machines and then the sequence which you determine as a consequence will in fact be the one which will try to minimize the makespan.

(Refer Slide Time: 54:46)



Computations

	M/c A+B	M/c B+C
1		
2		
3		
4		
5		

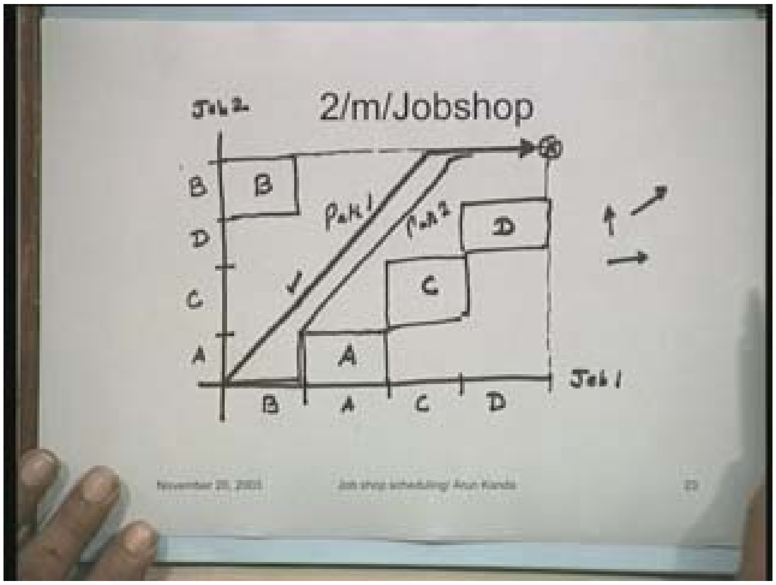
You can just solve the problem in this particular way. This can apply both to the case, when there is no dominance and also to the case when there is dominance. Both the cases that we considered, only thing is when there is no dominance, then you have this particular situation. The next problem that we can solve for this particular situation is a job shop situation. A 2 m job shop situation means that there are 2 jobs, there are m machines and there is a job shop situation. That means sequence is the same. This problem can be solved graphically. It is a very simple graphical solution. For instance what you can do is you can plot 2 jobs on 2 axes. It is a 2 job situation, so you can say job 1 and job 2 in this manner and then what can happen is that each of these jobs can require processing on different machines.

You can have for instance if suppose there are machines which are like B A C D. Let us put it this way, that means what we were saying is that for job number 1, this is the time required for processing on machine B, this is the processing time required on machine A and so on. So we write down these times in the sequence in which the processing is required. This is for a job shop and you can also find out the same thing here. For instance you might say that this is A, then this is C, then this is D and then this is B in that sense. This is the sequence in which job 2 requires processing on different machines, so it is a general job shop situation. What you then do is something very simple. This is A so you can mark out this area. Let us say here this is called A, then as far as B is concerned you can mark out this area here and look at this area and call out this B.

Then as far as C is concerned you have this area here and this area here and this is the area that you can mark out as C and as far as D is concerned you have this area and then you have this area here and you have this area here. So this is let us say and whatever you are trying to do achieve you are simply trying to say that as far as these jobs are concerned you will finish here and you start from here. You have this particular situation, now why we put this time is that here in this region both job A, that is job 1 and job 2 are both requires machine A. This is not possible. Similarly here both so we have to basically avoid these regions and what we have to do is determine a path from here to here. The path can possibly be in this case.

In fact there is no difficulty we can go from here and ultimately come here. So as far as path is concerned we have got path number 1, you can also look at the other possibility; you can say for instance you can come here. You can go here and then take a path like this and come here. This is path 2. The basic idea here is that a path can consist of either a horizontal movement or a vertical movement or a 45 degree movement. Horizontal movement means job one is being processed, this means job 2 is being processed. This means both are being processed simultaneously. So you would then after having enumerated a couple of paths we are interested in that particular path which will minimize the total distance. So by looking at this diagram you can possibly identify like in this case this is the shortest path.

