Università degli Studi di Milano-Bicocca
Dipartimento Informatica Sistemistica e Comunicazione

Assignment 4. The Flow Shop Scheduling Problem

Exercise 1. The objective of the assignment is to apply different resolution methods to a non linear combinatorial problem. The problem that we will deal with is the Flow shop scheduling problem in which we have n machines and m jobs. Each job must be processed in all the machines following the same order, that is the i-th operation of the job must be executed on the i-th machine.

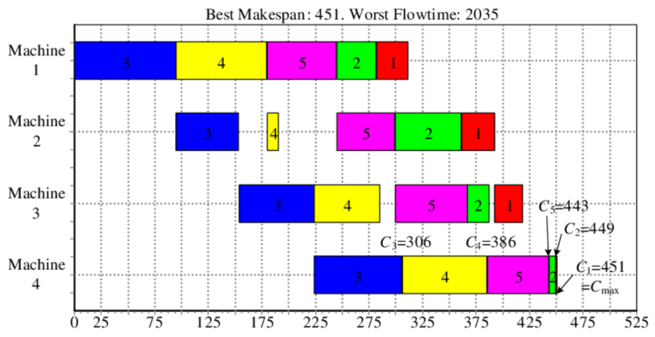
Moreover, we assume that machines can not work on more than one process simultaneously and that for each job the operation time in each machine is different and known at the beginning. You can assume assume that the order in which jobs are processed is exactly the same for all the machines.

The objective of the problem is to find an ordering of the jobs that minimizes total job execution time (also called *makespan*) that corresponds to the time on which all jobs get processed.

For example, let consider the following instance with five jobs and four machines:

	job(j)				
machine(i)	1	2	3	4	5
1	29	37	95	85	65
2	30	62	59	11	55
3	27	21	70	62	67
4	2	6	82	80	57

For these values, the optimal solution is to process jobs in the ordering 3,4,5,2,1 with a makespan of 451 as it is shown in the following figure:



In the webpage `http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html`(Flow shop sequencing) you will find a set of instances for 20 machines and jobs ranging from 20 jobs and 5 machines to 500 jobs and 20 machines. You will find also, for each instance the best known solution value.

Your task is to implement one or different local search techniques in order to find a sub-optimal solution on the biggest possible set of instances and compare your results against

- the best known solution values
- the results obtained using an algorithm available in R/Python.

Notice that we look for a permutation of the jobs, therefore you can use a large spectrum of techniques like ILS, Simulated Annealing, Genetic Algorithms, Tabu Search, Iterated Greedy...

P.S.: An extra point will be given if, in addition to meeting the above requirements, an algorithm not presented during the course is implemented and tested as well.