Operations Research, Spring 2022 (110-2)
Case Assignment 1

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

1 The story

As one of the most famous food manufacturing companies in Taiwan, the IEDO company produces and sell sausage, ham, jerky, and other products.[1] Due to the huge demand volume, whether its factory is operated efficiently has significant impact on the company's profitability.

Almost all the products made by IEDO must go through *the smoking station*, which consists of five machines (called "smokers" in many places) and opens at 7:30 AM. There are three types of processes that may happen in a smoking station: *boiling*, *baking*, and *smoking*. Different products have different sequences of processes. For example, to allow a typical set of bacon leave the smoking station, we need 100 minutes of baking, 25 minutes of smoking, 30 minutes of boiling, another 60 minutes of boiling, and 150 minutes of baking. Each of the other products has its own sequence of processes, and the amounts of time vary according to the production amount. All the processes must be completed in order, and a process may be started only after its previous process has been completed.

In each morning, you need to schedule several *jobs* to the five machines, where a job is to manufacture one product for one customer. Interestingly, while there are five machines in the smoking station, machine 1, which is the oldest and cheapest, can only do the boiling process. On the contrary, each of machines 2 to 5 are able to conduct all the three types of processes. To complete the jobs as soon as possible, you should not abandon machine 1. However, to utilize machine 1, a job must be split into two pieces so that the two pieces are completed by two different machines (one is machine 1 and the other is one of machines 2 to 5). Job splitting obviously complicates the scheduling task. Luckily, due to the complicated nature of food manufacturing, a job cannot be split arbitrarily. For each job, at most one *splitting timing* is predetermined, which specifies

[1] This case study is a simplification of a real industry project did by the instructor in the past.

exactly after which process may a job be split. Note that it is possible that some jobs cannot be split. If a job's splitting time is 0, the job cannot be split.

Let's make things clearer with the example provided in Table 1. In this example instance, which was something happened in a past day, there are twelve jobs to be done. These jobs may be categorized into several groups:

Job	Process Type						S.T.
	Process 1	Process 2	Process 3	Process 4	Process 5	Process 6	
1	Boiling	Baking	Smoking				2
2	Boiling	Baking	Boiling				1
3	Boiling	Baking	Boiling	Baking	Boiling		1
4	Smoking	Boiling	Baking	Boiling	Baking	Smoking	3
5	Boiling	Baking	Boiling				2
6	Boiling						0
7	Baking	Smoking					1
8	Baking	Smoking					0
9	Boiling	Baking	Boiling				2
10	Baking	Boiling	Baking	Boiling	Baking		2
11	Boiling	Baking	Boiling				2
12	Boiling						0

Table 1: Process types of a example instance (S.T. means splitting timing)

- Jobs 6 and 12 have only one process. Their splitting timings are thus naturally 0.
- Job 8 has two processes, first baking and then smoking. However, its special characteristic makes it impossible to be split. It therefore should be scheduled to only one machine (and obviously it cannot be scheduled on machine 1).
- Jobs 1, 4, 7, and 10 all have multiple jobs and may be split. For example, job 1 requires three processes, which are boiling, baking, and smoking. As its splitting timing is 2, job 1 may be split into two pieces, where the first piece contains the first two processes and the second piece contains the third process. Nevertheless, note that neither the first piece nor the second piece may be done by machine 1. Only machines 2 to 5 may be used to complete job 1. As another example, job 4 has six processes and may be split into two pieces, each having three processes. None of the two pieces may be done by machine 1.

- Jobs 2, 3, 5, 9, and 11 all have multiple jobs and may be split. Moreover, one of the two pieces contains only boiling and may be done by machine 1. For example, job 3 may be split into two pieces where the first piece contains only one process of boiling. As machine 1 can do boiling, the first piece of job 3, if job is split, may be done on machine 1. The second piece may then be started on one of machines 2 to 5 after machine 1 finishes the first piece.

To make a schedule, certainly more information is needed. Table 2 lists the processing time of each process of each job and due time of each job. For example, there are three processes in job 1, each requiring 2.7 hours, 1 hour, and 0.5 hour. If job 1 is scheduled on a single machine starting at 7:30 AM, it will take $2.7 + 1 + 0.5 = 4.2$ hours to complete job 1. Its completion time will be 11:42 AM, 4.2 hours after 7:30 AM. As its due time is 12:30 AM, job 1 meets the due time and is not tardy.

