Ann Oper Res
DOI 10.1007/s10479-017-2678-x

ORIGINAL PAPER

Dispatching algorithm for production programming of flexible job-shop systems in the smart factory industry

Miguel A. Ortíz[1] · **Leidy E. Betancourt**[1] ·
Kevin Parra Negrete[1] · **Fabio De Felice**[2] · **Antonella Petrillo**[3]

Abstract In today highly competitive and globalized markets, an efficient use of production resources is necessary for manufacturing enterprises. In this research, the problem of scheduling and sequencing of manufacturing system is presented. A flexible job shop problem sequencing problem is analyzed in detail. After formulating this problem mathematically, a new model is proposed. This problem is not only theoretically interesting, but also practically relevant. An illustrative example is also conducted to demonstrate the applicability of the proposed model.

Keywords Sequencing problem · Flexible job-shop system · Reconfigurable system · Optimization

---

✉ Antonella Petrillo
antonella.petrillo@uniparthenope.it

Miguel A. Ortíz
mortiz1@cuc.edu.co

Leidy E. Betancourt
lbetanco3@cuc.edu.co

Kevin Parra Negrete
kparra7@cuc.edu.co

Fabio De Felice
defelice@unicas.it

[1] Department of Industrial Engineering, Universidad de la Costa CUC, Barranquilla, Colombia

[2] Department of Civil and Mechanical Engineering, University of Cassino and Southern Lazio, 03043 Cassino, Italy

[3] Department of Engineering, University of Naples “Parthenope”, 80143 Naples, Italy

Journal: 10479 Article No.: 2678 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2017/10/6 Pages: 25 Layout: Small

Nomenclature

n	Number of pieces
m	Number of machines
G(i)	Number of operations of the piece i (i = 1…n)
H(j)	Number of operations in the tail of the machine j (j = 1 …m).
fj	Instant the machine j is available for a new operation
pk,i,j	Processing time of operation k of the item i in the machine j
rk,i	Instant availability of operation (k, i)
Rpk,i,j	(release date o ready date) Early instant to start the operation (k, i) on the machine j: rpk,i,j = max { rk,i, fj for all (k,i) ϵ Ej
fpj	Early instant to start a new operation on the machine j (if no queue, infinite value is assigned): fpj = min(k,i) ϵ Ej { rpk,i,j } if Ej is not empty fpj = ∞ if Ej is empty
t_{start} (k, i)	Manufacturing start time scheduled operation [t_{start}] (k, i) = rpk,i, j
t_{end} (k, i)	Manufacturing final instant programmed operation [t_{end}] (k, i) = tstart (k, i) + D · pk,i,j

1 Introduction

Manufacturing companies spend considerable efforts to improve their production processes and to optimize production scheduling in order to increase their production efficiency (Barrios et al. 2015; Herazo-Padilla et al. 2015). Therefore, there is a need for heuristics to find optimal solutions to these problems. One class of scheduling heuristics widely used in industry is *dispatching rules*. Dispatching rules as a special kind of priority rules are applied to assign a job to a machine (Hildebrandt et al. 2010). Specifically in this research, the Flexible Job Shop Problem (FJSP) is analyzed. FJSP, introduced by Brandimarte in 1993, is an extension of the classical job-shop scheduling problem, which allows an operation to be processed by any machine from a given set. The problem is assigning each operation to a machine and ordering the operations on the machines, so that the maximal completion time (makespan) of all operations is minimized. Before analyzing the specific problem of our research it is important to define what *"Scheduling"* means. It is a term used in our everyday vocabulary, although we may not always have a good definition of this term in mind. In the scheduling process, it is necessary to know the type and the amount of each *resource* in order to determine when the tasks can feasibly be accomplished. Many of the early developments in the field of scheduling were motivated by problems arising in manufacturing. Scheduling theory is concerned primarily with mathematical models that relate to the process of scheduling. The development of useful models, which leads in turn to solution techniques and practical insights, has been the continuing interface between theory and practice (Pinedo 2001). A solution to a scheduling problem is any feasible resolution that combines theory and practice, so that "solving" a scheduling problem amounts to answering two kinds of questions: *Which resources should be allocated to perform each task?* and *When should each task be performed?* In other words, a scheduling problem gives rise to allocation decisions and sequencing decisions. Traditionally, many scheduling problems have been viewed as problems of optimization. Today, Scheduling represents a body of knowledge about models, techniques, and insights related to actual systems (Baker 2005).

Most of the literature on scheduling theory, relates to regular measures as total flow time, number of tardy jobs, and total tardiness (Hassin and Shani 2005; Digiesi et al. 2013). The total tardiness criterion, in particular, has been a standard way of measuring conformance to due dates, although it ignores the consequences of completing jobs early and penalizes only those jobs that finish late. However, this emphasis began to change with the growing interest in just-in-time (JIT) production. As stated by Karimi-Nasab and Modarres (2015) a production manager should minimize the total cost of the plant by simultaneously deciding about lot sizes and process routing of some items over several planning periods. Decisions about the production schedule are traditionally made in a sequential manner, followed by a number of recursive corrective actions. In this context, mathematical programming approaches are useful in order to obtain an optimal solution. The most favorite area of research in the literature is dedicated to providing different formulations for the problem according to various working conditions that can arise in real case observations. Clearly, not all workshop machines are the same. In a production department there are many different types of workplace machines.

Such differences could cause variations in processing speeds of machines to perform jobs (Prot et al. 2013). Thus, it is necessary to develop a specific mathematical model for each specific scenario.

The aim of this research is to present a *dispatching algorithm* for programming production of flexible job-shop systems. After mathematically formulating this problem, a model is proposed in the textile sector. In this regard, the proposed algorithm is an adaptation of Calleja and Pastor's (2014) since it incorporates *quality restrictions* considering the technical skills required for performing certain operations in some complex products. Thus, the main contribution of our paper is on the inclusion of quality restrictions in the mathematical model. These constraints allow modeling the selection of worker-machines according to the product type. In this regard, the model is applied in a textile company when some unskilled resources cannot perform the stamped and jacquard towels, and a high defective percentage may appear. This has been adopted aiming to avoid non-quality overcosting and delivery delays. This is the main contribution to the state of art, considering that these restrictions have been poorly developed in the reported literature.

The remainder of this paper is organized as follows: Sect. 2 presents a brief analysis on literature review concerning the flexible job shop problem; Sect. 3 defines problem formulation and dispatching rules; Sect. 4 shows an illustrative example and experimental results. Finally in Sect. 5, conclusions are presented.

2 Literature review

Scheduling for the flexible job-shop is a very important topic in the fields of production management and combinatorial optimization (Fattahi and Fallahi 2010). In fact, as discussed by Sun et al. (2014) flexible job-shop scheduling problem subject to machine breakdowns is one of the challenging problems in manufacturing. Solving this problem means increasing production efficiency, reducing costs and improving product quality (De Felice and Petrillo 2013).

