Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-31 (2016) 109–113

Assessing the Performance of Dispatching Policies for Hybrid Flowshop Manufacturing Systems

Jairo R. Montoya-Torres*, Elyn L. Solano-Charris, Andres Muñoz-Villamizar****

**School of Management, Universidad de los Andes, Calle 21 # 1-20 Bogota, D.C., Colombia (e-mail: jr.montoya@uniandes.edu.co)*

***Escuela Internacional de Ciencias Económicas y Administrativas, Universidad de La Sabana km 7 autopista norte de Bogota, D.C., Chia, Colombia (e-mail: {erlyn.solano, andres.munoz1}@unisabana.edu.co)*

Abstract: This paper considers the problem of scheduling a given set of jobs or tasks in a hybrid flowshop system. Since this problem is known in the literature to be NP-hard, most of academic works has been centered on the analysis of meta-heuristics and hybridized procedures. However, intelligent manufacturing systems in actual industrial practices mostly prefer the use of simple heuristic algorithms, such as dispatching rules. The objective of this paper is to experimentally test the performance of well-known dispatching rules such as FCFS (First Come First Serve), EDD (Earliest Due Date), SLACK, and SRMWK (Slack per Remaining Work). To the best of our knowledge, this is the first work that aimed at this assessment. Various objective functions are considered: makespan, total tardiness, and the number of tardy jobs. Computational experiments are carried out using datasets from the literature.

Keywords: scheduling; decision support; industrial applications; dispatching rules.

1. INTRODUCTION

In contrast with the classical era of manufacturing, characterized by the important role of humans, the modern era of manufacturing systems "is ubiquitously permeated by computer and robotic devices (and linked by network connections)" (Silva 2014). So, manufacturing in this new perspective is a complex and demanding process where new challenges are presented, such as the introduction of automated planning and scheduling processes. At the operational decision-making level, scheduling of jobs or tasks is an important component of Intelligent Manufacturing Systems (IMS). The concept of IMS advocates that all the production resources can be modeled as an association between a physical part (hardware) and an informational part (software) (Thomas and Trenteseaux 2013). Hence, the intelligence of the informational part is of high importance.

From the computational point of view, scheduling is one of the hard optimization problems found in real industrial contexts. According to Pinedo (2008), scheduling problems deal with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. Among the various types of scheduling problems, the flexible or hybrid flowshop scheduling problem (HFSP) is one of the most challenging. This is a NP-hard optimization problem, even for the case of a system with only two processing stages in which one stage contains two machines and the other stage contains a single machine (Gupta 1998). The HFSP generalizes the well-known flowshop scheduling problem by adding some flexibility at each processing station thanks to use of parallel machines, as shown in Figure 1. In addition, this can increase the overall capacities and avoid bottlenecks if some operations are too long (Khalouli et al. 2011). The HFSP is of great importance in real industrial practice (Ruiz and Vázquez-Rodríguez 2010): it can be found in both manufacturing and service systems, including electronics, paper, and textile industries, manufacturing of photographic films, internet service architectures, and container handling systems.

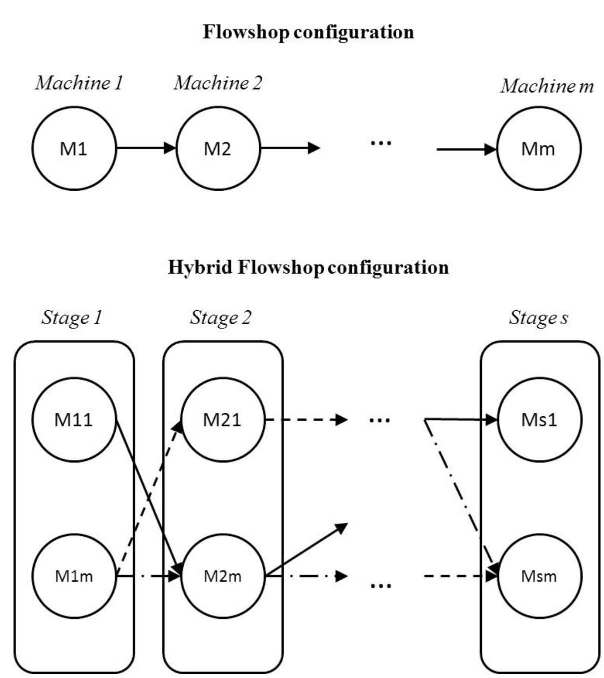

Fig. 1. Flowshop versus hybrid flowshop configurations.

Formally speaking, the problem is described as follows. A set of n jobs are to be processed on a set of s stages in series, each one containing a set of m_s machines in parallel. These

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.12.170

parallel machines can be identical (i.e., of equal capacity and processing speed) or not. For the case of this paper, we consider that all machines at each stage are identical. Each job j ($j = 1,\ldots, n$) has to be processed on only one machine at each stage. A machine can only execute one job at a given time. The processing routes of all jobs are identical (i.e., job j is first processed on one machine at stage 1, then at stage 2, and so on). The processing time of job j on any machine of stage s is denoted as p_{js}. Preemption is not allowed, which means that the execution of a job on a machine cannot be interrupted. Job j has to be finished before a given due date, denoted as d_j. Without lost of generality, processing times of jobs and due dates are supposed to be non-negative integers.