Job	Processing Time (in Hours)						D.T.
	Process 1	Process 2	Process 3	Process 4	Process 5	Process 6	
1	2.7	1	0.5				12:30
2	1.6	1.4	0.9				12:30
3	1	0.9	0.2	0.2	1.4		12:30
4	0.5	0.7	1	0.6	0.3	0.5	12:30
5	0.8	1	0.9				12:30
6	2.7						12:30
7	1.4	1.5					17:30
8	1.1	1.1					17:30
9	0.8	1	0.8				17:30
10	1	0.5	0.7	1.1	1.1		17:30
11	1.1	1.4	1.5				17:30
12	2.3						17:30

Table 2: More information of the example instance (D.T. means due time)

The Gantt chart of an example complete schedule for the above instance is depicted in Figure 1. In this schedule, job 2 is split so that its boiling process is on machine 1 and its baking and second boiling processes are on machine 2. Jobs 5, 9, and 11 are also split so that machine 1 may be utilized. All the jobs are completed around 18:00.

There are two objectives of your scheduling task. First, the *number of tardy jobs* should be minimized. According to Figure 2, jobs 1 and 10 are tardy, and thus the number

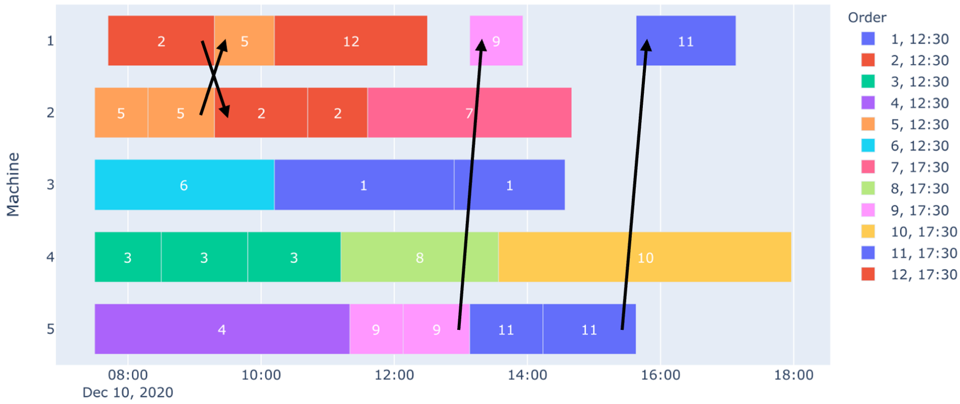

Figure 1: A (bad) schedule for the example instance

of tardy jobs is 2 in this schedule. Second, the *makespan*, which is latest completion time among all jobs, should be minimized. According to Figure 2 again, the makespan is around 18:00. A better schedule is depicted in Figure 3. In this better schedule, there is no tardy job, and the makespan is slightly before 17:30. The first objective has higher priority than the second one. In other words, when comparing two schedules, the one with the smaller number of tardy jobs is considered better regardless of their makespans. Their makespans are compared and the smaller one is preferred only when their numbers of tardy jobs are identical.

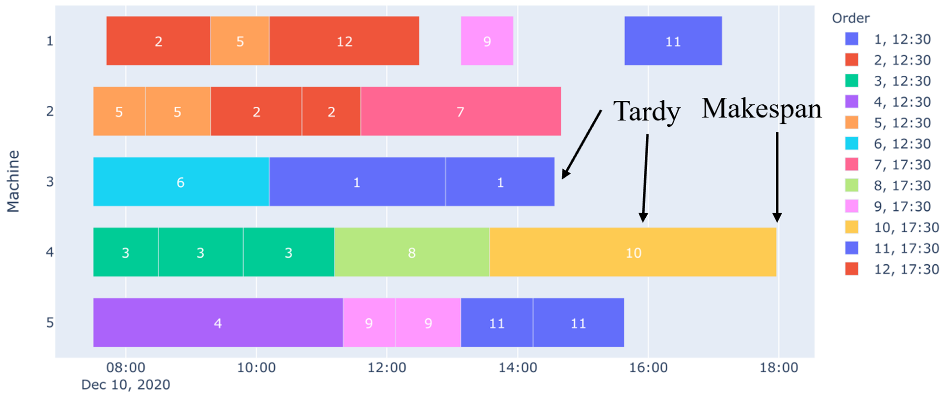

Figure 2: Illustration of the objective functions

Once a piece of a job is started on a machine, it cannot be stopped until the whole piece has been completed. If a job has two pieces, however, there may be a time gap between the completion time of the first piece and the starting time of the second piece even if the two pieces are on the same machine.

