TD 608
Project Management and Analysis

Part I
Project Conception and Execution

Milind Sohoni

Lecture 8

Operations Research Problems in Project Execution

Question 1 : Suppose that we have a list of tasks $\{T_1, \ldots, T_k\}$, where each task T_i has a start-time s_i, and end-time e_i and a JCB requirement r_i which is a positive integer. We must arrange for JCBs for each of the tasks.

Once a JCB is assigned to a task, it cannot be moved till the task is complete.

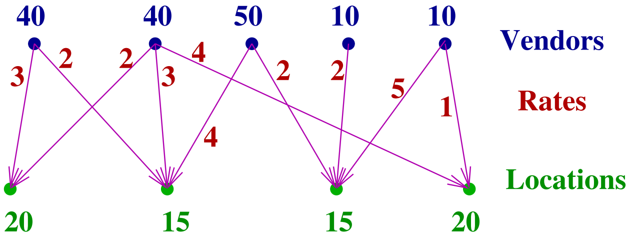
Tasks	T1	T2	T3	T4	T5	T6	T7
Start	1	2	4	5	8	10	15
End	5	7	10	8	10	12	18
Req.	1	2	2	2	2	3	2

- What is the total number of JCBs required for the project?
- What is a valid assignment of the JCBs to the tasks?

Operations Research Problems in Project Execution

Question 2 : Our project has locations $\{L_1, \ldots, L_k\}$ and each location L_i has demand d_i bags of cement per week. There are r vendors $\{V_1, \ldots, V_r\}$ of cement. Each vendor can supply no more than b_i bags per week. Furthermore, the cost of supply of a bag of cement from vendor V_i to location L_j is r_{ij}.

What is an optimal purchase order for each vendor and for each location.



Question 1 again

Question 1 : Suppose that we have a list of tasks $\{T_1, \ldots, T_k\}$, where each task T_i has a start-time s_i, and end-time e_i and a JCB requirement r_i which is a positive integer. We must arrange for JCBs for each of the tasks.

Once a JCB is assigned to a task, it cannot be moved till the task is complete.

Tasks	T1	T2	T3	T4	T5	T6	T7
Start	1	2	4	5	8	10	15
End	5	7	10	8	10	12	18
Req.	1	2	2	2	2	3	2

- What is the total number of JCBs required for the project?
- What is a valid assignment of the JCBs to the tasks?

An Observation

Lets look at the input carefully. Order the tasks in increasing order of start times.

Tasks	T1	T2	T3	T4	T5	T6	T7
Start	1	2	4	5	8	10	15
End	5	7	10	8	10	12	18
Req.	1	2	2	2	2	3	2

We make an activity chart:

Time	1.5	2.5	3.5	4.5	5.5	...	17.5
Tasks	T1	T1,T2	T1,T2	T1,T2,T3	T2,T3,T4	...	T7
JCBs	1	3	3	5	6	...	2

From here, we see that at $t = 5.5$ there must be 6 JCBs working. So clearly, a minimum of 6 JCBs are required.

The neat thing is that 6 are sufficient

The algorithm

Tasks	T1	T2	T3	T4	T5	T6	T7
Start	1	2	4	5	8	10	15
End	5	7	10	8	10	12	18
Req.	1	2	2	2	2	3	2

So let the JCBs be $J_1, \ldots, J_6$. Here are the basic steps:

- Prepare a combined list of start and end-times in sorted order. In case of conflict, keep the end-times before the start-times.

s1	s2	s3	e1	s4	e2	e4	s5	e3	e5	s6	e6	s7	27
1	2	4	5	5	7	8	8	10	10	10	12	15	18

- Start with the full collection $J_1, \ldots, J_6$ as the current set of available JCBs.
- For start-times, issue JCBs as per requirements from current set of available JCBs.
- At end-times receive JCBs already issued add to your current set of available JCBs.
- You will never run short!

A Typical Run

Tasks	T1	T2	T3	T4	T5	T6	T7
Start	1	2	4	5	8	10	15
End	5	7	10	8	10	12	18
Req.	1	2	2	2	2	3	2

	s1	s2	s3	e1	s4	e2	e4	s5	e3	e5	s6	e6	s7	e7
	1	2	4	5	5	7	8	8	10	10	10	12	15	18
	-1	-2	-2	+1	-2	+2	+2	-2	+2	+2	-3	+3	-2	+2
6	5	3	1	2	0	2	4	2	4	6	3	6	4	6

The Schedule

The schedule for each JCB is easily constructed:

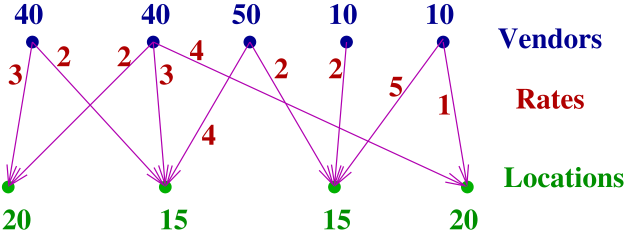
	s1	s2	s3	e1	s4	e2	e4	s5	e3	e5	s6	e6	s7	e7
	1	2	4	5	5	7	8	8	10	10	10	12	15	18
	-1	-2	-2	+1	-2	+2	+2	-2	+2	+2	-3	+3	-2	+2
6	5	3	1	2	0	2	4	2	4	6	3	6	4	6
J1	T1	T1	T1	*	T4	T4	*	T5	T5	*	T6	*	T7	*
J2	*	T2	T2	T2	T2	*	*	T5	T5	*	T6	*	T7	*
J3	*	T2	T2	T2	T2	*	*	*	*	*	T6	*	*	*
J4	*	*	T3	T3	T3	T3	T3	T3	*	*	*	*	*	*
J5	*	*	T3	T3	T3	T3	T3	T3	*	*	*	*	*	*
J6	*	*	*	*	T4	T4	*	*	*	*	*	*	*	*

- Note that JCB6 is used only for the period 5-8 and never used after that.
- If T6 is delayed by 2 units to 7-10, that will yield a saving of 1 JCB.
- Thus if the slack permits, this should be done.

