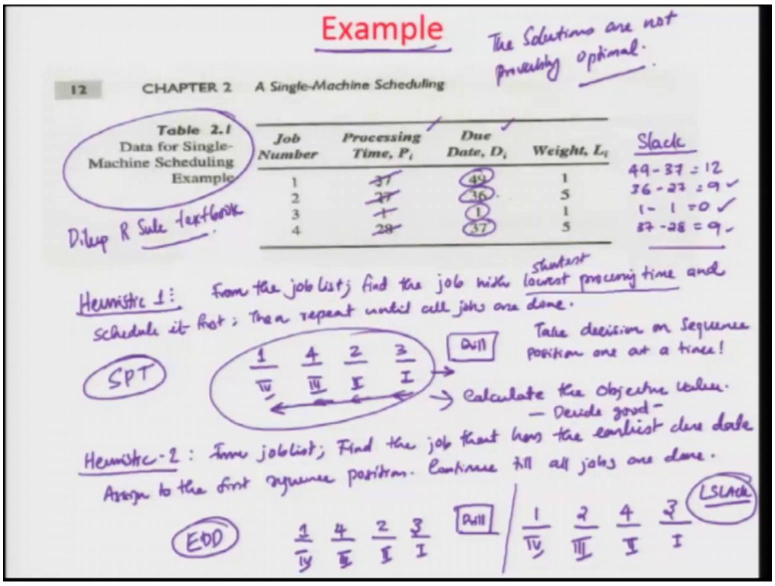
Advanced Green Manufacturing Systems

Prof. Deepu Philip
Dr. Amandeep Singh Oberoi
Department of Industrial & Management Engineering
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

Lecture - 40
Overview of Optimization Methods – Part 2

Good afternoon students, welcome to one other lecture in the Advanced Green Manufacturing Systems. So, continuing on with the simple heuristics and why heuristics are used because here they can start with one variable (Refer Time: 00:29) the solution without losing the feasibility and once all the variables are taken care of then that final solution which is feasible, then most of the algorithms stop. So, let us look at this an example.

(Refer Slide Time: 00:40)



Example

The Solutions are not provably optimal.

12 CHAPTER 2 A Single-Machine Scheduling

Table 2.1
Data for Single-Machine Scheduling Example

Dilep R Sule textbook

Job Number	Processing Time, P_i	Due Date, D_i	Weight, L_i
1	~~37~~	49	1
2	~~27~~	36	5
3	~~1~~	1	1
4	~~28~~	37	5

Slack
49 - 37 = 12
36 - 27 = 9 ✓
1 - 1 = 0 ✓
37 - 28 = 9 ✓

Heuristic 1: From the job list; find the job with ~~lowest~~ shortest processing time and schedule it first; Then repeat until all jobs are done.

SPT

$\frac{1}{IV}$ $\frac{4}{III}$ $\frac{2}{II}$ $\frac{3}{I}$ → Dull

Take decision on Sequence position one at a time!

→ Calculate the objective value. — Decide good —

Heuristic-2: From joblist; Find the job that has the earliest due date Assign to the first sequence position. Continue till all jobs are done.

EDD

$\frac{1}{IV}$ $\frac{4}{III}$ $\frac{2}{II}$ $\frac{3}{I}$ Dull

$\frac{1}{IV}$ $\frac{2}{III}$ $\frac{4}{II}$ $\frac{3}{I}$ LSLACK

The previous case example, here this example is taken from Dileep R Sule textbook titled Industrial Scheduling and this I am using this example to just demonstrate this particular case. So, let us say there is a heuristics. So, in this case there is two things one is a processing time and there is a due date; so, heuristics 1 ok.

The first heuristic we thing about from the list of the job from the job list find the job with lowest processing time and schedule it first ok, then repeat until all jobs are done ok.

So, what we are doing is let us say we have a four sequence positions. So, let us call this as sequence position I II III IV and here is the drilling machine. So, drill I means it is close so, the first one that is going to the drilling machine right. So, we go to the job and find out the processing time which is the one with the lowest processing time that is 3. So, it goes in these sequence position right.

Then this is over. So, we do not consider this anymore done. Now let us cut this one out. The next one is we go through this list and find which is a next lowest one which is 27. So, what we put here is 2, this is done. Then we go through the list we find that the next lowest processing time is 4 28 so, this goes here and then we find out the last one is this one. So, this goes into 1 ok.

