Expert Systems With Applications 137 (2019) 130–156

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Review

The blocking flow shop scheduling problem: A comprehensive and conceptual review

Hugo Hissashi Miyata, Marcelo Seido Nagano*

Production Engineering Department, São Carlos School of Engineering, University of São Paulo, Av. Trabalhador São-carlense, 400, 13566-590 São Carlos, SP, Brazil

ARTICLE INFO

Article history:
Received 20 February 2019
Revised 29 June 2019
Accepted 30 June 2019
Available online 30 June 2019

Keywords:
Scheduling
Blocking flow shop
Solution methods
Literature review
Classification

ABSTRACT

This paper presents a literature review on the *m*-machine flow shop scheduling problem with blocking conditions. In total, 139 papers are reviewed and classified, ranging from 1969 up to early 2019. Results showed that makespan was the objective function most adopted by the researchers, with 62% of the total covered, followed by multi-objective based (12%), total flow time-based (11%), due date-based (7%), stochastic based functions (6%) and cycle time (3%). Regarding the purpose of the paper, approximately 92% of the papers proposed solution methods, where 76% of the papers developed heuristic methods and 16% of exact methods, and 8% of the papers considered the analysis of the problem and literature reviews. Directions for future researches include the proposition of solution methods for mono-objective functions as total flow time-based and due date-based, development of solution methods for the *m*-machine flow shop with RCb and RCb* constraints and adoption of more than one additional constraint to the problem.

1. Introduction

Scheduling deals with the allocation of resources over a given time periods with the goal to minimize one or more objectives. As a decision-making process, scheduling plays an important role for manufacturing and production systems, transportation, distribution settings and even some types of service (Pinedo, 1995). The efforts of the researchers in this area have been contributed significantly, since classical problems of the literature until practical applications ((Maccarthy & Liu, 1993), (Gupta & Stafford Jr, 2006), (Allahverdi et al., 2008), (Allahverdi, 2015)). From the classical scheduling problems, permutation flow shop has been attracted one of the most attention by the researchers, since its first publication (Johnson, 1954). This environment is characterized by processing *n* jobs on *m* machines. All jobs are processed in the same order, i.e., each job starts on machine 1, then it is processed on machine 2, up to the last machine. The intermediate storage capacity between machines are considered infinite and machines are always available for processing jobs (Maccarthy & Liu, 1993). Although the classic PFSP constraint considers the buffers capacity as infinite, in practice, some flow shop configurations deal with or without limited buffers, causing a blocking in the jobs processing.

According to Hall and Sriskandarajah (1996), one of the reasons of occurrence of blocking is the lack of intermediate storage between machines or production stages. In practical applications, (Grabowski & Pempera, 2000), because of technological constraints, blocking occurs in some production stages of concrete blocks. In chemical industry, partially-processed chemical products sometimes must be kept in machines because of lack of intermediate storage (Liu & Kozan, 2009). Chen et al. (2014) considered an electronic manufacturing shop that assembles and tests printed circuit boards (PCBs) for consumer electronic products. After assembly, a batch of PCBs is subject to a sequence of tests in two environmental stress screening (ESS) chambers: The first chamber is used to subject the batch to vibration test and the second chamber is used test it to extremes temperature. The technological requirements does not allow that a batch wait between two ESS chambers. As a result, a batch must be blocked inside the first ESS chamber if the second ESS is busy. Other production systems, such as chemical, pharmaceutical industries (Hall & Sriskandarajah, 1996), just-in-time production lines (Prasad, Rajendran, & Chetty, 2006) and robotic cells (Ribas & Companys, 2015) can be modeled as blocking flow shop. Another possible cause of blocking lies in the production technology itself, e.g., temperature or other characteristics of the materials, requires that finished job remains on machine to avoid deterioration or additional costs. In the treatment of industrial waste, different waste types are brought by trucks and are unloaded in tanks. Each product needs to be processed on a blender. The product flows slowly from the tank to the blender.

* Corresponding author.
E-mail addresses: hhmiyata@usp.br (H.H. Miyata), drnagano@sc.usp.br, drnagano@usp.br (M.S. Nagano).