Now to Question 2

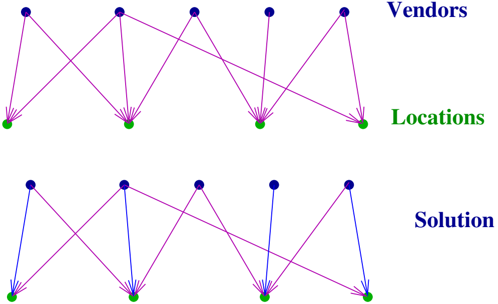
Question 2 : Our project has locations $\{L_1, \ldots, L_k\}$ and each location L_i has demand d_i bags of cement per week. There are r vendors $\{V_1, \ldots, V_r\}$ of cement. Each vendor can supply no more than b_i bags per week. Furthermore, the cost of supply of a bag of cement from vendor V_i to location L_j is r_{ij}.

What is an optimal purchase order for each vendor and for each location.



Simpler Question 2

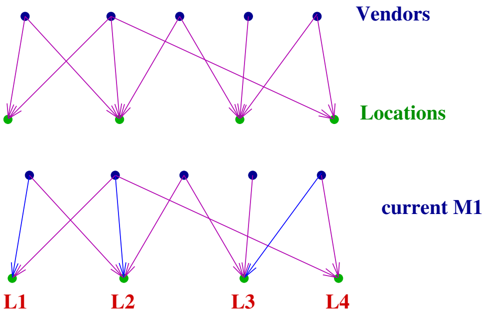
The Assignment Problem : Our project has locations $\{L_1, \ldots, L_k\}$ and each location L_i has demand 1 bag of cement per week. There are r vendors $\{V_1, \ldots, V_r\}$ of cement. Each vendor can supply exactly 1 bag per week. Furthermore, the cost of supply of a bag of cement from vendor V_i to location L_j is either 1 or ∞ (i.e., V_i cannot serve location L_j). Compute if the demand can be met at each location, and the vendor which will supply that location.



The Solution:Step I

Step I: Construct an initial allocation. This need not be optimal.

- Start with the locations $L_1, \ldots, L_k$ in any order.
- For every location L_i if an unused vendor V_{ij} can be found, the assign that vendor to location L_i.
- Stop after processing the location list.
- The matching so obtained is called your current matching M_1. This need not be optimal. Note L_4 is un-matched.

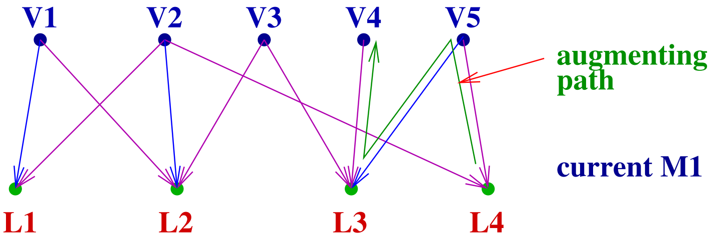


Step II

Augmenting path in M_1 :

- A path in the graph which starts from an unmatched location and goes to an unused vendor.
- It travels from location to vendor along an unmatched edge.
- It goes from vendor to location along a matched edge.

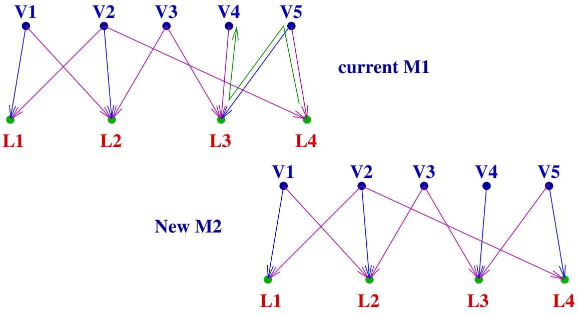
Step II: Look for an augmenting path



Step III

Step III: Update the matching to get M_2

- Make all unmatched edges in the augmenting path as matched.
- Make all matched edges as unmatched.
- This will produce a new matching M_2 which is of a larger size!



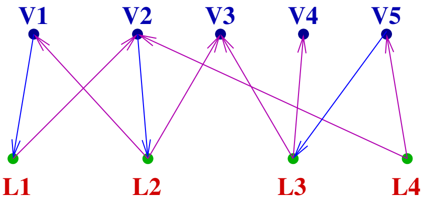
Finally...

Step IV: Apply Step II and Step III till no augmenting path is found.

Declare the matching so obtained as the Optimal Matching M

Moot Question : How is one to find an augmenting path?

- Reverse all unmatched edges.
- Start from every unserved location L_i , one at a time.
- See if you can reach an unused vendor by travelling in the graph.



Note that there are many augmenting paths:

- $L_4 \rightarrow V_5 \rightarrow L_3 \rightarrow V_4$
- $L_4 \rightarrow V_5 \rightarrow L_3 \rightarrow V_3$
- $L_4 \rightarrow V_2 \rightarrow L_2 \rightarrow V_3$