Job Shop Scheduling : N Tasks through M Machines

1. Background

A job shop consists of a set of distinct machines that process jobs. Each job is a series of tasks that require use of particular machines for known durations, and which must be completed in specified order. The job shop scheduling problem is to schedule the jobs on the machines to minimize the time necessary to process all jobs (i.e, the makespan) or some other metric of productivity. Job shop scheduling is one of the classic problems in Operations Research.

Data consists of duration table. Each task lists a job name, name of the required machine, and task duration. This formulation is quite general, but can also specify situations with no feasible solutions.

The minimization of Cmax in the general flow-shop case is an NP-hard problem in the strong sense (see Garey et al). Several heuristics have been proposed to solve it, and in particular the one of Campell Dudek and Smith (CDS) and the one of Nawaz, Enscore and Ham (NEH). Each of these heuristics gives good results, but none of them guarantees the optimal solution. Often, we will complement these algorithms with a 2-opt or a 3-opt.

2. Notation:

n: number of jobs.
m: number of machines.
pij: processing time of job i on machine j

Example: n = 5 Tasks through m = 4 Machines. Above the duration table (pij) :

Job / Machine	**M 1**	**M 2**	**M 3**	**M4**
J 1	5	4	7	2
J 2	7	8	3	4
J 3	6	3	4	9
J 4	4	6	7	6
J 5	9	1	2	5

3. Algorithm:

This heuristic simply consists in generating m-1 sub-problems of the 2-machine flow-shop type, solving them and selecting the best solution. The sub-problem k is defined by :
Processing time on the virtual machine 1 : pi1= the sum of pij (j in [1..k])
Processing time on the virtual machine 2 : pi2= the sum of pij (j in [k+1 .. m]).
For each of these problems, the optimal order is calculated with the Johnson algorithm and this order is then applied to the basic problem to obtain the Cmax(k). Then, it is enough to choose the best one on the whole Cmax(k).

4. Your Problem Details:

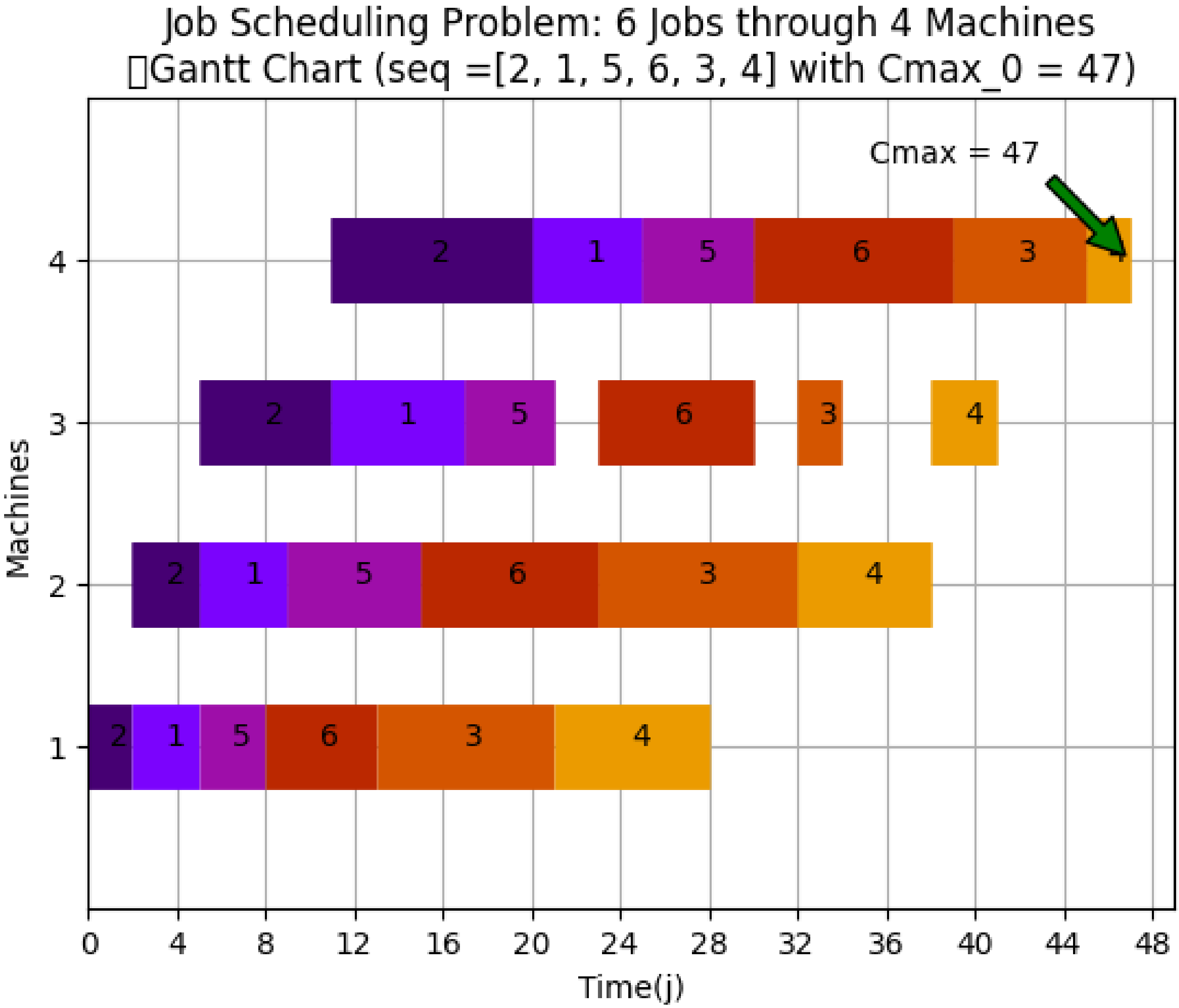
Your problem is a job scheduling problem with 6 jobs and 4 machines.
This table resume the tasks durations. Each task is a (job,machine) pair.