https://doi.org/10.1016/j.eswa.2019.06.069
0957-4174/

The tank is ready to processing a new job only when the previous job is fully transferred in the blender. In the processing of metallic parts, each part needs first to be heated at the right temperature in one of several ovens (first stage), in order to be prepared for processing on a single press (second stage). The oven is blocked to process the next lot of parts until the last part of the previous lot is completed on the press (Martinez et al., 2006). Gong, Tang, and Duin (2010) considered a two-stage flow shop in iron and steel industry, in the operations of soaking an rolling, represented as a batching machine and a single machine, respectively. After finished, the set of jobs must remain inside the batching machine to avoid material deterioration, blocking new operations to be processed. In cider brewing, one cannot melt apples of different customers. First, apples are poured in a bath and then pressed to make apple juice. The blocking occurs because new customer apples cannot be poured in bath before all of the first costumer's apples have been pressed (Trabelsi, Sauvey, & Sauer, 2012). The blocking flow shop with two machine and makespan minimization is equivalent to the no-wait flow shop problem with two machine and makespan minimization (Reddi & Ramamoorthy, 1972). Both can be solved by Gilmore and Gomory Algorithm (Gilmore & Gomory, 1964) in polynomial time. For $m > 2$, the problem is proved to be NP-Hard ((Röck, 1984), (Martinez et al., 2006), (Fernandez-viagas, Leisten, & Framinan, 2016)). Because of the complex nature of the problem, exact methods are used to solve a small number of jobs. For larger instances, the use of heuristics methods is more common. Hall and Sriskandarajah (1996) presented a review of problems with no-wait and blocking in process and covered papers from 1960 to 1993. Until the last year covered by the aforementioned authors, most of the publications are directed to the study of complexity analysis and development of exact methods for the $F_m|block|C_{max}$ problem. Emmons and Vairaktarakis (2013) presented a literature review on the applications of flow shop with blocking. The authors provided a review of some complexity analysis and the main solution methods for the problem. Since then, a significant amount of researches has been developed for the problem, mainly, the proposition of solution methods.

This paper presents a literature review for the m-machine blocking flow shop. First, the motivation of this paper lies in identifying the literature state, from the first paper found until the most recent published so far. Another reason is draw the general panorama of the area to bring to light the development of solution methods and trends to the study of new problems. This paper can also be used as a guide for the interested audience as researchers, production managers and beginners in the area, since it gathers a list of papers directly related with the problem. Finally, with the literature state mapped, identification of gaps in the area and suggestions for future researches can be pointed out.

The remaining of the paper is organized as follows. Section 2 presents the notations and the classification of the static scheduling problem. Section 3, an up-to-date research review for the m-machine blocking flow shop is given. Section 4 shows some quantitative analysis of the results given by the problem attached, objective-function adopted and the purpose of the papers. In the last section, conclusions and directions for future research are discussed.

2. Definitions and notations

The m-machine blocking flow shop scheduling problem can be stated as follows: There are n jobs that must be processed through m machines. The processing of job i on machine j is denoted as operation $O_{i,j}$. Each operation i ($i = 1, ..., n$) has a processing time associated with each machine ($j = 1, \ldots, m$), defined as $p_{i,j}$. Buffers between machines are not available, i.e., intermediate storage capacity is considered zero. Consequently, a job finished on machine j blocks it until next downstream machine is available. The specific literature identifies two different type of blocking occurrences:

(a) Release when starting blocking (*RSb*|*block*): upstream machine remains blocked by job i until this job i starts on the next downstream machine (Fig. 1). In practical applications, the occurrence of *RSb*|*block* constraint can be seen in the processes of block concrete (Grabowski & Pempera, 2000), robotic cells (Carlier et al., 2010), iron and steel industries (Gong, Tang, & Duin, 2010), electronic manufacturing shop (Chen et al., 2014), among others. For convenience, the condition *RSb* will identified as *block* throughout the sections.

(b) Release when completing blocking (RCb*): machine will be available to processing the next job after the previous job leaves the subsequent machine. RCb* constraint (Fig. 2) can be seen in production lines when two subsequent machines depend on the same resources, which impede them to process at same time. In a particular case of release when completion blocking, defined as RCb, a machine will be immediately available to process its next operation after the same job processed in it is finished and it has left the subsequent machine (see Fig. 3). Examples of RCb and RCb* constraints can be seen in aeronautics parts fabrication and waste treatment industry, described by Martinez et al. (2006) and in cider brewing operations, presented by Trabelsi et al. (2012).

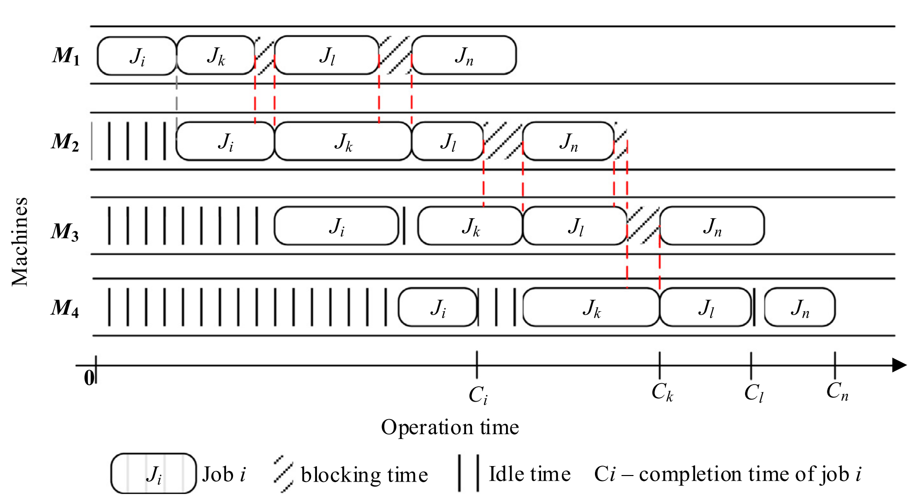

Fig. 1. 4-machine flow shop with RSb constraint.