As in a large number of real-life optimization problems in economics and business, the NP-hardness of the hybrid flowshop scheduling problem means that large-sized problem instances cannot be solved in an exact (optimal) manner within a reasonable amount of time. Hence, the use of approximate algorithms is the main alternative to solve this class of problems. With the improvement of computing performance, the past 25 years have witnessed the development of numerous meta-heuristic algorithms in various communities that sit at the intersection of several fields, including Artificial Intelligence, Computational Intelligence, Soft Computing, Mathematical Programming, and Operational Research. Scheduling is one of the hard optimization problems found in real industrial contexts for which several meta-heuristic procedures have been successfully applied (Jourdan et al. 2009).

The scientific literature has extensively reported academic works and real-life applications for the single objective hybrid flowshop scheduling problem. Survey papers are presented by Ruiz and Vázquez-Rodríguez (2010), Chen (1994), Linn and Zhang (1999), Tyagi et al. (2013) and Yenisey and Yagmahan (2014). Lots of exact (optimal) procedures, heuristics and meta-heuristics have been proposed (see for example the works of Lee and Kim 2004, Lee et al. 2004, Choi et al. 2005, Paternina-Arboleda et al. 2008, Luo et al. 2011, Mirabi et al. 2013). Most of the meta-heuristics are: Genetic Algorithms, Simulated Annealing, Tabu Search and Ant Colony Optimization algorithms.

However, industrial practice mostly prefer the use of simple heuristic algorithms, such as dispatching policies, instead of using black-box decision-aid tools based on meta-heuristics or hybridized procedures (Kaban et al. 2012). The objective of this paper is to experimentally test the performance of well-known dispatching policies on the hybrid flowshop scheduling problem. To the best of our knowledge, this is the first work that aimed at this comparison. We will separately evaluate three different objective functions: the makespan or total duration of the schedule, which is related to maximize the productivity of the machines; the total tardiness; and the number of tardy jobs, associated with the customer service. The makespan is computed as $C_{\max} = \max C_j$, where C_j is the completion time of job j. The tardiness of the schedule is computed as $T_{max} = \max\{0, C_j - d_j\}$, where d_j is the due date of job j. Finally, the number of tardy jobs is computed as $\sum U_j$, where U_j is a binary variable with value equals to 1 if job j is tardy (that is, if $C_j > d_j$), and 0 otherwise.

It is important to note that the analysis proposed here is not a multi-objective approach. Instead, computational experiments are carried out using well-known datasets from the literature in order to evaluate the performance of various dispatching rules on various objective functions.

The rest of this paper is organized as follows. Section 2 presents in detail the dispatching policies implemented in the analysis, while Section 3 describes the computational experiments and the analysis of results. This paper ends in Section 4 by presenting some concluding remarks and suggesting lines for further research.

2. DESCRIPTION OF DISPATCHING POLICIES

Practitioners very often select simple heuristic procedures to find feasible solutions to complex decision-making problems. As explained previously, the aim of this work is to experimentally test the performance of well-known simple heuristic algorithms (i.e., dispatching rules), to solve the hybrid flowshop scheduling problem. These algorithms were selected based on the results of previous studies in the literature showing their good performance to flowshop-type scheduling problems (Pinedo 2008, Choi et al. 2015, Allaoui and Artiba 2004, Vallada et al. 2008). Selected algorithms are:

- FCFS (First Come First Serve): Jobs are assigned to machines as they become available in the same order as they arrive to the stage for processing. As jobs do not have release dates to the first stage, the first job is executed first in the sequence, the second job is executed second in the sequence, and so on.
- EDD (Earliest Due Date): For processing at each stage, jobs are ordered in a list by increasing order of their due dates. Once a machine becomes available, the first job in the list is assigned for processing.
- SLACK: At each stage, once a machine becomes available at time t, the job with the minimum value of $d_j - C_j(\pi)$ is selected, where $C_j(\pi)$ is the completion time of job j□π if it is scheduled at the end of current sequence π.
- SRMWK (Slack per Remaining Work): At each stage, once a machine becomes available at time t, the job with the minimum value of $\frac{[d_j - C_j(\pi)]}{\sum_{j=1}^{m_s} p_{js}}$ is selected for processing, where $C_j(\pi)$ is the completion time of job j□π if it is scheduled at the end of current sequence π; and p_{js} is the processing time of job j on stage s.

In addition, the previous dispatching rules are hybridized as follows:

- SLACK+FCFS: At the first stage jobs are scheduled following the SLACK rule, while at the second stage (and onwards for configurations with more than two stages), jobs are scheduled following the First-Come-First-Served rule.