For the above reasons, recently, scheduling and the more general topic of scheduling under uncertainty have attracted the interest of many researchers (Guo 2006). The flexible job-shop problem is an extension of the classical job-shop scheduling problem, which allows

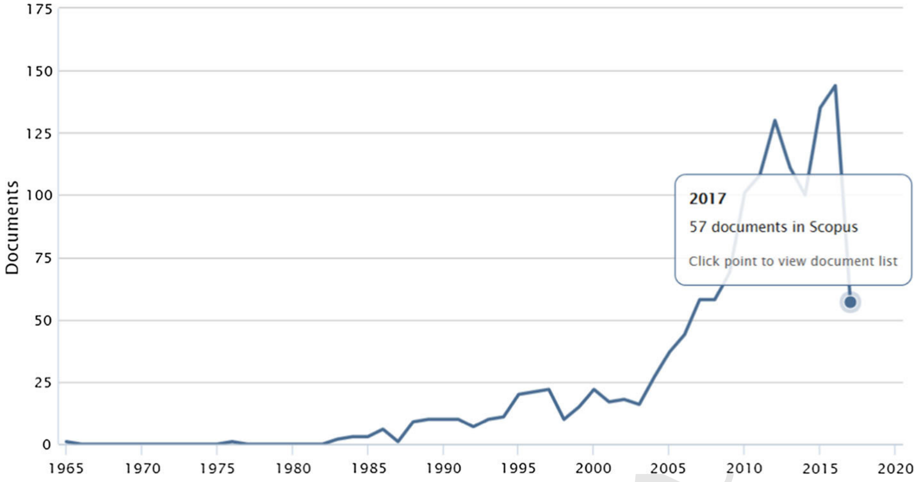

Fig. 1 Documents by year. (*Source*: Scopus)

an operation to be processed by any machine from a given set. The problem is to assign each operation to a machine and to order the operations on the machines, such that the maximal completion time (makespan) of all operations is minimized (Kacem et al. 2002; Hu 2015).

The difficulty of FJSP suggests the adoption of heuristic methods also for small manufacturing operations, producing reasonably good schedules in a reasonable time, instead of looking for an exact solution (Jansen et al. 2000). Heuristic algorithms are developed to solve the parallel-machine job-shop problems where the criterion is the minimization of the makespan by Sotskov and Gholami (2015).

Because of the difficulty of its resolution the flexible job-shop problem has been analyzed by several authors. Multiple approaches have been proposed to solve FJS problems in the literature. Some of them are based on genetic algorithms (Türkylmaz and Bulkan 2015; Demir and Işleyen 2014), others are based on hybrid methods combining FJS problem and a dedicated continuous material flow model (Bozek and Wysocki 2015) or as a combination of group constraint with flexible flow shop to minimize makespan (Kurz and Askin 2004; Logendran et al. 2005). In some other cases authors studied a hybrid flow shop scheduling problem with assembly operations at stage two (Fattahi et al. 2014; Yokoyama 2004; Riane et al. 2002; Wang and Liu 2013).

For a comprehensive review of the phenomenon related to our research, an investigation on Scopus database, the largest abstract and citation database of peer-reviewed literature, was carried out. Below, a summary of literature review concerning our research is analyzed.

Pertinent articles were searched using the string "*flexible job shop problem*", according to the standards of Scopus database. Three main criteria were adopted to select relevant articles. Articles were considered suitable for this review if the string "*flexible job shop problem*" was found in (1) article title, or in (2) abstract or in (3) keywords. The analysis on Scopus pointed out that from 1965, when the first article was published on Scopus, until July 2017 (the research period) a set of 1424 documents were published, of which 825 were articles, 514 conference papers and the remaining were books, editorials, letters, etc. The literature search highlighted an increasing number of publications, in fact, 57 documents have already been published in 2017 (Fig. 1).

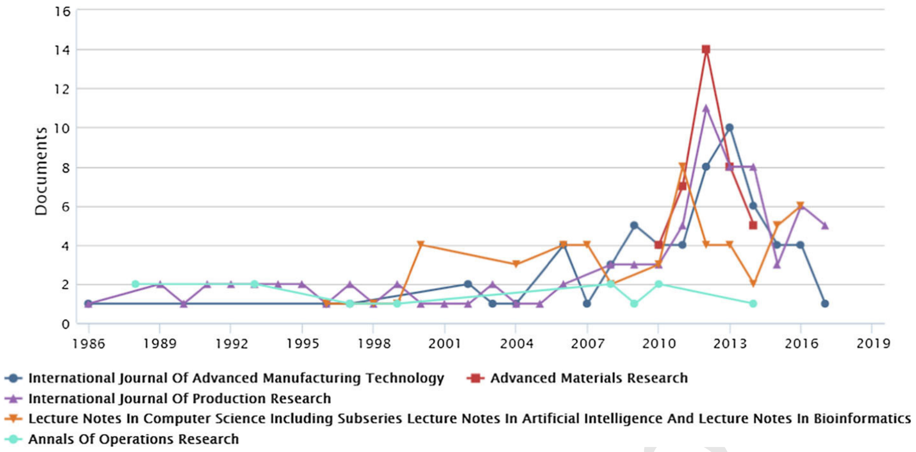

Fig. 2 Documents per year by source. (*Source*: Scopus)

Most of them have been published on the International Journal of Advanced Manufacturing Technology, 84 articles from 1965 to 2017, as shown in Fig. 2.

Most of them were published between 2012 and 2017 (a total of 677 documents). Furthermore, most of the publications (i.e., 544) have been published in China.

Considering our specific field of interest, the search was refined applying an additional filter. The search string used was *“flexible job shop problem AND dispatching algorithm*”. Relevant articles were selected according to the three criteria described above. 62 articles of the 1424, were identified, from 1983 (when the first article was published on Scopus) to July 2017 (the research period). Then, the field of inquiry was limited only to the third criterion “keywords”. In this case, only 14 documents were selected.

Considering the 14 selected documents, the analysis showed that most of them were published in China (i.e., 4 documents) and in Singapore (i.e., 3 documents) followed by Iran (i.e., 2 documents) and Spain (i.e., 2 documents), etc.

Among the identified documents, Jungwattanakit et al. (2009), considered a flexible flow shop scheduling problem, where at least one production stage is made up of unrelated parallel machines. They aimed to minimize a convex sum of makespan and the number of tardy jobs.

While in 2015, Shen and Yao proposed an interesting model, developing a new mathematical model for the multi-objective dynamic flexible job-shop scheduling problem (MODFJSSP); in 2014 Gholami and Sotskov presented an adaptive algorithm with a learning stage for solving the parallel machines job-shop problem modeling a job-shop problem via a weighted mixed graph. Still in 2014, Calleja and Pastor proposed a relevant study concerning a dispatching algorithm with transfer batches. They aimed to minimize the average tardiness of production orders considering two variants: (1) an ordered variant and (2) a randomized variant. In this regard, our proposed algorithm is an adaptation of Calleja and Pastor’s since it incorporates *quality restrictions* considering the technical skills required for performing certain operations in some complex products. Despite the amount of research on the flexible job-shop problem, it is quite difficult to achieve an optimal solution to this problem with traditional optimization approaches because the flexible job-shop scheduling problem allows an operation to be processed by any machine from a given set.

We can conclude that scheduling operations is one of the most critical issues in the planning and managing of manufacturing processes, and one of the most difficult problems in this area

Fig. 3 Dispatching algorithm scheme. *Source*: Calleja and Pastor (2014)

is the Job-shop Scheduling Problem. To find the best schedule can be very easy or very difficult, depending on the shop environment, the process constraints, and the performance indicator (Pezzella et al. 2008). Thus, the aim of this research is to cover this gap.

3 Problem formulation and dispatching rules

This section deals with a model based on a *flexible job-shop system* where "n" parts (orders) have to be produced in "m" machines (resources, operators or work centers). The execution of each type of item requires performing a series of operations whose sequence could be different depending on the production route of the part. Specifically, each operation is assigned to one of the "m" machines and has a particular and known processing time. The problem consists of setting a program for each machine with the purpose of optimizing a key indicator index that measures the program efficiency. To solve the problem of sequencing of the company a heuristic dispatching algorithm is proposed. Figure 3 shows dispatching algorithm scheme.