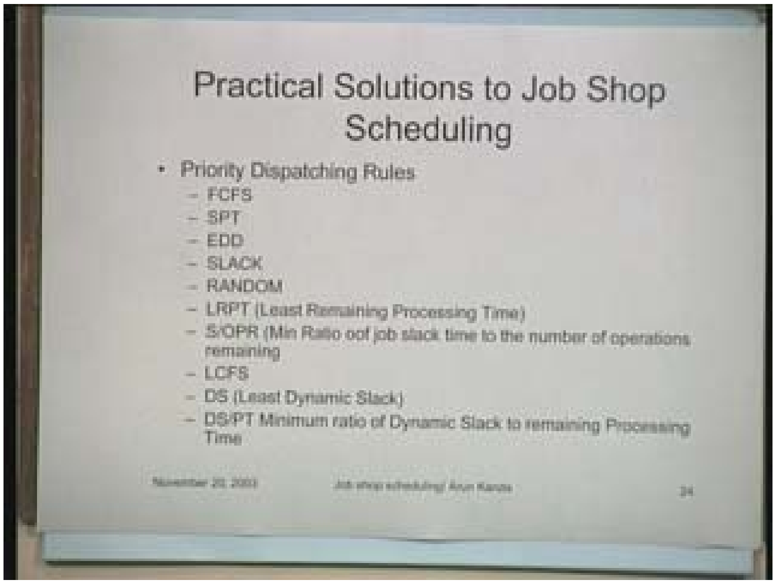
(Refer Slide Time: 59:48)



2/m/Jobshop

This is the procedure that can be followed for determining the solving the 2 M job shop problem. In practical solutions to big job shops, you use a variety of dispatching rules. A list is here, first come first serve, SPT, EDD, slack random least remaining processing time, minimum ratio of slack time to number of operations last come first serve least dynamic slack and so on.

(Refer Slide Time: 01:00:18)



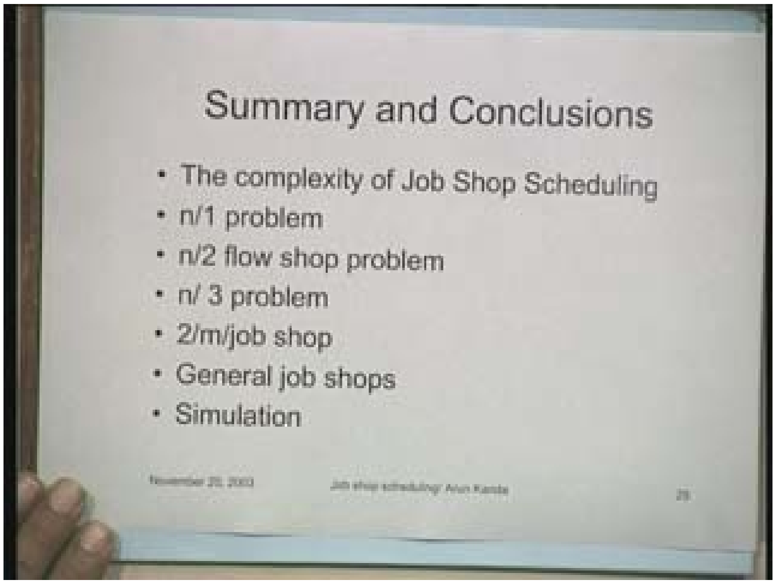
Practical Solutions to Job Shop Scheduling

- Priority Dispatching Rules
  - FCFS
  - SPT
  - EDD
  - SLACK
  - RANDOM
  - LRPT (Least Remaining Processing Time)
  - S/OPR (Min Ratio oof job slack time to the number of operations remaining
  - LCFS
  - DS (Least Dynamic Slack)
  - DS/PT Minimum ratio of Dynamic Slack to remaining Processing Time

November 20, 2003 Job shop scheduling/ Arun Kanda 24

What we are simply trying to say is that what is normally done is that through a process of simulation you try to identify what is the best you can do and then try to take that particular scheduling rule which gives the best solution. Let us try to summarize what we have tried to do in this lecture. We have looked at the complexity of the job shop scheduling problem. We have looked at the n one problem where the objective was not to minimize the makespan but also but to consider average inventory and other things. The n 2 flow shop situation, Johnson's rule was a solution for this case. n 3 problem too could in fact be solved by a variance of the job shop scheduling problem. 2 M job shop problem can be handled graphically. General Job shops and simulation could actually solve this particular case.

(Refer Slide Time: 01:01:14)



Summary and Conclusions

- The complexity of Job Shop Scheduling
- n/1 problem
- n/2 flow shop problem
- n/ 3 problem
- 2/m/job shop
- General job shops
- Simulation

I think with this we conclude our discussion on job shop scheduling.

Thank you very much!