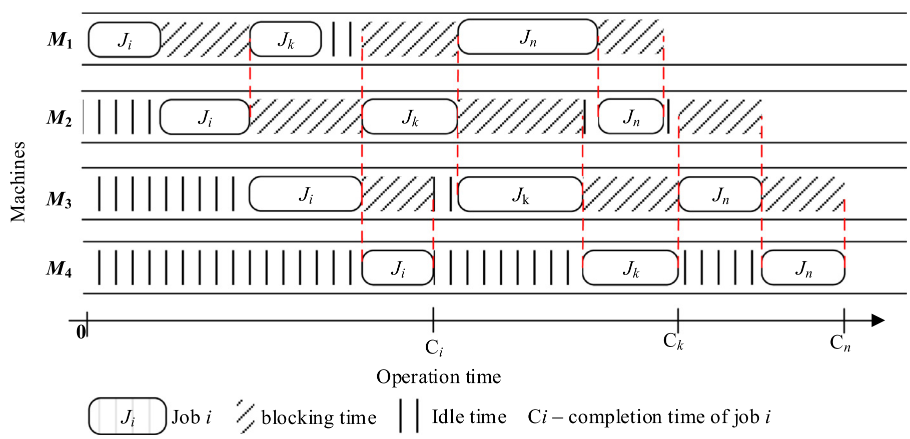

Fig. 2. 4-machine flow shop with RCb* constraint.

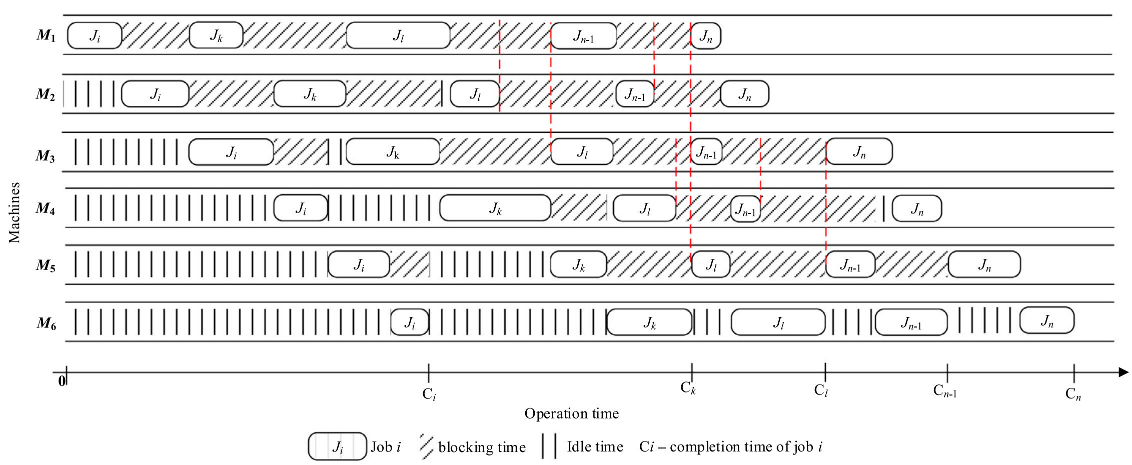

Fig. 3. 4-machine flow shop with RCb constraint.

Departure time of jobs can be computed using the recursive equations presented by Pinedo (1995). Let $D_{i,j}$ denote the departure time of job i on machine j, i.e., the time that job i leaves machine j and is allowed to be processed on machine $j+1$. Departure times of each job on each machine can be computed as follows:

$$D_{1,j} = \sum_{k=1}^{j} p_{1,j}, \qquad j = 1, \ldots, m-1 \tag{1}$$

$$D_{i,0} = D_{i-1,1}, \qquad i = 2, \ldots, n \tag{2}$$

$$D_{i,j} = max\left(D_{i,j-1} + p_{i,j}, D_{i-1,j+1}\right), \qquad j = 1, \ldots, m-1,\ i = 2, \ldots, n \tag{3}$$

$$D_{i,m} = D_{i,m-1} + p_{i,m}, \qquad i = 1, \ldots, n \tag{4}$$

In *RCb* and *RCb** constraints, departure times from the second until the last job are determined by:

$$D_{i,0} = D_{i-1,2}, \qquad i = 2, \ldots, n \tag{5}$$

$$D_{i,j} = max\left(D_{i,j-1} + p_{i,j}, D_{i-1,j+2}\right), \qquad j = 1, \ldots, m-2,\ i = 2, \ldots, n \tag{6}$$

$$D_{i,m-1} = max(D_{i-1,m}, D_{i,m-2}) + p_{i,m-1} \quad i = 2, \ldots, n \tag{7}$$

Thus, departure times of each job on each machine for m-machine flow shop scheduling problem with RCb* or RCb constraints are computed through Eqs. (1) and (4–7).

