Upper bounds for minimizing the total completion time in a two-machine Flow shop under release dates and blocking constraint

Ines Agrebi[1] and Talel Ladhari[2]

[1]ISG TUNIS, noussa315@yahoo.fr

[2] ESSECT TUNIS, talel_ladhari2004@yahoo.fr

Keywords: Scheduling, Flowshop, Upper bound

Flow shops are useful tools in modeling manufacturing processes. A flow shop is a job processing facility which consists of several machines and several jobs to be processed on the machines. In a flow shop all jobs follow the same machine or processing order. Flow shop refers to the fact that job processing is not interrupted once started. Our objective is to find a sequence for the jobs so that the completion time is minimum. It is well known that this is a difficult problem to solve in a reasonable amount of time. In fact, the flow shop scheduling can be described as follows: There are n different jobs which have to be processed on m different machines. All jobs are processed on all machines in the same order. The processing times of the jobs on machines are fixed respectively to the order in which the processing is done. So, the problem is characterized by a matrix P = (pij), pij> 0, i=1,….,n and j=1,….,m of processing times. Each machine process exactly one job at a time and each job is processed on exactly one machine at a time. The problem then is to find a sequence of jobs such that the total completion time of the last job in the sequence on the last machine is minimum.

Problem Description:

The flowshop scheduling problem, studied in this research work, can be described as follows; We have, J = {J1, J2,…Jn} a set of n jobs which must be processed on two machines {M1, M2}.

We denote by:

- pij :Processing time.

- All jobs have to be processed on all machines in the same order; each of the n jobs must be processed during p1j time units without preemption firstly on machine M1, and then during p2j time units on machine M2.

- rj: Job j entering the system at its release date 'rj', so the processing of job j cannot be started before its release date rj.

- Each machine is available at time zero and can only perform one operation at a time; each job can have only one operation in progress simultaneously. The order by which the jobs are processed on any machine is identical and is not known.
- The Objective function consists in determining best scheduling in order to reduce the total completion time, the time where all operations are completed (∑ Cj).
Generally, in flowshop scheduling problem with two machines, a set of n jobs, J = {J1, J2… Jn}, must be executed on a set of 2 machines, M = {M1, M2}. Every job Ji requires an operation order, Oi = {Oi1, Oi2,…, Oin}, that must be executed according to his

manufacture process. Operation Oij needs an execution time Pij on machine Mij ∈ M and a release date rj on which the job J enter the system. Every machine can only execute one job at any time. Blocking constraint is authorized.

According to the three-field notation Graham et al. (1979) and Pinedo (2012), this strongly NP-hard problem is denoted by F2 | rj Blocking | ∑Cj: A problem of a two-machine flow shop with job j entering the system at its release date rj subject to minimizing the total completion time, blocking constraint must be taken into account in the fact that the machine remains blocked until this job starts on the next machine in the routing.
Proposed Solution:

We proposed a meta-heuristic based on local search to solve the treated problem which is: Genetic Local Search (GLS) (Holland 1975) : This approach is based on the mechanism of natural selection and genetic operators and used to find near optimal solutions for complex combinatorial optimization problems such as scheduling ones. The main idea of this method is that it considers a population of feasible solution and works with populations permits us to identify and explore properties which good solutions have in common. It consists of three simple operators which are reproduction, crossover and mutation.
The following algorithm shows our proposed Genetic Local Search (GLS):

Computational results:

The problem under consideration has never been studied in the literature, it is the first time that we tried to solve it. In fact, to give near-optimal solutions for large size instances in reasonable time. We proposed the local search heuristic (GLS).

So, in order to assess the quality of the proposed meta-heuristic, an evaluation is conducted on two classes A (A1, A2, A3) and B (B1, B2, B3, B4) of generated instances. The number of jobs n is taken equal to 10; 20; 30; 50; 100; 200. For each size n, 30 instances are generated, we use the schemes provided by Hariri and Potts (1983) and Haouari and Ladhari (2000) and used by Ladhari and Rakrouki (2009) and Rakrouki et al. (2012). To better investigate the performance of this method, we used three measures the average relative gap denoted by Gap, the average relative percentage deviation denoted by ARPD and the computational time (t) required to solve the given instance.
Extensive experiments are carried out on a large set of randomly generated instances reveal that our proposed local search based heuristic presents variable performances.