This is a constructive heuristic (it begins with an empty initial solution and elements are added according to certain criteria in order to obtain the final solution) and direct (once an operation is programmed, it is not reconsidered or modified in following steps).

At a given moment of the process, the set E (eligible operations) consists of operations with its precedents in the subset P (operations already scheduled).

When programming a feature of a part, the operation automatically proceeds from E to P and the following operation of the piece is moved from N (set of unscheduled operations) to E (unless the transaction is the last piece).

Initially, an n/m problem is located at E, the first "n" parts operations with unprecedented operation, and the remaining operations in N; P is empty.

At time "t", the subsets N, E and P are in a specific state. When the algorithm finishes, all operations are in P.

The set E is sub-divided into subsets that are characterized by operations that are to be processed in the machine or machines. Thus, Ej ($E = E1 \cap E2 \cap \ldots \cap Em$) is the subset of E to be performed on the machine j, also called operation queue on the machine j (which may be empty during application).

In each iteration, the programming is done in two phases. First, choosing the machine or machines with the least available capacity to start a new operation, then programming an operation among operations of subset Ej according to the established rules of priority.

In the following paragraphs, a description of rules for machine and operation selection is detailed.

3.1 Machine selection

The machine will be chosen according to the following priority rules:

- Rule 1: Select the machine j that is available sooner. This means, the machine with the least fpj: fpmin = min {fpj}. If there is a tie between several machines, go to Rule 2.

- Rule 2: Use the rules for the selection of operations and choose the machine whose operation has the highest priority. If the selected operation can be performed on multiple machines, go to Rule 3.
- Rule 3: Choose the fastest machine to perform the operation of Rule 2. In case of a tie between several machines, go to Rule 4.
- Rule 4: Choose any.
- Rule 1 makes operations begin as soon as possible (when a machine is released, then is chosen to be programmed with a corresponding operation). This contributes to the robustness of the program upon preventing the impact of unexpected events that may affect the rate of production.

In this case, the production efficiencies of the machines associated with each process stage were proved to be statistically equivalent (p value > 0.05) through a test of difference between means (t test). Therefore, the machines were concluded to have a similar processing time, which has been then incorporated into the proposed approach. This can be underpinned when considering their similarities in terms of technological level and age, which can surely contribute to validating the assumption.

3.2 Operation selection

After selecting the machine, it is necessary to apply a rule to select the operation. This step is very important, since it determines the way in which operations are ordered in the sequence. It is therefore advisable to establish an order of priority with the aim of minimizing as much as possible the average delay in deliveries. The chosen priorities are ordered from highest to lowest importance, as follows:

- Rule 1: Select the operation that corresponds to the item with the highest number of delay units. The percentage of delay units is obtained by calculating the difference between the amount of demanded units of a piece in the previous period, and the number of delay units in that period, divided by 100.
- Rule 2: Prioritize the operation of the item with more days late. *Days late* is defined as the difference between the current date and the due date.
- Rule 3: Prioritize the operation of the item with earliest due date.
- Rule 4: Prioritize the operation of the item whose following operations represent the greater processing time. It is calculated by adding the processing times of pending operations. If a given operation has several possible processing times (as they can be manufactured on different machines), take the greater processing time.
- Rule 5: Prioritize the item with higher average monthly demand.

In summary, a priority order has been chosen based on the need to reduce delays in deliveries, so “p” item is given to the operation of a part, depending on its level of delay (Rule 1), days of delay (Rule 2), earliest due date (Rule 3), longer pending processing time (Rule 4) and higher average monthly demand (Rule 5).

3.3 Dispatching algorithm scheme

In this section dispatching algorithm scheme is analyzed and presented.

(a) Start

- Place the first operations of the parts with their respective $r_{1,i}$ values in the subset of candidate operations.

- For each machine j, calculate fp_j value.
- Determine fp_{min} and its respective machine.

(b) Machine selection

- If $fp_{min} = \infty$, all operations have been programmed.
- Otherwise, select a machine according to fp_{min} (Rule 1). In case of a tie, select one according to Rules 2, 3 and 4 of operation selection.
- Upon selecting the machine, create the subset of candidate operations formed by the eligible operations that the machine is able to perform.

(c) Operation selection

- If there is just one candidate operation, this must be programmed.
- Otherwise, apply the priority rules for operation selection in order to choose the operation to program.

(d) Update

- Program the selected operation (k, i) setting its initial ($t_{start(k,i)}$) and end ($t_{end(k,i)}$) instants.

$$T_{start\,(k,i)} = rp_{k,i,j} \tag{1}$$

$$T_{end(k,i)} = T_{start(k,i)} + Dp_{k,i,j} \tag{2}$$

where D is the production unit demand related to the programmed operation.

- Locate the eligible operation in the subset of programmed operations with its respective initial and end instants.
- If it is not the final operation of the unit production “i”, move its next operation from N to E subset.
- j' is the machine associated to the next operation of unit production “i”:
  - If $j' = j$ (The same machine performs the consecutive operations of unit production “i”). In this case, the second operation cannot be initialized until the first operation ends, so the next operation (k + 1, i) is placed in the subset of candidate operations with availability time equals to finishing time of the previous operation $t_{end(k,i)}$ (see Fig. 4)

$$r_{(k+1,i)} = t_{end(k,i)} \tag{3}$$

- If $j' \neq j$ (the machines performing consecutive operations are different) and If $p_{(k+1,i,j')} \geq p_{(k,i,j)}$, the availability time (hour) of the next operation $r_{(k+1,i)}$ is equal to the initial date of the previous operation (k, i) plus the necessary time to produce and move a transfer lot of 25 production units from machine j to machine j' (see Fig. 5):

$$r_{(k+1,i)} = t_{initial(k,i)} + \frac{0.16 + 25p_{k,i,j}}{60} \tag{4}$$

- If $j' \neq j$ (the machines performing consecutive operations are different) and $p_{(k+1,i,j')} < p_{(k,i,j)}$, the availability time (hour) of the next operation $r_{(k+1,i)}$ is equal to the end date of the first operation (k, i) minus the processing time (k + 1, i) in machine j plus the necessary time to produce and move a transfer lot of 25 production units from machine j to machine j’ (see Fig. 6):

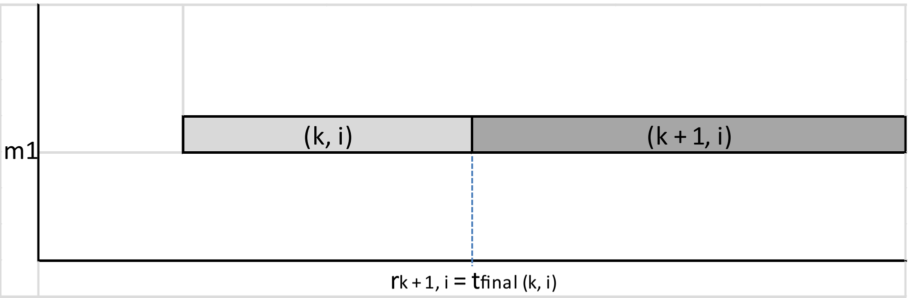

Fig. 4 Case: The same machine performs the consecutive operations of unit production "I". *Source*: Calleja and Pastor (2014)