So, 3 2 4 1 is the so one now we are done with all the jobs and now we have a sequence. So, we iteratively built a sequence in this direction. So, take decision on sequence position one at a time ok. So, each sequence position one after another we take then we move to this, then we move to this and we move to this ok.

So, this type of a heuristic what we just did is wherever we find out the lowest or shortest processing time or shortest this process is called as SPT Shortest Processing Time this is also sometimes known as a single pass heuristics because in one pass you are able to get a solution ok.

Another example of this is what we call as heuristic 2 you can call it as now from job list find the job that has the earliest due date. Assign to the first sequence position, continue till all jobs are done ok.

So, how will we do that? If we think about it we say again have the four sequence positions I II III IV and here is your drilling machine ok. So, now, we go through the due dates. The due date the job with low the earliest due date in this case is one. So, you take that one and assign job 3 right here ok.

Then the next one that we think about it is 36. So, that is job 2 this is then job 4 and then we get to job 1 ok. So, in this particular case you can see that the sequence of this kind of this heuristic is called as earliest due date ok. So, that is another example.

The third way you can do is you can calculate something called as a slack. Slack means what is the difference with. So, the slack of this job is this is the due date minus processing time 49 minus 37 there is a one slack.

This is 36 minus 27, then this is 1 minus 1 that is equal to 0, then 37 minus 28. So, in this case is 28 plus 38 so, it is 9, then this 1 minus 1 is 0 this also 9, then that this is 12 ok.

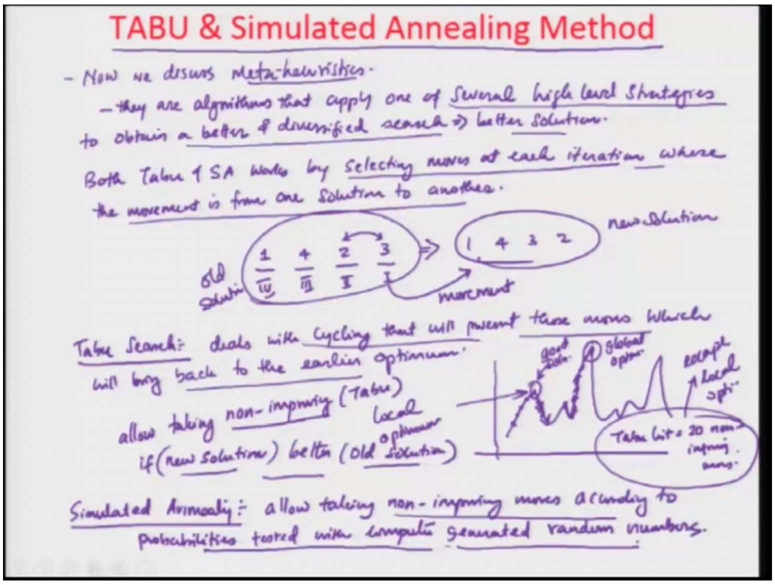
So, what we do is then find out the. So, first we calculate the slack and then take the job with the least amount of slack and then assign to the sequence position. So, then you can get in a probably get a different sequence in this regard ok. So, those kind of ways you can do this.

So, if you think about this approach then you your sequence position will be I II III IV four sequence positions. The first one you assign is job 3, second one let us say we picked up job 4 in this regard because this is the least slack. So, we picked job 4 then we picked this guy. So, it is job 2 and then there is job 1 ok. So, this gives you a different sequence this is called a L slack approach heuristic or least slack heuristic right.

So, there are many approaches that you can use to solve this problem and each one of them will give you one different solution. So, what you do is once you find the solution calculate the objective function value so decide good. Sometimes what people do is they take different heuristics like this 3, 4 different heuristics and then from that they decide which result is better compared to other kind of cases.

So, these kind this is what we call as a heuristic where these solutions all these solutions these solution and these solutions. The solutions are not provably optimal; we cannot really prove that the solutions are optimal that is the most important part of such an approach in this regard.

(Refer Slide Time: 07:36)



TABU & Simulated Annealing Method

- Now we discuss Meta-heuristics.
  - they are algorithms that apply one of several high level strategies to obtain a better & diversified search ⇒ better solution.