The representation of the problem followed the $\alpha|\beta|\gamma$ field notation proposed by Graham, et al. (1979). Field $\alpha = \alpha_1\alpha_2$ specifies the technological processing environment. In this paper, field α_1 = F represents the flow shop configuration and α_2 = 2, …, m determines the number of machines. Additionally, Soukhal et al. (2005) and Yuan et al. (2007) used the term $F_m \to D$ to represent m-machine flow shop scheduling problem (F_m) with material handling system (D), which means that jobs are processed first on m-machine flow shop, transferred from the processing facility and delivered to customers by a truck.

Field $\beta = \{\beta_1, \ldots, \beta_n\}$ defines the additional constraints of the problem. In addition to *block* (*RSb*), *RCb* and *RCb** constraints, the following terms will also appear throughout the text:

- Limited buffers ($b_{j,j+1}$): intermediate storage between machines has a limited space, i.e., buffer $b_{j,j+1}$ is allowed to store up to r jobs.
- Sequence-dependent setup times (st_{sd}): setup time to prepare machine k to processing job j depends of job i, i.e., each machine j has a different setup time value assigned for each pair of jobs i and k.
- Anticipatory setup times (st_{ant}): Setup may be performed on machine j before job i arrives on j. In the particular case found, Logendran and Sriskandarajah (1993) defined that setup can only be anticipated if machine j is not blocked by job i–1.

- Non-anticipatory shared setup time (*non-sharedst*): it occurs when jobs grouped by batches share the same configurations of machine setup and, consequently, setup is performed only once for a batch of jobs. In Chen et al. (2014), the technological constraints of the process impose that the shared setup time is not allowable to be anticipated.
- No-wait jobs (*no-wait*): job i must be processed continuously throughout the machines. An idle time in the first machine must ensures that job i is processed continuously from the first until the last machine (Pinedo, 1995).
- Release dates (r_i): as defined by Pinedo (1995), release dates defines the start time that the jobs are allowed to be processed.
- Batching machine ($batch(j)$): machine j can process a set of jobs in batch (Chen et al., 2014).
- Multi-task flexibility ($mtflx(k)$): It implies that a machine k is capable to carry out the relating operations of machine p (Khorasanian & Moslehi, 2017).
- Preemption (*prmp*): operation of job i on machine j can be interrupted. After the machine return to operation, the processing of job i can be resumed with or without additional costs by the interruption (Pinedo, 1995).
- Number of trucks (v), capacity of truck e (c_e) and transportation times (F_i): (Soukhal et al. 2005) integrated the delivery problem with the blocking flow shop scheduling problem. As a constraint to deliver the finished jobs, the authors denoted the number of identical trucks used to delivery finished jobs as $v,\ where\ e = 1, \ldots, v$. Each truck e has a capacity c_e, which means that truck e can transport up to c jobs in one shipment. The elapsed time to transport job i from the output system to the costumer and to come back is defined as transportation time.
- Deteriorating jobs (αp_{ij}): According to Lee et al. (2010), αp_{ij} means that processing times of jobs increase as they are processed later. A factor multiplier α represents the deterioration rate that affects the processing time of job i on machine j.
- Permutation flow shop (*prmu*): Permutation means that the processing of jobs follows the rule First-In-First-Out (FIFO) in all machines (Pinedo, 1995).
- Lot-streaming (*lot-str*): a job or a lot of jobs can be divided into smaller sub-lots to move the completed sub-lots at one machine to downstream machines so that successive operations can be overlapped (Ventura & Yoon, 2013).
- Ordered shop (ord): As highlighted by Koulamas and Panwalkar (2018), the following conditions hold in an ordered shop: if $p_{ij} < p_{kj}$ for any two jobs i and k on machine j, then $p_{ij} \leq p_{kj}$ on all machines $j = 1, \ldots, m$ (ordered jobs). if $p_{ij} < p_{ig}$ for any two machines j and g then $p_{ij} \leq p_{ig}$ on all jobs $i = 1, \ldots, n$ (ordered machines).
- Controllable processing times ($p_{i,j}^{min} \leq p_{i,j} \leq p_{i,j}^{max}$): Al-Salem and Kharbeche (2016) considered that $p_{i,j}$ is assumed to be a decreasing linear function of a given acceleration cost $c_{i,j}$, that is allocated to the processing of $O_{i,j}$. Thus, the data associated with each operation includes three parameters: a non-compressed processing time $p_{i,j}^{max}$; a compressed processing time $p_{i,j}^{min}$ and a compression rate $\alpha_{i,j}$ that determines the processing time according to $c_{i,j}$.
- Machines breakdown (brkdwn) – machine failure can occur at any moment of its operational time. All the machines are subject to failure and a repair time is required to bring the machine to be operational again. Han et al. (2019) adopted an uniform distribution function to simulate machine breakdown an repair time given the probability of machine breakdown is the same at any time.
- Transfer times (t_i): Carlier et al. (2010) adopts this constraints as the elapsed time that a robot takes to transfer job i from the current machine to the subsequent machine.