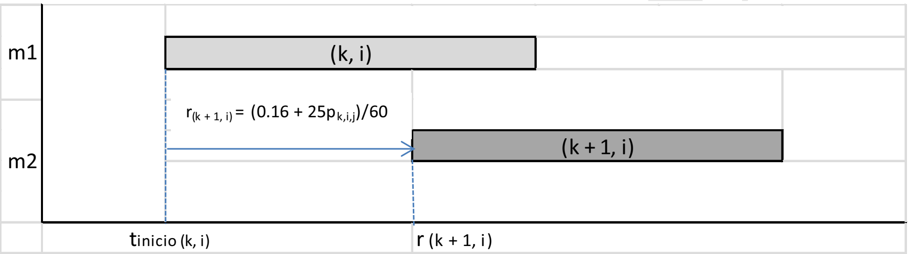

Fig. 5 Case: The machines performing consecutive operations are different—$p_{(k+1,i,j')} \geq p_{(k,i,j)}$. *Source*: Calleja and Pastor (2014)

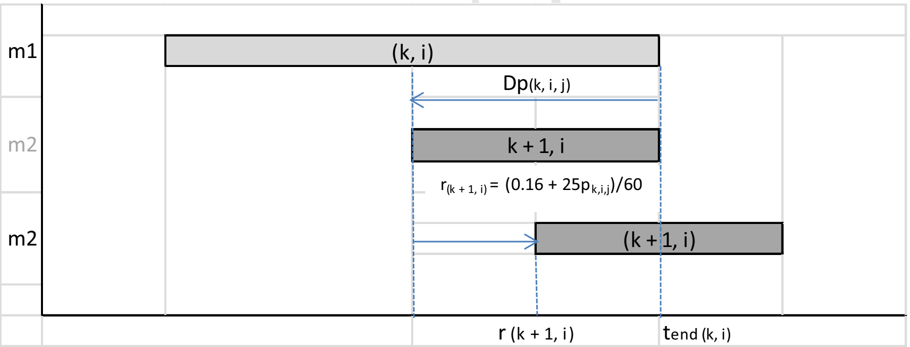

Fig. 6 Case: The machines that perform consecutive operations are different—$p_{(k+1,i,j')} < p_{(k,i,j)}$. *Source*: Calleja and Pastor (2015)

$$r_{(k+1,i)} = t_{end(k,i)} - Dp_{(k,i,j)} + \frac{0.16 + 25p_{k,i,j}}{60} \tag{5}$$

- Update fj′ values of the machine j′ in this way:
  - If the machine j′ has not already been used, $f_j = 0$.

  - If any operation in machine j has been programmed, then, $f_j = t_{end(k,i)}$, i.e., the machine j will have an availability time f_j that is equal to the finishing time of the last programmed operation in the machine j.

- Calculate $Rp_{k,i,j}$

$$Rp_{k,i,j} = m\acute{a}x\left(r_{k,i}, f_j\right) \qquad for\ all \quad (k,i) \in E_j \tag{6}$$

- Determine fp_{min}

$$fp_{min} = \min\left(rp_{k+1,i}\right) \tag{7}$$

- Return to step 1.
- Objective function

Calculate average tardiness according to:

$$Average\ tardiness = \frac{\sum_{i=1}^{n} max\,(0, Ci - di)}{n} \tag{8}$$

where C_i is the completion time of the job i, and d_i is the due date of job i.

3.4 Robustness rules

In this section, robustness rules that have been integrated into the design algorithm are shown. The purpose of these rules is to help creating a more robust program, i.e. more efficient, despite the random events that could occur during execution. Robustness rules used in the proposed program are:

(a) *Do not unnecessarily delay the processing operations*: This robustness rule has been implemented through Rule 1 of Machine Selection: The machine that is available sooner is prioritized; so that the operations begin as soon as possible which reduces the impact of future potential delays due to some unexpected setback that may alter the production rate.

(b) *Keep the bottleneck machines fed*: This approach is applied in order to avoid that bottleneck machines stop because of a random event occurring in a previous stage. If an inventory is not kept in front of a machine that turns out to be a bottleneck, and the machine that feeds the bottleneck suddenly fails, then the bottleneck would be idle and may not subsequently be able to recover the lost time. For the case study presented in this paper, the transfer batch between machines was calculated to be 25 units. This means that the second machine have at least a supply of 25 items while working in parallel with the first machine. With this transfer batch size, the makespan will be minimized and consequently, the tardiness will be also reduced.

(c) *Prioritize less flexible work*: This rule corresponds to the fourth rule of choice of operation. The least flexible work (those with more pending processing time) is programmed first. In this way, if a setback occurs, it is easier to reprogram the rest of the orders.

4 An illustrative example and experimental results

The dispatching algorithm proposed in this paper (18/17/G/Average tardiness) was applied in the finishing or completion process of a Colombian textile company. The stages of the process and its sequence are shown in Fig. 7.

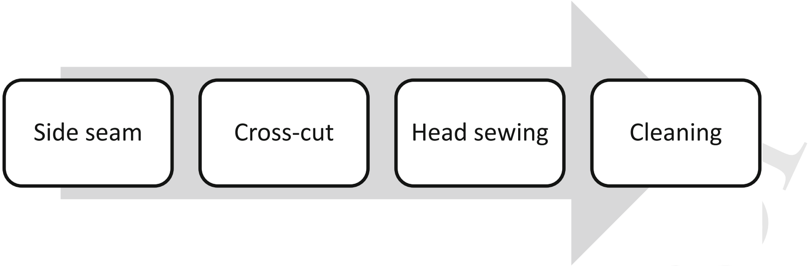

Fig. 7 Stages and sequence of finishing process

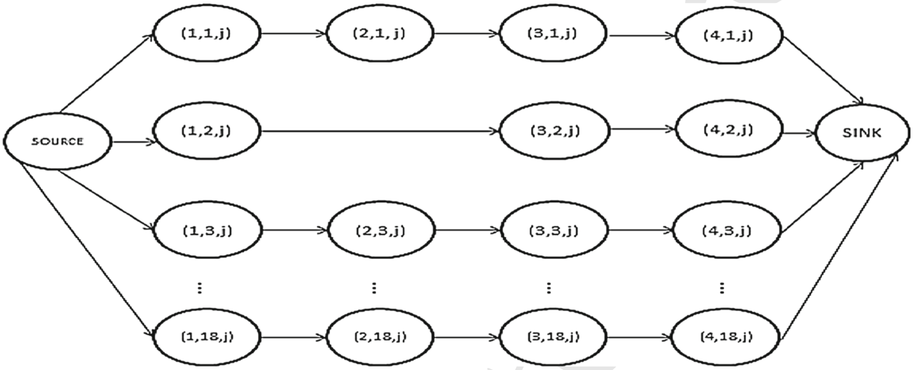

Fig. 8 Structure and workflow of jobs for the (18/17/G/Average tardiness) production system under study

The operation SIDE SEAM has 4 worker-machines CL1, CL2, CL3 and CL4; meanwhile, CROSS-CUT has 2 operators CT1 and CT2, HEAD SEWING has 7 worker-machines CC1, CC2, CC3, CC4, CC5, CC6 and CC7 and the CLEANING operation has 8 workers divided into 4 couples L1, L2, L3 and L4. On the closing date of October 15, the company has orders from 18 different items with different due dates. Each item has a different operation path so not all items go through all operations (see Fig. 8). In this company, the regular working day is eight hours with two hours of overtime each day. The diversity of items managed, and the production system complexity presents a significant challenge at the moment of programming operations.

Table 1 shows the information regarding processing times of operations to be performed for each item sample. Since resources for an operation are identical, the times are the same for any resource. In this table "xxx" means that the operation is not part of the operation path of the item.