Job / Machine	M 1	M 2	M 3	M 4
J 1	3	4	6	5
J 2	2	3	6	9
J 3	8	9	2	6
J 4	7	6	3	2
J 5	3	6	4	5
J 6	5	8	7	9

5. Visualizing Results with Gantt Charts:

1) The result of subproblem number 1 solved by Johnson's algorithm is :

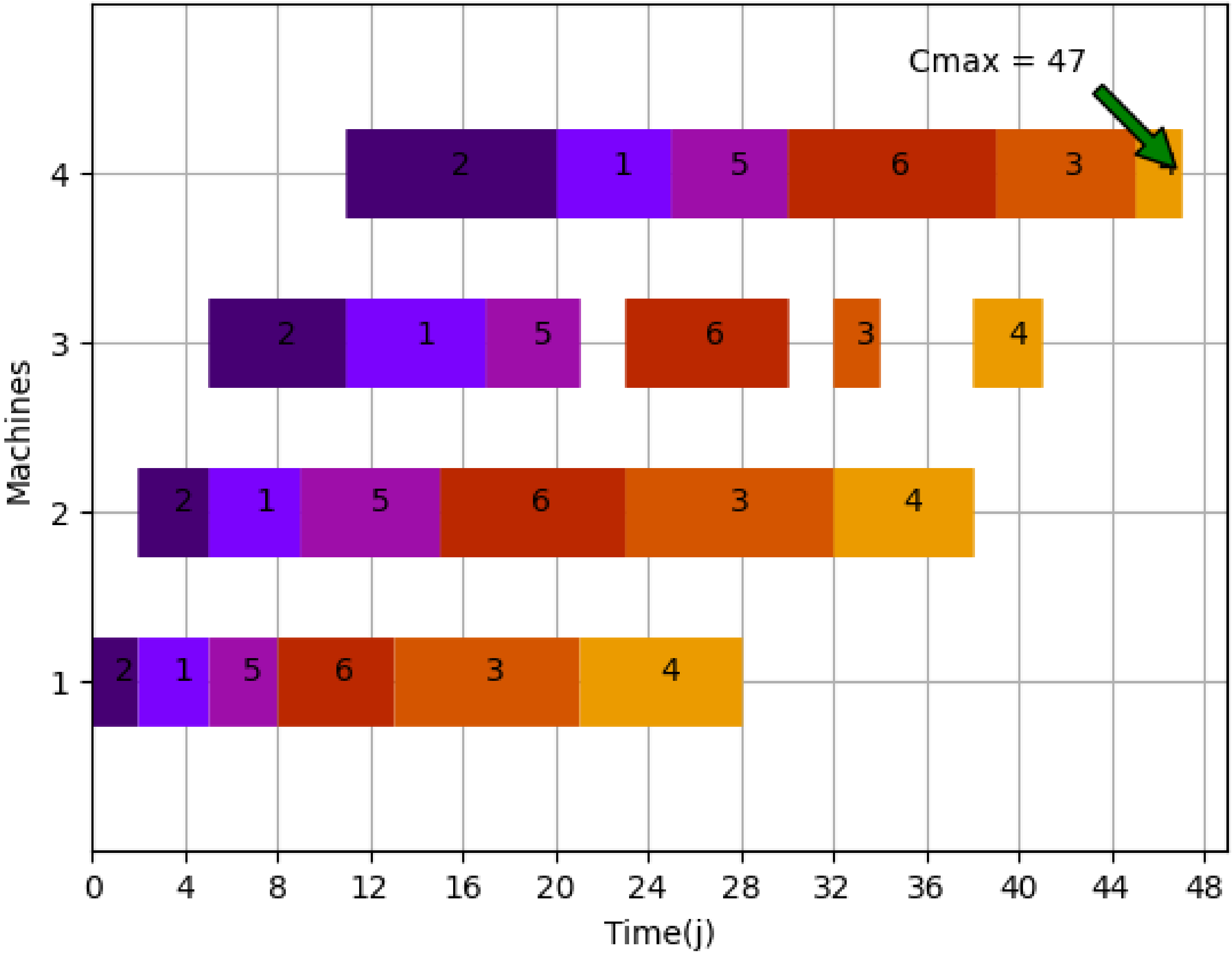
The optimal scheduling is therefore: 2=>1=>5=>6=>3=>4with cmax value of the real problem in this case is 47



5. Final Scheduling Result:

To conclude, among the results of the subproblems, we retain the following optimal sequence solution :|| 2=>1=>5=>6=>3=>4 || with a value of Cmax = 47

Job Scheduling Problem: 6 Jobs through 4 Machines
□Gantt Chart (seq =[2, 1, 5, 6, 3, 4] with Cmax_0 = 47)


Cmax = 47
Machines
Time(j)