Term γ represents the objective-function of the problem. The following terms are used represent the measures found:

- Makespan (C_{max})/Expected Makespan ($E(C_{max})$): Corresponds to the completion time of the last job that left the system, i.e., $C_{max} = C_{n,m}$. As highlighted by Pinedo (1995), in blocking flow shop, makespan corresponds to the departure time of nth job on the mth machine ($D_{n,m}$).
- Total flow time ($\sum_i C_i$)/Expected Total flow time ($E(\sum_i C_i)$) or mean completion time ($\sum_i C_i/n$) or weighted mean completion time ($\sum_i W_i C_i/n$): Sum of the completion times of all jobs in the last machine. In the case of mean completion time, total flow time is divided by the number of jobs n. As makespan, completion times of job i on machine m is determined by its departure time $D_{i,m}$.
- Total tardiness ($\sum_i T_i$) and Maximum Tardiness (T_{max})/Expected Tardiness ($E(T_{max})$): In accordance with Ronconi and Henriques (2009), tardiness of job i is determined by $T_i = (0,\ D_{i,m} - d_i)$, where d_i is the due date assigned to job i. Total tardiness corresponds to the sum of tardiness of all jobs. Maximum tardiness is defined as $T_{max} = max_i(T_i)$.
- Cycle time (CC): According to McCormick et al. (1989), CC is given by $lim_{k\to\infty} \frac{b_{1,1}^k - b_{1,1}^1}{k-1}$ where $b_{i,j}^k$ is the starting time of job i at station j in minimum part set or product mix k.
- Customer responsiveness (CR): Cohn et al. (2010) determined CR as providing customers with the right product in the right amount at the right time in a cost-effective way.
- Agreement Index (AI): Yang and Liu (2018) used AI to measure the degree of compliance between completion time of job i C_i and its due date d_i.
- Expected total absolute differences in waiting times ($E(\sum_{i=1}^n \sum_{j=1}^n |w_{[i]}^2 - w_{[j]}^2|)$) – Indicates the differences in waiting time between the completion time of job i and start time of job k on the downstream machine (Jia, 1998).
- Energy consumption (E_W): Generally, the total energy consumption in a flow shop is composed by energy consumption for processing setup, transportation, machine idle, processing of jobs, and public use (Wang et al., 2018).
- Total sum of idle or blocking times ($\sum_i (I_i)\ or\ \sum_i (B_i)$): Idle time of job i (I_i) occurs when machine j must wait job i finish its processing on machine j–1. Blocking time (B_i) occurs according to conditions $RSb,\ RCb\ or\ RCb^*$. The differences between idle time and blocking time for each blocking constraint can be seen in Figs. 1, 2 and 3.

3. Classification of the literature according to the objective-function

In this section, terms and notations described above are used to describe the notation of the problems found. The search process taken into account papers that considered the m-machine blocking flow shop scheduling problems, i.e., the intermediate storages or buffers equal to zero. The review also covered the m-machine flow shop with limited buffers, where the analysis of the problem covered buffers equal to zero ($b = 0$). The presented review considered only papers written in English language. Flexible or hybrid blocking flow shop and distributed flow shop were not considered in this review. The papers search process ranged from the first publication found until the most recent so far (May 2019). ISI web of knowledge and Scopus were chosen as databases. Table 1 shows the keywords, total results found for each set of keywords in

Table 1
Search results in databases.

Data base	Keywords	Results found	Filtered results
ISI web of knowledge	"blocking flow shop" or "flowshop with blocking" or "blocking flowshop" or "flow shop with blocking"	120	90
	"flow shop with limited buffers" or "flowshop with limited buffers" or "flow shop with no intermediate storage" or "flow shop with no intermediate storage" or "flow shop with finite buffers" or "flowshop with finite buffers" or "flow shop with finite intermediate storage"	60	14
	"flow shop with limited storage" or "flowshop with limited storage"	5	2
	"assembly line with blocking"	1	0
Scopus	"flow shop with blocking" or "flowshop with blocking" or "blocking flow shop" or "blocking flowshop" and not "flexible flow shop" and not "parallel flow shop"	221	27
	"flow shop with limited buffers" or "flowshop with limited buffers" or "flow shop with no intermediate storage" or "flowshop with no intermediate storage" or "flow shop with finite buffers" or "flowshop with finite buffers" and not "flexible flow shop" and not "flexible flowshop" and not "hybrid flow shop" and not "hybrid flowshop" and not "distributed flowshop" and not "distributed flowshop"	64	2
	"flow shop with limited storage" or "flowshop with limited storage" and not "hybrid flowshop" and not "hybrid flow shop" and not "flexible flowshop" and not "flexible flow shop"	40	3
	"assembly line with blocking"	3	1
		Total	**139**