Both Tabu & SA works by selecting moves at each iteration where the movement is from one solution to another.

Tabu Search: deals with cycling that will prevent those moves which will bring back to the earlier optimum.

allow taking non-improving (Tabu)

if (new solution) better (old local optimum solution)

Simulated Annealing: allow taking non-improving moves according to probabilities tested with computer generated random numbers.

Now, from here we move to. So, we already talked about the heuristic approach now we get into what we call as TABU and Simulated Annealing Method not going to get into examples in this case, but I am just going to familiarize with you because there is another presentation by Dr. Amandeep Singh in this regard which will help you to understand this process better.

So, now we discuss meta heuristics ok. So, heuristics is something like where you start taking one decision variable and then you incrementally keep on doing it until you get all without violating the feasibility and once you get the feasible solution you call it good.

So, then we talk about something called meta heuristics; they are algorithms that apply a one of several high level strategies to obtain a better diversified say search to obtain a better and diversified search better solution. So, what we this algorithm see meta heuristics the difference between the normal heuristic and meta heuristics is that they have a complicated high level strategy to get the better and diversified solutions for these kind of problems ok.

So, both TABU search and Simulated Annealing SA works by selecting moves at each iteration where the movement is from one solution to another ok. The idea here is that, you select a move for each iteration where the movement is from one solution to another. So, like for example, if you think about the previous case you have four sequence positions if you look into this one is 1 4 2 3. So, we say 1 4 2 3 and now we want to find out one of the way to do it is, we say we exchange these positions.

So, we move to a next solution called 1 4 3 2 and see whether this solution is, this is the new solution this is called movement ok. So, you move from this particular solution this is the old solution.

So, you move from one solution to another you are you are not really changing one thing you are basically actually moving from one solution to another right and until you find out which is a best possible solution. So, the TABU search on the other hand what does TABU search do? Deals with cycling that will prevent those moves which will bring back to the earlier optimum ok.

So, what we do is we try to this algorithm prevents the cycling. So, TABU moves are something that you are not supposed to take ideally speaking. So, if I draw a graph let us say the solution space of this problem is something like this and let us say you started from here and you built a solution. So, you move from one solution to another you keep on moving, moving, moving until you reach this particular point ok.

So, this say is fine this is a one good solution. The parlance for this good solution is called as local optimum we call this as a local optimum ok. So, from here what happens is there is no other way in any of this area even if you start from here you will go back to this optimum only.

So, what is the way to get out of this optimum and reach this particular one let us say this is you know global optimum how do you do that? The one way to do is allow it to reach take this moves allow taking non improving this is called as a TABU ok, by taking this move you are actually deteriorating the solution right.

The comparison is if new solution better old solution is a new solution better than the old solution that is what we are actually comparing in a meta heuristic and yes then you take this. But in a TABU move you allow them to take this moves which are which does not allow this to uh satisfy. So, that is called as a TABU search ok.

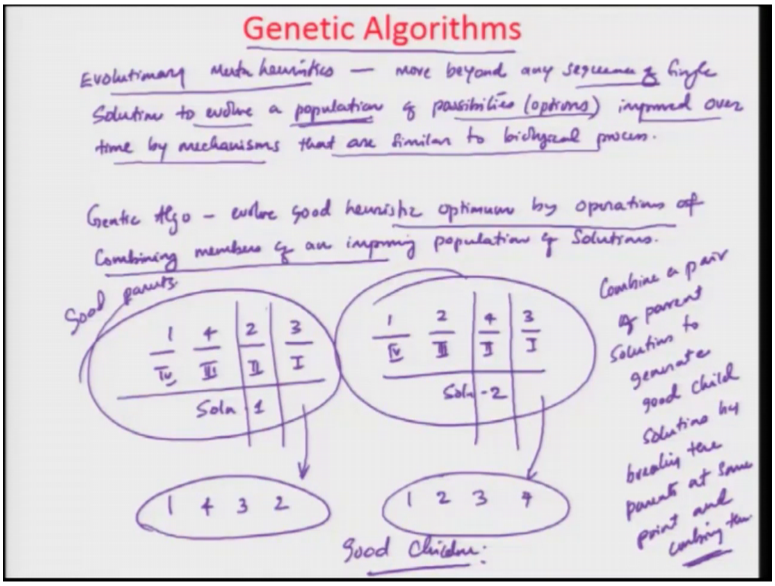
And Simulated Annealing on the same way is also of the same type. So, the Simulated Annealing is that allow taking non improving moves you allow the algorithm to take non improving move moves according to probabilities tested with computer generated random numbers computer generated random numbers.