Given all the information above, please try to help the IEDO company solve the scheduling problem.

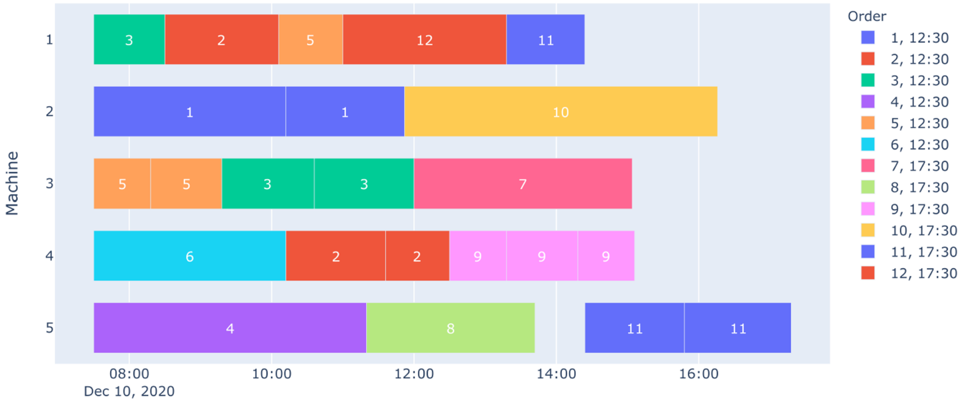

Figure 3: A (good) schedule for the example instance

2 Problems

For each of the following problems, please formulate a mathematical program (which ideally should be a linear integer program) that solves the problem, and then write a computer program to solve given instances in the MS Excel file “OR110-2_case01.xlsx”. You may find that MS Excel is not powerful enough to solve the given instances. In this case, watch videos in the second OR course made by the instructor on Coursera to learn how to write Python to invoke Gurobi Optimizer to solve the instances. It is fine if you prefer other optimization software/library as long as you may solve the instances.

In your report, write down your mathematical formulation using rigorous mathematical notations and expressions. Then use tables, figures, and business language to write down an optimal schedule obtained by your formulation. Here “business language” means those who have no knowledge in Operations Research should be able to understand and execute your schedule. Do not include your computer programs in your report.

1. Let’s start with a simple case by assuming that only the four machines (machines 2 to 5) that can do all the three types of processes are used. Moreover, let’s assume that all jobs cannot be split. To deal with two objective functions, you should solve an instance in two stages. In the first stage, ignore makespan and minimize the number of tardy jobs. After you find the minimum number of tardy job, say t, add a constraint to restrict the number of tardy job to be t and then minimize the makespan. Your second-stage optimal solution is what you want.

2. Now let’s assume that jobs may be split but still only machines 2 to 5 are used. Do you see your schedules become more efficient than those in Problem 1? Why or why not?

3. Finally, let's assume that jobs may be split and all five machines are used (and machine 1 can do only the boiling process). This is the real problem faced by the IEDO company. If you may solve it, you are quite close to using Operations Research to solve real-world business optimization problems!

By the way, in Case Assignment 2 the number of machines may not be five, and the number of machines with limited functions may not be one. Try to make your formulation as general as possible to save your time in doing Case Assignment 2!

3 Submission rules

- This case assignment is due at ***23:59, April 9***. Those who submit their works late but are late by less than one hour gets 10 points off. Works that are late more than one hour get no point.
- For this case assignment, students should work in teams. Each team should contain ***either four or five*** students. Teams that do not follow this rule gets no point.
- For each team, **one and exactly one** student should submit their work on behalf of all team members. Please submit a ***PDF file*** through NTU COOL and make sure that the submitted work contains the student IDs and names. Those who fail to do these will get 10 points off.
- You are required to ***type*** your work with LaTeX (strongly suggested) or a text processor with a formula editor. Hand-written works are not accepted. You are responsible to make your work professional in mathematical writing by following at least those specified in homework assignments. Those who fail to follow these rules may get at most 10 points off.
- Problems 1, 2, and 3 count for 30, 30, and 40 points, respectively.