each database and the filtered results, respectively. First, keywords shown in Table 1 were used to conduct the search on ISI web of knowledge. With the filtered results in hand, all the repeated references found on Scopus were excluded. From the total generated, 139 papers were filtered. Section 3 is structured according to the group of similar measures, as in Rossit, Tohmé, and Frutos (2018): 3.1) completion time-based objective that included makespan (C_{max}) and total completion time-based costs; 3.2) due date-based objective; 3.3) cycle time; 3.4) multi-objective functions and; 3.5) stochastic time-based costs. Tables 2–7 followed the models proposed by Ruiz and Vázquez-rodríguez (2010) and Rossit et al. (2018), where the authors dispose the publications according to the notation of the problem and comments, in a temporal basis. The majority of the references aimed to propose solution methods for the problem, i.e., exact methods and heuristic methods. Exact methods are divided into mixed integer linear programing (MILP), branch-and-bound and bounded dynamic programming. Heuristic methods are divided into two groups: a) constructive heuristics, which builds the solution in a recursive manner until all jobs are scheduled and; b) metaheuristics and local searches, which uses other mechanisms of search from an initial solution already provided. A third percentage of papers considered the analysis of the problem and literature reviews. With the exception of Hall and Sriskandarajah (1996) and Emmons and Vairaktarakis (2013), papers of the last percentage are included in the respective subsections concerning the adopted objective-function.

3.1. Completion time based measures

3.1.1. Makespan (C_{max})

The importance of makespan (C_{max}) as a performance measure for flow shop scheduling problem with blocking conditions is reflected in the number of publications that adopted it. From the 139 papers covered, minimization of C_{max} corresponds 62% of them. The objective of the papers included the proposition of solution methods and complexity analysis of several types of blocking occurrences. Table 2 shows a summarization of the covered papers that adopted C_{max}. Given the diversity of problems found, this section is subdivided according to the adopted additional constraints. Thus, the next subsections are named as $F_m|block|C_{max}$, $F_m|block,\beta_i|C_{max}$ $F_m|$ RCB,block or/and RCb*, $\beta_i|C_{max}$ and $F_m|b_{j,j+1},\beta_i|C_{max}$.

3.1.1.1. $F_m|block|C_{max}$. Levner (1969) proposed a branch-and-bound (B&B) for the $F_m|block|C_{max}$. The bounds on the optimal value are derived from blocking properties. Reddi and Ramamoorthy (1972) suggested that $F_2|block|C_{max}$ problem is reducible to a special case of the Travelling Salesman Problem. The authors proposed the application of Gilmore and Gomory algorithm (Gilmore & Gomory, 1964), which solves the problem in polynomial time. Suhami and Mah (1981) proposed a Branch-and-bound (B&B) for the $F_m|block|C_{max}$ problem. The authors developed a lower bound based on estimates of the mean and the variance of C_{max} of the fixed partial sequences. Numerical example of the proposed B&B and some computational results are shown. A version of Genetic Algorithm (GA) is developed by Caraffa et al. (2001) for the $F_m|block|C_{max}$ problem. GA tries to mimic the process of natural selection and starts from an initial population, where each individual represents a permutation. At each generation, the fitness value of every individual is evaluated. Multiple individuals are chosen by a selection operator and the picked individuals are modified using crossover and mutation operators to form a new improved population. The algorithm terminates when the stop criterion is reached. The GA designed by the authors generates the population at random and uses Roulette Wheel Parent operator in selection phase, two-point crossover and shift change operators in crossover and mutation phases, respectively. The authors showed that GA outperformed GENIUS heuristic in quality of solution and CPU time. Ronconi (2004) proposed MinMax heuristic (MM) for the $F_m|block|C_{max}$ problem. In MM, the first and last jobs are chosen according to the shortest processing time on the first and the last machines, respectively. From the second position on, the algorithm considers the sum of maximum values between the processing times of consecutive machines and the total processing times of the job. NEH insertion scheme are applied to the solutions generated by MM and Profile Fitting (PF) heuristics in order to improve the results of the proposed methods, creating MME and PFE. NEH is a constructive heuristic proposed by Nawaz, Enscore Jr, and Ham (1983) for the $F_m|prmu|C_{max}$. NEH is composed by two phases. First, jobs are sorted in non-increasing order of their total processing times, generating an initial list. In the second phase, the first two jobs of the initial list are removed and inserted in the partial sequence S. The partial sequence between the two possible combinations of the two jobs that generates the minimum C_{max} is chosen to the next iteration. The next job of the initial list is inserted on all possible positions of the partial sequence. The partial sequence that generates the minimum C_{max} is chosen. Thus, the next job of the initial list is inserted on all possible positions of the partial sequence. The sequence with minimum C_{max} is chosen for the next iteration. The algorithm repeats the process until all jobs from initial list are scheduled. Throughout the paper, second

Table 2
Summarization of papers that adopted C_{max} mono-objective function.

References	Problem	Comments
(Levner, 1969)	$F_m\|block\|C_{max}$	B&B
(Reddi & Ramamoorthy, 1972)	$F_2\|block\|C_{max}$	The problem can be reduced to TSP. The problem can be solved in polynomial time by Gilmore and Gomory Algorithm in polynomial time.
(Dutta & Cunningham, 1975)	$F_2\|b_{j,j+1}\|C_{max}$	Dynamic programming
(Papadimitriou & Kanellakis, 1978)	$F_2\|b_{j,j+1}\|C_{max}$	The problem is NP-complete
(Suhami & Mah, 1981)	$F_m\|block\|C_{max}$	B&B
(Leinsten, 1990)	$F_m\|b_{j,j+1}\|C_{max}$	Constructive heuristics
(Logendran & Sriskandarajah, 1993)	$F_2\|block, st_{ant}\|C_{max}$	The problem is NP-hard in strong sense. The proposed heuristic uses Gilmore and Gomory Algorithm to solve the problem.
(Levner et al., 1995)	$F_2\|block, st_{ant}\|C_{max}$	The two-machine blocking flow shop robotic cell with one robot and setup time to minimize makespan can be solved in polynomial time by Gilmore and Gomory Algorithm.
(Norman, 1999)	$F_m\|b_{j,j+1}, st_{sd}\|C_{max}$	TS
(Nowicki, 1999)	$F_m\|b_{j,j+1}\|C_{max}$	TS
(Grabowski & Pempera, 2000)	$F_m\|prmu, no\text{-}wait, block\|C_{max}$	TS
(Caraffa et al., 2001)	$F_m\|block\|C_{max}$	GA
(Ronconi, 2004)	$F_m\|block\|C_{max}$	Constructive heuristic
(Li & Tang, 2005)	$F_m\|b_{j,j+1}\|C_{max}$	TS
(Ronconi, 2005)	$F_m\|block\|C_{max}$	B&B
(Soukhal et al. 2005)	$F_2 \to D\|block, v, c_e, F_i\|C_{max}$	$F_2 \to D\|block, v=1, c_e=2\|C_{max}$, $F_2 \to D\|block, v=1, c_e=2, F_i=h\|C_{max}$ and $F_2 \to D\|block, v=1, c_e=3\|C_{max}$ problems are strongly NP-hard.
(Martinez et al., 2006)	$F_m\|block, RCb\|C_{max}$	$F_2\|RCb\|C_{max}$ problem is equivalent to the $1\|\|C_{max}$. $F_3\|RCb\|C_{max}$ is a polynomial problem. $F_4\|RCb(1,2), block(2,3), RCb(3,4)\|C_{max}$, $F_3\|RCb(1,2), block(2,3)\|C_{max}$ and $F_3\|block(1,2), RCb(2,3)\|C_{max}$ are polynomial problems. $F_5\|RCb\|C_{max}$, $F_4\|RCb(1,2), RCb(2,3), block(3,4)\|C_{max}$, $F_4\|block(1,2), RCb(2,3), RCb(3,4)\|C_{max}$ and $F_4\|block(1,2), RCb(2,3), block(3,4)\|C_{max}$ are NP-hard problems.
(Wang et al., 2006)	$F_m\|b_{j,j+1}\|C_{max}$	GA
(Companys & Mateo, 2007)	$F_m\|block\|C_{max}$	B&B
(Grabowski & Pempera, 2007)	$F_m\|block\|C_{max}$	TS
(Yuan et al., 2007)	$F_2 \to D\|block, v, c_e, F_i\|C_{max}$	$F_2 \to D\|block, v=1, c_e=2\|C_{max}$ is binary NP-hard $F_2 \to D\|block, v=1, c_e=3, F_i=h\|C_{max}$ is strongly NP-hard
(Liu et al., 2008)	$F_m\|b_{j,j+1}\|C_{max}$	PSO
(Pitty & Karimi, 2008)	$F_m\|block\|C_{max}$ and $F_m\|block, st_{sd}\|C_{max}$	MILP
(Jarboui et al., 2009)	$F_m\|block\|C_{max}$	EDA
(Liu & Kozan, 2009)	$F_m\|prmu, no\text{-}wait, block, b_{j,j+1}\|C_{max}$	Constructive heuristic
(Qian et al. 2009)	$F_m\|b_{j,j+1}\|C_{max}$	DE
(Carlier et al., 2010)	$F_m\|block, t_j\|C_{max}$	GA
(Companys et al., 2010a)	$F_m\|block\|C_{max}$	Constructive heuristic
(Companys et al., 2010b)	$F_m\|block\|C_{max}$	SA
(Duan et al., 2010)	$F_m\|block\|C_{max}$	HS
(Gong et al., 2010)	$F_2\|no\text{-}wait, block, batch, sharedst\|C_{max}$	Constructive heuristic
(Lee et al., 2010)	$F_2\|block, \alpha p_{ij}\|C_{max}$	B&B
(Trabelsi et al., 2011)	$F_m\| RCB, block, RCb^*\|C_{max}$	$F_4\|RCb(1,2), block(2,3), RCb(3,4)\|C_{max}$ and $F_3\|RCb^*\|C_{max}$ are polynomial problems. MILP
(Wang et al., 2010)	$F_m\|block\|C_{max}$	DE
(Liang et al., 2011)	$F_m\|block\|C_{max}$	PSO
(Pan et al., 2011)	$F_m\|b_{j,j+1}\|C_{max}$	HS
(Pan et al., 2011)	$F_m\|b_{j,j+1}\|C_{max}$	DE
(Ribas et al., 2011)	$F_m\|block\|C_{max}$	IG
(Tang & Tang, 2011)	$F_2\|b_{j,j+1}, batch, st_{si}\|C_{max}$	PSO
(Wang et al., 2011)	$F_m\|block\|C_{max}$	HS
(Bao et al., 2012)	$F_m\|block\|C_{max}$	HS
(Bautista et al., 2012)	$F_m\|block\|C_{max}$	Bounded Dynamic Programming
(Davendra et al., 2012)	$F_m\|block\|C_{max}$	DE
(Han et al., 2012)	$F_m\|block\|C_{max}$	ABC
(Maleki-Darounkolaei et al., 2012)	$3AF\|block, st_{sd}\|\alpha(\sum_i W_i C_i/W)+(\alpha-1)C_{max}$	SA
(Pan & Wang, 2012)	$F_m\|block\|C_{max}$	Constructive heuristics
(Sauer & Sauvey, 2012)	$F_m\|RCb\|C_{max}$	GA
(Trabelsi et al., 2012)	$F_m\| prmu, block, RSb, RCb^*, RCb\|C_{max}$	GA
(Wang et al., 2012)	$F_m\|block\|C_{max}$	SA