So, what happens in TABU search is that you say fine you do one thing, you allow the program. So, in this case for an example let us say you are here, you are here at this point and the you say the TABU list is let us say 20 non improving moves 20 non improving moves.

So, what happens is it will probably move from here 1 2 3 like this and then at within. So, it will allow the system to take 20 non improving moves and within this if you find that there is one move that is improving like this then you will keep on climbing this climbing this climbing this until you reach this.

So, the idea of this TABU list is to escape local optimum escape local optimum same way a by using probabilities you are trying to you know escape out of the local optimum and move towards a different solution. So, that is the idea right. So, simulated annealing and TABU search the only difference is that here there is a list here there is a set of probabilities generated using random numbers right.

(Refer Slide Time: 15:44)



Genetic Algorithms

Evolutionary Meta heuristics — move beyond any sequence of single Solution to evolve a population of possibilities (options) improved over time by mechanisms that are similar to biological process.

Genetic Algo - evolve good heuristic optimum by operations of Combining members of an improving population of Solutions.

Combine a pair of parent Solutions to generate good child Solutions by breaking the parents at same point and combing them

And we come to the last set or the last aspect of it is which is of the meta heuristics is what we call as the genetic algorithm ok. So, mostly people call this as evolutionary meta heuristics ok. So, why is it called evolutionary meta heuristics?

Here is what we are doing is we move beyond any sequence of single solution. So, you move outside this single solution approach that we already mentioned to evolve a population. So, you are not move moving from one solution to another, but what you are trying to do is you are trying to evolve a population of possibilities or you are trying to find a you population of options or you can think in this possibilities are or options improved over time by mechanisms that are similar to biological process.

So, what you do here is, instead of moving from one solution to another you develop mechanisms that allows you to create a pool of solutions and from the pool of solution which is what you call as a population then you find the best one of the best possible option that you can get.

So, what you do is. So, genetic algorithms; so genetic algo they evolve good heuristic optimum by operations of combining members of an improving population of solutions. So, this can seems like a tough word, but I will try to make it easy. What you do is you use an operation of combining solutions.

So, let us take simple example what we said in the previous case there are four sequence positions for us IV III II and I. So, let us look at the least slack and this one. So, it is III II IV I. So, one solution is 3, 2, 4, 1 and the other solution is I II IV III 1, 2, 4, 3 ok.

So, let us say the we have three solutions. So, this is your solution 1 and this is your solution 2 for the time being assume that these are good solutions let us assume these are two good solutions.

So, one way of doing the solution is that, you can create a scenario of doing this. So, we want to take this job which is at the four sequence position let us take a cut here this guy is four we want to change this and what we do is we take. So, in this case I want to try the second sequence position or let us try to find a place where we are trying to do this.

So, we move four here. So, if I get a solution 1 4 4 3 then this is not feasible because what happens is you know this has gone away. So, one option we can do is, you can do it as 1 4 you move it here and then put 3 2 ok. So, this guy comes this way and this case I will have 1 2 2 3. So, instead that is an infeasible solution. So, you can do this 1 2 3 and 4 ok.

So, these two solutions were derived out of this guy these solutions ok. So, you can think about it is actually not the case because you require a little bit more of a complicated solution to do this is not really good example what I gave, but the idea is that you remove. So, what the idea is that, this something called combine a pair of parent solutions to generate good child solutions by breaking the parents at some point and combining them.

So, what you are actually doing is, you are cutting the parent in some places and then you combine them in a particular fashion. So, what we have done is we cut the solution out this sequence position out move them here and put the next one in that position so, that kind of a thing right.

So, assume that because if this is a good parent. So, we can say good parent. So, from both good parents you should be able to get good children that is the idea and so, I creating a scenario of like this for evolving which we mix the biological evolution is what we call as a genetic algorithm.

This actually were very simple crude explanation for you guys to understand because there is a b c explanation that is coming from Amandeep in a much better fashion. So, it would probably help you to understand that better. So, thank you very much for your patience listening and we will see you in the concluding lecture.

Thank you.