Investigation for solving hybrid flow shop problem with consumable resource

Imane Laribi[1], Fayçal Belkaid[1], Farouk Yalaoui[2], Zaki Sari[1]

1.Manufacturing Engineering Laboratory of Tlemcen, University of Tlemcen
B. P. N°119, Tlemcen 13000, Algérie
im.laribi@gmail.com, f_belkaid@yahoo.fr, z_sari@mail.univ-tlemcen.dz
2.ICD -LOSI (UMR-STMR CNRS 6281), Université de Technologie de Troyes 12 rue Marie Curie,
cs*42060 , 10004 Troyes Cedex, France*
farouk.yalaoui@utt.fr

Keywords: Scheduling, Hybrid flow shop, Heuristics, consumable resource, makespan

1 State of art

Hybrid flow shop scheduling problems combine the properties of flow shop and parallel machines scheduling problems. In this case, jobs visit the stages of the shop in the same order. The current trend in the development of hybrid flow shop was that of increasing complexity and relevance models. Additional constraints were introduced (e.g. Dugardin et al. [1] with the re-entrant process, Trabelsi et al. [2] with blocking constraint and some other papers with no-wait constraint, resource constraint,...etc) in order to extend the standard hybrid flow shop to a more complex, convenient and practical problem.

Scheduling problems with non- renewable resource constraints have been widely investigated in the literature, particularly for one stage system e.g. Belkaid et al. [3] considered the problem of consumable resources in parallel machine environment.

In hybrid flow shop literature, many papers do not address the influence of system with non-renewable resource, however, such a problem may arise in a manufacturing environment in which for example each product must initially go through a testing operation and then will be processed on any of parallel machines beside consumption of additional no-renewable resource so in this case lack of resource can disrupts the efficiency of scheduling. This provides us with the motivation to consider the proposed problem, which extends the works reported above.

The present paper further extends hybrid flow shop scheduling research by considering the problem with consumable resource, the objective is the minimization of the makespan with identical parallel machine at second stage and one common testing machine at the first stage, several heuristics based on priority dispatching rules are compared and evaluated on small, medium and large instances.

2 Problem description

There are N jobs to be processed in two stages in the same technological sequence, first in stage 1 then in stage 2. Stage 1 has one machine, but in stage 2 there are m parallel identical machines. Jobs for their processing in stage 2, besides machines, require additional resources. *nbRes* denotes the number of consumable resources. Each resource has a limited quantity R_{rt} *(r=1...nbRes)* denotes the quantity available of resource r at time t. Each job, during its processing in stage 2, require C_{ir} *(i=1…N)* units of additional consumable resources, the arrival of each resource is represented by a curve-shaped staircase. The processing time of job i $(i = 1,..., N)$ is equal to P_{i1} if it is executed on stage 1 machine, and the processing time of job i in stage 2 is equal to P_{i2}. Jobs can start on the second stage as soon as it has completed its processing time on the first stage and if all necessary resources are available at this time. Pre-emption is not allowed and machines can process only one job at a time. The makespan were used as an objective for our problem.

3 Solving approach

There are several exact algorithms to solve this problem but they are limited to solve only small instances in an acceptable time, for the medium and large instances the heuristics and metaheuristics are the most used as an alternative for this problem, however this heuristics cannot guarantee an exact solution for the problem. Several priority dispatching rules de pending not only on processing times but also on resource requirements of jobs were proposed to order jobs on stage 1. In this section we implement some heuristics to solve this problem, they are summarized as follows:

(1) H1, H2: Order the jobs in the first stage using the LPT rule and then FAM (First Available Machine) rule is used to determinate the machine assignment in the second stage. H2 is the same like H1 but we use SPT rule for order jobs in the first stage.

(2) H3, H4: a sequence of N jobs is generated in no increasing order of the summation of resource required for each job noted $\sum_{r=1}^{nbRes} Cir$ (i.e., LRC rule) and then sequence the stage 2 jobs using FAM rule. In the case of the heuristics H4 SRC rule was used in order to sequence jobs in stage 1.

(3) H5, H6: the heuristic H5 first sequences jobs on the machine at stage 1 according to Longest processing time to resources consumption ratio $(Pi1 + Pi2) / \sum_{r=1}^{nbRes} Cir$ (i.e., L-PT/RC). Given the order of jobs at stage 1, FAM rule solves the minimum makespan problem at second stage. In contrast, H6 schedules jobs at the first stage according to the shortest processing time to resources consumption ratios (i.e., S-PT/RC)

(4) H7: Sequences of N job is generated with Johnson's algorithm at the first stage and then sequence the second stage jobs using FAM rule.

(5) H8: jobs at the first stage are randomly sequenced and for the second stage FAM rule is used.

4 Experiments

The heuristics was tested and compared with some problem instances with the number of jobs are 20, 40, 60, 80 and 100. The number of identical machines *m* at stage 2 was set at 2, 4 and 6, and *nbRes* is equal to 1, 2, 3 resource type. Resource requirements *Cir* were generated from U [1, 5] (U [a, b] denotes the discrete uniform distribution in the range of [a, b]), processing time were generated from U [1, 100] even for the first and second stage. In order to evaluate the performance of each of the heuristics, we use the relative error between the current solution and the best solution for each instance and CPU time. Through these experimentations we can see that heuristic H5 provides the best performance, especially for the large instances problem. For future work we can try to adapt a genetic algorithm and compare the result with heuristics already adapted.

References

[1]Dugardin. F, Yalaoui. F, Amodeo. L (2010). New multi-objective method to solve reentrant hybrid flow shop. European Journal of Operational Research 203 (1) 22–31

[2]Trabelsi, W., Sauvey, C., & Sauer, N (2012). Heuristics and metaheuristics for mixed blocking constraints flowshop scheduling problems. Computers & Operations Research 39 (11) 2520–2527.

[3] Belkaid, F. Sari, Z. and Souier, M (2013). A Genetic Algorithm for the Parallel Machine Scheduling Problem with Consumable Resources. International Journal of Applied Metaheuristic Computing (IJAMC) 4 (2) 17-30.