(continued on next page)

Table 2 (*continued*)

References	Problem	Comments
(Wang & Tang, 2012)	$F_m\|block\|C_{max}$	PSO
(Chowdhury et al., 2013)	$F_m\|block\|C_{max}$	GA
(Davendra & Bialic-Davendra, 2013)	$F_m\|block\|C_{max}$	SOMA
(Davendra et al., 2013)	$F_m\|block\|C_{max}$	SOMA
(Lin & Ying, 2013)	$F_m\|block\|C_{max}$	AIS
(Pan et al., 2013)	$F_m\|block\|C_{max}$	MMA
(Ribas & Companys, 2013)	$F_m\|block\|C_{max}$	VNS
(Chen et al., 2014)	$F_2\|block,\ batch,\ \mathrm{r_{i,j}}\|C_{max}$	DE
(Davendra et al., 2014)	$F_m\|block\|C_{max}$	SSA
(Moslehi & Khorasanian, 2014)	$F_m\|b_{j,j+1}\|C_{max}$	VNS
(Zhao et al., 2014)	$F_m\|b_{j,j+1}\|C_{max}$	PSO
(Ding et al., 2015)	$F_m\|block\|C_{max}$	IG
(Han et al., 2015)	$F_m\|block\|C_{max}$	ABC
(Khorramizadeh & Riahi, 2015)	$F_m\|prmu,block,\ RSb,\ RCb^*,\ RCb\|C_{max}$	ABC
(Liu, Li, & Ren, 2015)	$F_m\|block\|C_{max}$	VNS-SSA
(Sadaqa & Moraga, 2015)	$F_m\|block\|C_{max}$	Meta-RaPS
(Tasgetiren et al., 2015)	$F_m\|block\|C_{max}$	DE
(Wang et al., 2015)	$F_m\|block\|C_{max}$	CuckooSA
(Zhang et al., 2015)	$F_m\|block\|C_{max}$	VNS-SA
(Al-Salem & Kharbeche, 2016)	$F_2\|batch(1), block,\ p_{i,j}^{min} \le p_{i,j} \le p_{i,j}^{max}\|C_{max}$	GA
(Eddaly et al., 2016)	$F_m\|block\|C_{max}$	PSO
(Han et al., 2016)	$F_m\|block\|C_{max}$	FFOA
(Fu et al., 2017)	$F_m\|block,\ \alpha p_{ij}\|C_{max}$	CRO-VNS
(Khorasanian & Moslehi, 2017)	$F_2\|block,mtflx,\ prmp\|C_{max}$	VNS
(Riahi et al., 2017)	$F_m\|prmu,block,\ RSb,\ RCb^*,\ RCb\|C_{max}$	SSA
(Takano & Nagano, 2017)	$F_m\|block,\ st_{sd}\|C_{max}$	B&B
(Tasgetiren et al., 2017)	$F_m\|block\|C_{max}$	IG
(Toumi et al., 2017)	$F_m\|block\|C_{max}$	B&B
(Koulamas & Panwalkar, 2018)	$F_m\|block,ord\|C_{max}$	Index rule
(Liu, Wang, & Zhang, 2018)	$F_m\|block\|C_{max}$	IBBO
(Panwalkar & Koulamas, 2018)	$F_m\|block\|C_{max}$	New flow shop schematic
(Riahi et al., 2018)	$F_m\|\mathrm{st_{sd}},RCb,\ RSb\|C_{max}$	CGLS
(Shao, Pi, & Shao, 2018)	$F_m\|block,\ st_{sd}\|C_{max}$	DWWO
(Shao, Pi, & Shao, 2018)	$F_m\|block\|C_{max}$	EDA-VNS
(Ozolins, 2019)	$F_m\|block\|C_{max}$	Bounded Dynamic Programming
(Takano & Nagano, 2019)	$F_m\|block,\ st_{sd}\|C_{max}$	Constructive heuristics

References to the abbreviations: B&B – branch-and-bound; TS – Tabu Search; GA – Genetic Algorithm; PSO – Particle Swarm Optimization; DE - