## Lexicographic optimization of a bi-objective flow shop with timelags

E.Dhouib(\*); D.Tuyttens(\*\*); J.Teghem(\*\*); T.Loukil(\*)
(\*) Laboratory LOGIQ, University of Sfax (Tunisia).
emna\_dhouib@yahoo.fr; taicir.loukil@fsegs.rnu.tn

(\*\*) Laboratory MathRO, University of Mons (Belgium) Daniel.Tuyttens@umons.ac.be; Jacques.Teghem@umons.ac.be

## ABSTRACT

We consider a permutation flow shop with time lags constraints. These constraints consist to impose some restrictions on the time elapsed between two successive operations of a same job. In the considered model there exist minimal and maximal time-lags which are respectively lower and upper bounds of the delay between the end of the operation on one machine and the beginning of the operation on the following machine. At each job is associated a due date. Two criteria are taken into consideration in a lexicographical manner: the primaray objective is the number of tardy jobs; the secondary one is the makespan. So the aim is to find the schedule within the minimal total completion time among the schedules with the minimal number of tardy jobs.

In a first step a mixed integer programming formulation is proposed and solved with CPLEX. It quickly appears that only small instances can be solved in a time limit of 5.000 seconds (experiments are made with 20 jobs and 3 or 5 machines). In a second step two local search metaheuristics are adapted to tackle the problem.

The first one is a Simulated Annealing (SA) based algorithm with some additional elements in regard with the classical SA scheme:

- Several solutions are generated randomly in the neighborhood and the best one is selected.
- Two acceptance tests of this solution are analyzed in regard of the lexicographical optimization.
- Several variants of a greedy heuristic are proposed to optimize the partial schedule of the tardy jobs, each time the first objective is improved.
- At the end of the SA procedure, an improvement scheme is applied to still trying to improve the final schedule.

The second metaheuristic is a GRASP algorithm. An iteration is composed of two phases:

- A greedy randomized search procedure is proposed to build a schedule taking into account the lexicographic optimization. At each step a list of possible jobs to schedule is based on the due date of the job and the generated total idle time of the machines.
- A local search to improve the schedule.

Extensive numerical experiments are made to compare both methods on instance with 60 or 80 jobs and 8 machines, and with 100 jobs and 5 machines. The aim of these experiments is not only to compare the performances of the two metaheuristics but also to analyze the impact of the respective values of the due dates and the time lags.