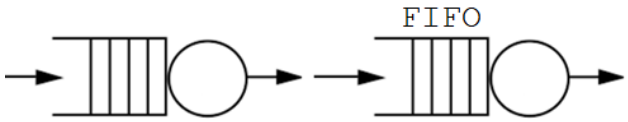
Ex. 1: Scheduling with two consecutive machines (Flow Shop)

Consider a scheduling problem with two machines. Each job must be processed two times: the first on the first machine (M1) and then on the second machine (M2).

The time for the job passage from the first to the second machine can be considered 0. Also, the time needed for a machine to unload/load a new job can be ignored.

A job can begin its processing on M1 after a release time.

In between the two machines there is a buffer, which is handled with a FIFO (First-In-First-Out) policy. The sequence of jobs that we send on M1 will be repeated on M2.



1. Model the problem with LP for minimizing the average flow time (equivalent to the sum of completion times of all the jobs).
2. If we change M2 buffer to a random-access one (in order to be free to send a different sequence on M2), will the system perform better?
3. Implement a greedy heuristic for the problem with FIFO buffer on M2 (a simple but efficient choice is the "Shortest Remaining Processing Time" strategy).

	J1	J2	J3	J4	J5	J6
Proc M1	72	53	97	16	24	19
Proc M2	43	33	23	55	38	43
Rel. T.	14	190	120	169	96	37

Ex. 1: Comments

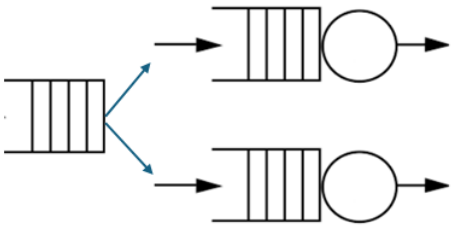
- Script Scheduling2Machines.m already contains the data.
- FIFO buffer: constraint the disjunctive constraints of M1 and M2 to use a shared binary variable (sequence will be the same on the two machines).
- Random access: define different binary variables for the disjunctive constraints of M1 and M2.
- The following pseudo-code implements the SRPT strategy on M1 (the same logic should be repeated for M2):

```
finished = 0
currTime = 0

while (not finished)
        search for a job i unscheduled on Mi1 with minimal p(1,i)+p(2,i) such that r(i)<=currTime
        if such a job exist
                set C(1,i)=currTime+P(1,i) and update currTime = C(1,i)
        else
                search for a job i, unreleased at time currT and with minimal r(i)
                if such a job does not exist, set finished = 1
                else set currTime = r(i)
```

Ex. 2: Parallel lines production system

Consider now two parallel machines. Your decision is now both to assign a job to the first or the second line, and the sequence of jobs on each line. The release times and processing times of the jobs are the same as in the previous exercise.
Implement the LP model to optimize the average flow time.



Ex. 2: Comments

An efficient formulation uses these reduced number of variables:

C(j) for j = 1,n (completion time of the job)
x(i) = {0,1} = job j is assigned to the first or second line
y(i,j) = for disjunctive constraints, for each couple of jobs i-j (one job sequence only).

Model features are:

- Objective function as the sum of all the completion times (equivalent to the average flow time)
- A job can start processing on the machine it is assigned only after its release time.
- All couple of jobs assigned to the same machine needs a disjunctive constraint.