Simulation Modelling Practice and Theory 79 (2017) 23–36

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times

Eliana Maria Gonzalez-Neira[a,c,*], Daniele Ferone[b], Sara Hatami[b], Angel A. Juan[b]

[a] Departmento de Ingeniería Industrial, Pontificia Universidad Javeriana, Bogotá, Colombia
[b] IN3 – Computer Science, Multimedia and Telecommunications Department, Open University of Catalonia, Barcelona, Spain
[c] Doctorado en Logística y Gestión de Cadenas de Suministros, Universidad de La Sabana, Chía, Colombia

ARTICLE INFO

Article history:
Received 4 May 2017
Revised 11 September 2017
Accepted 12 September 2017

Keywords:
Distributed assembly flowshop
Stochastic optimization
Simulation-optimization
Metaheuristics
Biased randomization

ABSTRACT

Modern manufacturing systems are composed of several stages. We consider a manufacturing environment in which different parts of a product are completed in a first stage by a set of distributed flowshop lines, and then assembled in a second stage. This is known as the distributed assembly permutation flowshop problem (DAPFSP). This paper studies the stochastic version of the DAPFSP, in which processing and assembly times are random variables. Besides minimizing the expected makespan, we also discuss the need for considering other measures of statistical dispersion in order to account for risk. A hybrid algorithm is proposed for solving this $\mathcal{NP}$-hard and stochastic problem. Our approach integrates biased randomization and simulation techniques inside a metaheuristic framework. A series of computational experiments contribute to illustrate the effectiveness of our approach.

1. Introduction

Scheduling optimization deals with allocating, in a specified period of time, limited resources to tasks in order to achieve one or more goals determined by the decision maker [85]. One of the scheduling optimization problems that has been extensively studied during the last decades is the flowshop problem (FSP) [83,94]. The FSP has applications in different industry sectors, among others: the metallurgical, chemical, textile, iron and steel making, etc. [65,97]. In a FSP, a set of m machines have to process a set of n jobs, which are available at time zero. Each job requires to pass through all the machines. The processing time of job j in machine i is denoted by p_{ij}. Typically, at any given time each machine is able to process just one job, and each job can only be processed by one machine. Notice that there are $n!$ possible job sequences per machine. Therefore the total number of possible job sequences adds up to $(n!)^m$. Usually, it is assumed that the job sequence of the first machine is kept for all the remaining machines. Thus, the number of feasible job sequences is reduced from $(n!)^m$ to $n!$. This FSP variant is known as the permutation flowshop problem (PFSP). The PFSP is one of the most studied problems in scheduling optimization [23,60,102].

In the scheduling literature, it has been usual to assume that processing times are deterministic and known in advance. However, in real life these times are frequently subject to certain degree of uncertainty, which can be due to different fac-

* Corresponding author.
E-mail addresses: eliana.gonzalez@javeriana.edu.co, elianagone@unisabana.edu.co, egonzalezn55@gmail.com (E.M. Gonzalez-Neira), dferone@uoc.edu (D. Ferone), shatami@uoc.edu (S. Hatami), ajuanp@uoc.edu (A.A. Juan).

http://dx.doi.org/10.1016/j.simpat.2017.09.001
1569-190X/

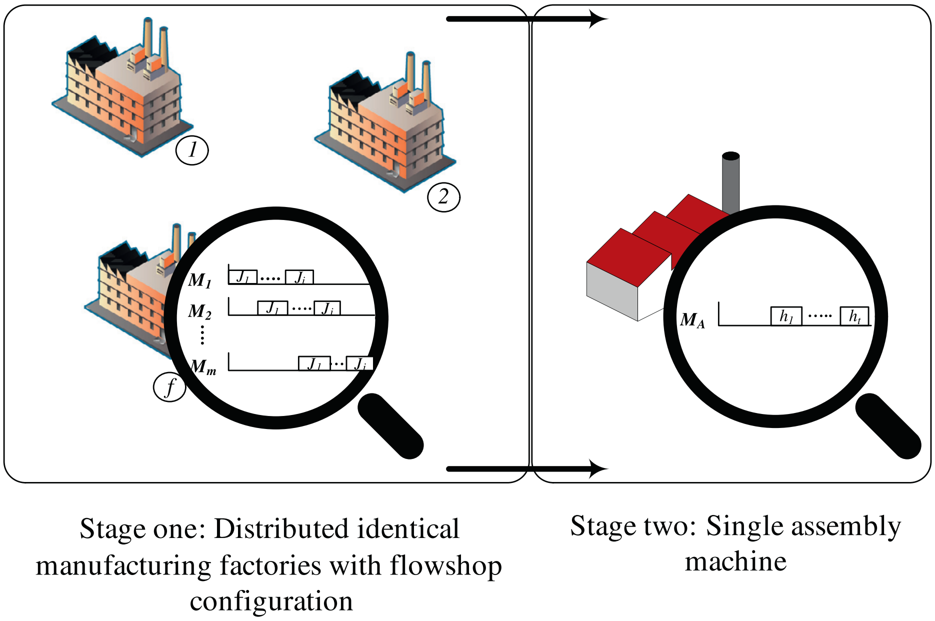

Fig. 1. A schematic representation of the DAPFSP.

tors, e.g.: machine breakdowns, machine unavailability, human participation, environmental conditions, etc. [90]. Therefore, instead of assuming deterministic processing times, this paper considers a more realistic version in which these times are random variables. While the PFSP has been intensively studied in the literature, the stochastic counterpart has received much less attention [26].

Besides assuming deterministic data, the PFSP literature has traditionally considered a single factory. Nowadays, however, many industries have more than one production center in which jobs are processed. These multi-factory systems are commonly denominated as distributed manufacturing systems (DMS). A DMS allows enterprises to attain better product quality while reducing management risk and total cost [73,95]. In a DMS, it is likely that the different parts (modules) provided by each factory require an assembly stage in order to generate the final product. This is known as the assembly scheduling problem (ASP). Shop configurations associated with assembly operations give companies the capability of producing customized products, as well as modular products, which contributes to reduce production costs [62]. Offering a high diversity of products might represent a competitive advantage in an dynamic and global market. This type of shop configuration has applications in different industries, e.g.: fire-engine assembly plants [64], distributed database systems [1], personal computer manufacturing [88], plastic manufacturing [2], etc.

The combination of a DMS with an assembly system was initially analyzed by Hatami et al. [45], who introduced the concept of distributed assembly permutation flowshop problem (DAPFSP). The DAPFSP is a generalization of the distributed permutation flowshop problem (DPFSP). In the DPFSP there are several identical factories, each one with a different flowshop configuration. Also, there is a set of jobs that have to be assigned to the factories and then processed at the appointed one. The DAPFSP can be split in two stages. The first one is a *production stage* composed of f distributed and identical factories. Each factory is modeled as a PFSP, with a set M of m machines and a set N of n jobs. The second one is an *assembly stage*. This includes a single assembly machine in which a set T of t final products are assembled. Each product $h \in T$ is composed of $|N_h|$ parts, which are merged through a given assembly program. In this paper, it will be assumed that each of these parts is the outcome of processing a given job in a factory –i.e., there is a one-to-one relationship between a part and a job–, and that different parts of the same product can be generated in different factories. The assembly operation of product h can begin only once all its $|N_h|$ parts have been completed at the production stage. The assembly time of product h at the assembly machine is denoted by pp_h. In this environment two decisions are taken: *(i)* to which factory a job $j \in N$ has to be assigned; and *(ii)*, in which sequential order should the assigned jobs be processed at each factory. A schematic diagram of the DAPFSP is shown in Fig. 1.

To the best of our knowledge, only two conference papers have addressed stochastic versions of the DAPFSP (SDAPFSP) with random processing and assembly times. In the first one, Ji et al. [49] considered also a no-wait constraint in the processing stage. These authors used the PSOSAHT algorithm [71], which combines particle swarm optimization with simulated annealing concepts. In the second work, Du et al. [22] considered stochastic sequence-dependent setup times as well as stochastic job release times. Again, the PSOSAHT algorithm was used to solve the problem. Both papers employed a hypothesis-testing approach to evaluate and compare the stochastic results. As a contribution to the emerging interest in the SDAPFSP, we propose a simheuristic algorithm [56] to minimize the expected makespan. Our approach integrates biased