Accordingly to Table 1, products such as SHEARED and STAMPED SEMIPLAYERA TOWEL 70 × 130 cm, 250 g/m^2, SEMIPLAYERA TOWEL 70 × 140, 365 g/m^2 do not go through the seam side; while MULERAS 30 × 140 cm and smooth PONCHOS do not require cross-section; all other items go through the four operations with different processing times.

This sets up a *flexible job-shop* system, which is currently managed by the company under a FIFO (First in First Out) policy where orders are entered into the production system according to their order of arrival. These orders are indicated in the delivery schedule presented below (see Table 2). In this schedule, "O" means *October*, "N" *November,* and "D" *December*. For

Table 1 Processing times for each item in each finishing system operation. *Source*: Author

Item (i = 1, 2, 3,..., 18)	Processing time (min/unit)			
	CL (k = 1)	CT (k = 2)	CC (k = 3)	L (k = 4)
Sheared and printed towel, 30 × 45cm^2, 380g/m^2	0.38	0.04	0.9	0.13
Smooth mulera, 35 × 150 cm^2, 110 g/m^2	0.92	xxx	1.4	0.84
Sheared and printed towel, 60 × 120 cm^2, 320 g/m^2	1	0.083	1.2	0.67
Stamped and hand towel, 30 × 50 cm^2, 310 g/m^2	0.42	0.04	0.6	0.14
Smooth mulera, 30 × 140 cm^2, 110 g/m^2	0.86	xxx	1.2	0.67
Hand and stamped terry towel, 35 × 60 cm^2, 310 g/m^2	0.43	0.06	0.75	0.67
Smooth hand towel, 35 × 60 cm^2, 310 g/m^2	0.43	0.06	0.75	0.67
Stamped and shared towel, 60 × 120 cm^2, 320 g/m^2	1	0.083	1.2	0.67
Jacquard and crepé body towel, 60 × 120 cm^2, 450 g/m^2	1	0.083	1.2	0.67
Smooth poncho	0.86	xxx	1.2	0.67
Stamped hand towel 40 × 66 cm^2, 380 g/m^2	0.55	0.6	0.8	0.25
Semiplayera towel, 70 × 140 cm^2, g/m^2	xxx	0.1	1.4	0.91
Body towel, 60 × 120 cm^2, 365 g/m^2	1	0.083	1.2	0.67
Face towel, 50 × 90 cm^2, 365 g/m^2	0.75	0.07	1	0.42
Hand towel, 35 × 60 cm^2, 365 g/m^2	0.43	0.06	0.75	0.67
Stamped and sheared towel, 25.4 × 86.36 cm^2, 320 g/m^2	0.51	0.04	0.72	0.2
Sheared and stamped semiplayera towel, 70 × 130 cm^2, 250 g/m^2	xxx	0.1	1.4	0.2
Hand towel, 30 × 50 cm^2, 310 g/m^2	0.42	0.04	0.6	0.14

Table 2 Delivery schedule. *Source*: Author

Item	Description	O3	O8	O10	O30	N3.	N6.	N7	N8	N9	N13	N15	N16	N20	D5	D7	Total	Tardiness	% of tardiness	Days of tardiness
1	Sheared and printed towel, 30×45 cm^2, 380g/m^2							200									200	0	0	0
2	Smooth mulera, 35×150 cm^2, 110 g/m^2				2000												2000	0	0	0
3	Sheared and printed towel, 60×120 cm^2, 320 g/m^2									1002							1002	0	0	0
4	Stamped and hand towel, 30×50 cm^2, 310 g/m^2							24,000									24,000	0	0	0
5	Smooth mulera, 30×140 cm^2, 110 g/m^2								4500								4500	0	0	0
6	Hand and stamped terry towel, 35×60 cm^2, 310 g/m^2										3000	360	3000			2000	8360	0	0	0
7	Smooth hand towel, 35×60 cm^2, 310 g/m^2													6000	3000	7500	16,500	0	0	0
8	Stamped and shared towel, 60×120 cm^2, 320 g/m^2								110								110	0	0	0
9	Jacquard and crepé body towel, 60×120 cm^2, 450 g/m^2					1002											1002	0	0	0

Table 2 continued

Item	Description	O3	O8	O10	O30	N3.	N6.	N7	N8	N9	N13	N15	N16	N20	D5	D7	Total	Tardiness	% of tardiness	Days of tardiness
10	Smooth poncho											5950					5950	0	0	0
11	Stamped hand towel 40 × 66 cm^2, 380 g/m^2						100										100	0	0	0
12	Semiplayera towel, 70 × 140 cm^2, g/m^2											25					25	0	0	0
13	Body towel, 60 × 120 cm^2, 365 g/m^2											50					50	0	0	0
14	Face towel, 50 × 90 cm^2, 365 g/m^2											50					50	0	0	0
15	Hand towel, 35 × 60 cm^2, 365 g/m^2											25					25	0	0	0
16	Stamped and sheared towel, 25.4 × 86.36 cm^2, 320 g/m^2			45,000													45,000	45,000	100	5
17	Sheared and stamped semi-playera towel, 70 × 130 cm^2, 250 g/m^2	1000															1000	1000	100	12
18	Hand towel, 30 × 50 cm^2, 310 g/m^2		2500														2500	2500	100	7

this example and starting from October 15th, the company has not delivered 3 orders and should evaluate other programming technique that allows ensuring customer loyalty with fewer late deliveries.

Meanwhile, Table 3 presents the setup times and available resources for each operation referred to the products offered to the customers. In this example, *setup time* indicates the amount of time needed to prepare the towels before manufacturing. As it is shown in Table 3, this time depends on the kind of operation to be performed. First, *setup time* before SIDE SEAM or HEAD SEWING involves removing any excess of thread from previous processes and *setup time* before CROSS-CUT or CLEANING refers to properly putting the towels on a table, to avoid nonconforming units while respectively cutting or cleaning.

In Table 3, available resources for each operation and item are also displayed. It has to be noted that not all the resources that are able to perform a specific operation, have been assigned as *available* to produce a certain item. For the SIDE SEAM operation, there are 4 resources (worker-machines); however, 2 of the 4 workers (CL3, CL4) are not capable to perform this operation for 6 items because these products are printed or jacquard; features that require workers with a lot of experience in order to ensure high-quality products. This becomes a *quality restriction* for the scheduling algorithm; therefore those workers must not be programmed with these items.

Concerning the CROSS-CUT and CLEANING operations, all the resources are available for all the items. For the HEAD SEWING operation, there are 7 resources (worker-machines); however, 3 of the 7 workers (CC5, CC6, CC7) are not skilled enough to execute this operation for 6 items, due to the same reasons mentioned above in the SIDE SEAM operation.

Table 4 presents the average monthly demand for each item offered by the company under study. This information is useful at the moment of prioritizing the item with higher average monthly demand (See rule 5 in Operation Selection) in case of ties in the previous rules. This will allow ensuring the highest service levels for the most demanded items in this company. A significant number of delay days in these items may represent loss of customer loyalty and possible declining sales. For this reason *dispatching algorithm* is taken it into account as a policy.

The delivery schedule with the current policy (FIFO) and the technical proposal (dispatching algorithm) were evaluated. The results of applying dispatching algorithm are shown in Table 5.

Upon comparing the results of the current methodology (FIFO) and *dispatching algorithm* as the proposed methodology, in terms of average tardiness, Table 6a shows that the FIFO policy currently used in the production system of the textile company, offers late delivery in all the orders with an average of 25.2 days/order; while Table 6b, showing the results obtained with the implementation of the dispatching algorithm, shows only three items with due date, obtaining an average delay index equals to 1.91 days/order. This constitutes a significant reduction of 92.42% representing a positive impact on customer satisfaction and loyalty due more punctual deliveries.

Figure 9 shows a comparison between FIFO and dispatching programming policies, where 22 out of 23 orders have less delay days in dispatching algorithm. It is also noticed that 20 out of 23 orders did not show any delay at the time of delivery.

However, 10 tests were performed (including test 1 described in this paper), to validate these results with data provided by the company. Each test has been performed for each algorithm and compared with the discrepancy percentage between average tardiness of both methods. If discrepancy percentage is greater than 0, means that the value obtained with dispatching algorithm is better than the algorithm used by the company. If it is less than 0, the algorithm applied by the company is better than the dispatching algorithm. Finally, if

Table 3 Setup times and available resources for each operation referred to the items offered by the textile company. *Source*: Author

Item	Operation (resources)	Setup time (min)
Sheared and printed towel, 30 × 45 cm^2, 380g/m^2	SIDE SEAM (CL1, CL2)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4)	1
	CLEANING (L1, L2, L3, L4)	0.5
Smooth mulera, 35 × 150 cm^2, 110 g/m^2	SIDE SEAM (CL1, CL2, CL3, CL4)	3
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Sheared and printed towel, 60 × 120 cm^2, 320 g/m^2	SIDE SEAM (CL1, CL2)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4)	1
	CLEANING (L1, L2, L3, L4)	0.5
Stamped and hand towel, 30 × 50 cm^2, 310 g/m^2	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Smooth mulera, 30 × 140 cm^2, 110 g/m^2	SIDE SEAM (CL1, CL2, CL3, CL4)	3
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Hand and stamped terry towel, 35 × 60 cm^2, 310 g/m^2	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Smooth hand towel, 35 × 60 cm^2, 310 g/m^2	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Stamped and shared towel, 60 × 120 cm^2, 320 g/m^2	SIDE SEAM (CL1, CL2)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4)	1
	CLEANING (L1, L2, L3, L4)	0.5
Jacquard and crepé body towel, 60 × 120 cm^2, 450 g/m^2	SIDE SEAM (CL1, CL2)	3
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4)	1
	CLEANING (L1, L2, L3, L4)	0.5

Table 3 continued

Item	Operation (resources)	Setup time (min)
Smooth poncho	SIDE SEAM (CL1, CL2, CL3, CL4)	3
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Stamped hand towel 40 × 66 cm^2, 380 $\mathrm{g/m}^2$	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Semiplayera towel, 70 × 140 cm^2, $\mathrm{g/m}^2$	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Body towel, 60 × 120 cm^2, 365 $\mathrm{g/m}^2$	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Face towel, 50 × 90 cm^2, 365 $\mathrm{g/m}^2$	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Hand towel, 35 × 60 cm^2, 365 $\mathrm{g/m}^2$	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5
Stamped and sheared towel, 25.4 × 86.36 cm^2, 320 $\mathrm{g/m}^2$	SIDE SEAM (CL1, CL2)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4)	1
	CLEANING (L1, L2, L3, L4)	0.5
Sheared and stamped semiplayera towel, 70 × 130 cm^2, 250 $\mathrm{g/m}^2$	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4)	1
	CLEANING (L1, L2, L3, L4)	0.5
Hand towel, 30 × 50 cm^2, 310 $\mathrm{g/m}^2$	SIDE SEAM (CL1, CL2, CL3, CL4)	2
	CROSS-CUT (CT1, CT2)	0.5
	HEAD SEWING (CC1, CC2, CC3, CC4, CC5, CC6, CC7)	1
	CLEANING (L1, L2, L3, L4)	0.5

Table 4 Average monthly demand for the items offered by the textile company. *Source*: Author

Item	Average monthly demand
Smooth hand towel, 35 × 60 cm^2, 310 g/m^2	52,000
Hand and stamped terry towel, 35 × 60 cm^2, 310 g/m^2	35,600
Sheared and printed towel, 30 × 45 cm^2, 380g/m^2	15,000
Hand towel, 30 × 50 cm^2, 310 g/m^2	12,300
Stamped and hand towel, 30 × 50 cm^2, 310 g/m^2	12,000
Semiplayera towel, 70 × 140 cm^2, g/m^2	10,000
Sheared and stamped semiplayera towel, 70 × 130 cm^2, 250 g/m^2	8700
Sheared and printed towel, 60 × 120 cm^2, 320 g/m^2	5000
Face towel, 50 × 90 cm^2, 365 g/m^2	4300
Stamped and sheared towel, 25.4 × 86.36 cm^2, 320 g/m^2	3750
Stamped and shared towel, 60 × 120 cm^2, 320 g/m^2	3500
Smooth mulera, 30 × 140 cm^2, 110 g/m^2	3200
Smooth poncho	3100
Body towel, 60 × 120 cm^2, 365 g/m^2	3000
Smooth mulera, 35 × 150 cm^2, 110 g/m^2	2800
Stamped hand towel 40 × 66 cm^2, 380 g/m^2	2300
Jacquard and crepé body towel, 60 × 120 cm^2, 450 g/m^2	1520
Hand towel, 35 × 60 cm^2, 365 g/m^2	1300

the percentage is equal to 0, both methods had the same performance. A summary of the discrepancy percentages obtained in the tests is described in Table 7.

A t test with $\alpha = 0.05$ has been performed in order to determine if both methods are statistically different (see Table 8). According to this test, with a probability of 0.0000001% (one-tailed test) and 0.0000002% (two-tailed test), it is concluded that both algorithms have different performances. An average discrepancy percentage of 82.1% was obtained as a final result (see Table 7). In conclusion, *dispatching algorithm* provides a better performance than FIFO and it could be qualified as significant based on the differences presented in Fig. 10.

It is noticed that the proposed algorithm selects the operation with the highest priority among all candidate operations. However, the textile company in this study selects the product with the highest priority and programs all its operations. With this method, all the operations that correspond to a specific product used to be performed in parallel, which affects the average tardiness. FIFO also generates longer waiting times for the machines, which could be performing an operation. This does not occur in dispatching algorithm because it prioritizes

Table 5 Sequence of operations for the finishing process through dispatching algorithm—Test 1. *Source*: Author

Operation number	Item	Resource	Start (h)	End (h)
1	17	CT1	0.00	1.62
1	18	CL1	0.00	17.39
1	16	CL2	0.00	382.53
1	2	CL3	0.00	30.76
1	11	CL4	0.00	0.95
1	12	CT2	0.00	0.05
2	17	CC3	0.59	23.93
2	2	CC1	0.59	47.27
2	12	CC2	0.59	1.19
2	11	CT2	0.88	0.98
1	4	CL4	0.95	167.65
1	5	CL4	0.95	65.50
1	6	CL4	0.95	60.90
1	13	CL4	0.95	1.82
1	14	CL4	0.95	1.61
1	15	CL4	0.95	1.16
1	10	CL4	0.95	86.28
1	7	CL4	0.95	119.23
2	15	CT2	1.17	1.20
3	12	L3	1.19	1.58
3	11	L4	1.31	2.66
2	5	CC4	1.45	91.47
2	10	CC5	1.45	120.47
3	15	CC6	1.49	1.82
2	14	CT2	1.58	1.65
2	13	CT1	1.78	1.86
4	15	L3	1.82	2.11
3	14	CC7	1.99	2.84
3	13	CC6	2.28	3.30
4	11	L3	2.35	2.77
4	14	CC6	2.67	3.03
4	13	CC6	3.02	3.59
2	18	CT2	15.74	17.48
3	18	CC6	15.99	41.01
1	9	CL1	17.39	34.14
1	1	CL1	17.39	18.67
1	8	CL1	17.39	17.55
1	3	CL1	17.39	34.12
2	8	CT1	17.44	17.60
3	8	CC2	17.94	20.16
2	1	CT1	18.56	18.71
3	1	CC2	18.93	21.95

Table 5 continued

Operation number	Item	Resource	Start (h)	End (h)
4	8	L1	19.21	20.45
3	2	L2	19.62	47.55
3	17	L1	20.68	24.09
4	1	L2	21.57	22.00
2	3	CT1	32.77	34.16
2	9	CT2	32.79	34.18
3	3	CC2	33.27	53.33
3	9	CC3	33.29	53.35
4	18	L4	35.23	41.05
3	5	L3	41.50	91.76
4	3	L4	42.42	53.62
4	9	L1	42.44	53.64
2	6	CT2	52.56	60.79
3	6	CC7	52.88	157.40
3	10	L3	54.31	120.76
4	6	L1	64.33	157.69
2	7	CT2	102.76	118.99
3	7	CC3	103.08	309.35
4	7	L1	125.38	309.64
2	4	CT2	151.67	168.28
3	4	CC1	151.92	391.94
4	4	L4	335.99	391.83
2	16	CT2	352.55	378.91
3	16	CC2	352.85	892.87
4	16	L2	742.95	896.05

Table 6 Delay of orders by FIFO policy—Test 1. *Source*: Author

Item	Order	End date	Delivery date	Delay (days)
(a)				
1		27/11/2013	07/11/2013	20
2		24/11/2013	30/10/2013	25
3		10/12/2013	09/11/2013	31
4		29/11/2013	07/11/2013	22
5		07/12/2013	08/11/2013	29
6	1	11/12/2013	13/11/2013	28
6	2	12/12/2013	15/11/2013	27
6	3	17/12/2013	16/11/2013	31
6	4	23/12/2013	07/12/2013	16
7	1	19/12/2013	20/11/2013	29
7	2	22/12/2013	05/12/2013	17
7	3	26/12/2013	07/12/2013	19

Table 6 continued

Item	Order	End date	Delivery date	Delay (days)
8		10/12/2013	08/11/2013	32
9		27/11/2013	03/11/2013	24
10		14/12/2013	15/11/2013	29
11		27/11/2013	06/11/2013	21
12		16/12/2013	15/11/2013	31
13		16/12/2013	15/11/2013	31
14		16/12/2013	15/11/2013	31
15		16/12/2013	15/11/2013	31
16		16/10/2013	10/10/2013	6
17		21/10/2013	03/10/2013	18
18		16/10/2013	08/10/2013	8
Average delay				24.17
(b)				
1		18/10/2013	07/11/2013	0
2		20/10/2013	30/10/2013	0
3		21/10/2013	09/11/2013	0
4		24/11/2013	07/11/2013	17
5		25/10/2013	08/11/2013	0
6	1	31/10/2013	13/11/2013	0
6	2	31/10/2013	15/11/2013	0
6	3	31/10/2013	16/11/2013	0
6	4	31/10/2013	07/12/2013	0
7	1	15/11/2013	20/11/2013	0
7	2	15/11/2013	05/12/2013	0
7	3	15/11/2013	07/12/2013	0
8		18/10/2013	08/11/2013	0
9		19/10/2013	03/11/2013	0
10		28/10/2013	15/11/2013	0
11		16/10/2013	06/11/2013	0
12		16/10/2013	15/11/2013	0
13		16/10/2013	15/11/2013	0
14		16/10/2013	15/11/2013	0
15		16/10/2013	15/11/2013	0
16		13/01/2014	20/01/2014	0
17		18/10/2013	03/10/2013	15
18		20/10/2013	08/10/2013	12
Average delay				1.91

in each time the most important operation in the first available machine. This decreases the likelihood of refunds given for late deliveries and the resulting cost overruns for transportation and inventory; and also decreases penalties for infringement and discounts offered to the customers to keep their loyalty.

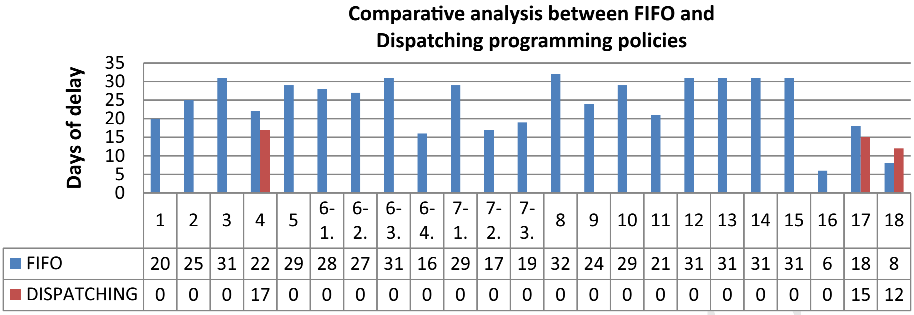

Fig. 9 Comparative analysis between FIFO and dispatching programming policies—Test 1. *Source*: Author

Table 7 Summary of delay and discrepancy percentage between dispatching and FIFO algorithms

Test number	Average tardiness (days/order)		Discrepancy percentage
	Dispatching	FIFO	
1	1.91	24.17	92.1
2	0	21.98	100.0
3	5.54	28.34	80.5
4	2.03	24.47	91.7
5	9.22	32.92	72.0
6	5.38	29.60	81.8
7	10.82	34.60	68.7
8	3.88	29.07	86.7
9	13.39	38.66	65.4
10	5.37	30.75	82.5
Average discrepancy			82.1

Table 8 *T* test for determining difference between dispatching and FIFO algorithms

	Dispatching algorithm	FIFO algorithm
Mean	5754	29,456
Variance	17,95,37,822	26,10,62,044
Sample size	10	10
Hypothetical difference	0	
Degrees of freedom	17	
t-statistic	11,29,17,905	
P(T <= t) − one tailed	12,685E−09	
Critical t value (one tailed)	1,73,96,0673	
P(T <= t)− two tailed	25,371E−09	
Critical t value (two tailed)	2,10,981,558	

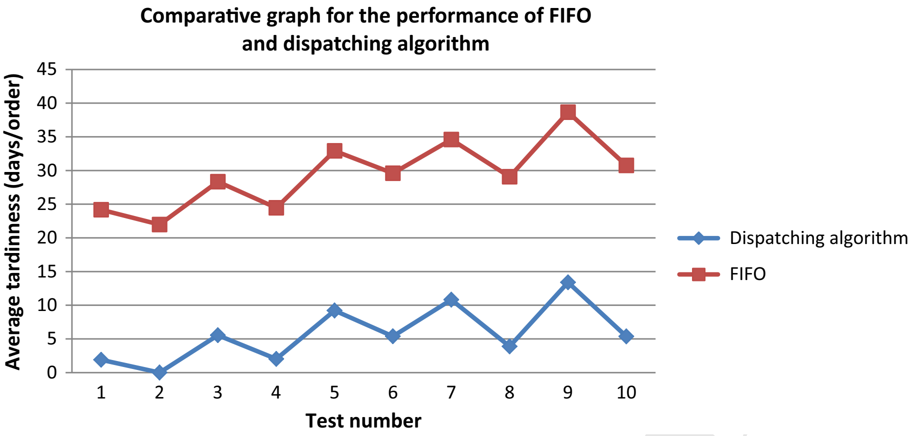

Fig. 10 Comparative analysis between FIFO and dispatching algorithms in 10 tests. *Source*: Author

5 Conclusion

Flexible job-shop scheduling is of significant importance to the implementation of real-world manufacturing systems. In order to capture the dynamic and multi-objective nature of flexible job-shop scheduling, and provide different trade-offs among objectives, this paper develops a dispatching algorithm for production programming of flexible job-shop systems in the textile sector. This paper formulated a real-world production-scheduling problem and also provided an efficient tool to solve it. The current study can be extended to consider other types of production environments. It should thus be useful to both practitioners and researchers. The experimental results show that the proposed algorithm is better than the results provided by the existing dispatching rule. The research carried out in this paper opens up opportunities to study and improve flexible job-shop problems. Future developments of our research are extending it to other scenarios. It would also be interesting to see how good our rules are compared to more complex heuristics algorithms. Regarding the managerial implications, the proposed model is very useful for both production managers and other practitioners from the textile sector to underpin the scheduling process in FJS systems. The model is more realistic and reliable compared to FIFO, which tends to be the most used scheduling rule in this industry. However, it has been evidenced that this is not the most beneficial alternative when considering minimum average tardiness. Thus, by applying this method, delivery delays can be minimized and consequently, customer satisfaction rates may increase. Additionally, the use of this model must be supported by the implementation of a DSS (Decision Support System) for an effective and quick decision-making during the planning process.

References

Alvarez-Valdes, R., Fuertes, A., Tamarit, J. M., Giménez, G., & Ramos, R. (2005). A heuristic to schedule flexible job-shop in a glass factory. *European Journal of Operational Research*, *165*(2005), 525–534.

Baker, K. R. (2005). *Elements of sequencing and scheduling*. Hanover, NH: Tuck School of Business.

Barrios, M. A. O., Caballero, J. E., & Sánchez, F. S. (2015). A methodology for the creation of integrated service networks in outpatient internal medicine. In *Ambient intelligence for health* (pp. 247–257). Springer.

Bozek, A., & Wysocki, M. (2015). Flexible job shop with continuous material flow. *International Journal of Production Research*, *53*(4), 1273–1290.

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by Tabu search. *Annals of Operations Research*, *41*(3), 157–183.

Calleja, G., & Pastor, R. (2014). A dispatching algorithm for flexible job-shop scheduling with transfer batches: An industrial application. *Production Planning & Control*, *25*(2), 93–109.

De Felice, F., & Petrillo, A. (2013). Simulation approach for the optimization of the layout in a manufacturing firm. *24th IASTED international conference on modelling and simulation*, MS 2013; Banff, AB; Canada; 17 July 2013 through 19 July 2013 (pp. 152–161).

Demir, Y., & Işleyen, S. K. (2014). An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations. *International Journal of Production Research*, *52*(13), 3905–3921.

Digiesi, S., Mossa, G., & Mummolo, G. (2013). A sustainable order quantity model under uncertain product demand. *7th IFAC conference on manufacturing modelling, management, and control, MIM 2013* (pp. 664–669). Saint Petersburg; Russian Federation; 19 June–21 June.

Fattahi, F., & Fallahi, A. (2010). Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *CIRP Journal of Manufacturing Science and Technology*, *2*, 114–123.

Fattahi, P., Hosseini, S. M. H., Jolai, F., & Tavakkoli-Moghaddam, R. (2014). A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. *Applied Mathematical Modelling*, *38*, 119–134.

Gholami, O., & Sotskov, Y. N. (2014). Solving parallel machines job-shop scheduling problems by an adaptive algorithm. *International Journal of Production Research*, *52*(13), 3888–3904.

Guo, W. J. (2006). *Algorithms for two-stage flexible flow shop scheduling with fuzzy processing times*. NanJing: NanJing University of Science & Technology.

Hassin, R., & Shani, M. (2005). Machine scheduling with earliness, tardiness and nonexecution penalties. *Computers & Operations Research*, *32*, 683–705.

Herazo-Padilla, N., Montoya-Torres, J. R., Isaza, S. N., & Alvarado-Valencia, J. (2015). Simulation-optimization approach for the stochastic location-routing problem. *Journal of Simulation*, *9*(4), 296–311.

Hildebrandt, T., Heger, J., & Scholz-Reiter, B. (2010). Towards improved dispatching rules for complex shop floor scenarios—A genetic programming approach. GECCO'10, July 7–11, 2010. Portland, Oregon, USA.

Hu, H. (2015). Adaptive scheduling model in hybrid flowshop production control using petri net. *International Journal of Control and Automation*, *8*(1), 233–242.

Jansen K, Mastrolilli M, & Solis-Oba R, (2000). Approximation algorithms for Flexible Job Shop Problems. In: *Lecture notes in computer science, Vol. 1776. Proceedings of the fourth Latin American symposium on theoretical informatics* (pp. 68–77). Berlin: Springer.

Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2009). A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers and Operations Research*, *36*(2), 358–378.

Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality Approach for Flexible Job-shop Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic. *Journal of Mathematics and Computers in Simulation*

Karimi-Nasab, M., & Modarres, M. (2015). Lot sizing and job shop scheduling with compressible process times: A cut and branch approach. *Computers & Industrial Engineering*, *85*, 196–205.

Kurz, M. E., & Askin, R. G. (2004). Scheduling flexible flow lines with sequence dependent setup times. *European Journal of Operational Research*, *159*(1), 66–82.

Logendran, R., Carson, S., & Hanson, E. (2005). Grouping scheduling in flexible flow shops. *International Journal of Production Economics*, *96*(2), 143–155.

Pezzella, F., Morganti, G., & Ciaschetti, G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers & Operations Research*, *35*, 3202–3212.

Pinedo, M. (2001). *Scheduling: Theory, algorithms, and systems*. Upper Saddle River, NJ: Prentice Hall.

Prot, D., Bellenguez-Morineau, O., & Lahlou, C. (2013). New complexity results for parallel identical machine scheduling problems with preemption, release dates and regular criteria. *European Journal of Operational Research*, *231*, 282–287.

Riane, F., Artiba, A., & Elmaghraby, S. E. (2002). Sequencing a hybrid two-stage flow shop with dedicated machines. *Int. J. Prod. Res.*, *40*, 4353–4380.

Shen, X.-N., & Yao, X. (2015). Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. *Information Sciences*, *298*, 198–224.

Sotskov, Y. N., & Gholami, O. (2015). Mixed graph model and algorithms for parallel-machine job-shop scheduling problems. *International Journal of Production Research*, *55*(6), 1549–1564.

Sun, D.-H., He, W., Zheng, L.-J., & Liao, X.-Y. (2014). Scheduling flexible job shop problem subject to machine breakdown with game theory. *International Journal of Production Research*, *52*(13), 3858–3876.

Türkylmaz, A., & Bulkan, S. (2015). A hybrid algorithm for total tardiness minimization in flexible job shop: Genetic algorithm with parallel VNS execution. *International Journal of Production Research*, *53*(6), 1832–1848.

Wang, S., & Liu, M. (2013). A heuristic method for two-stage hybrid flow shop with dedicated machines. *Computers & Operations Research*, *40*, 438–450.

Yokoyama, M. (2004). Scheduling for two-stage production system with setup and assembly operations. *Computers & Operations Research*, *31*, 2063–2078.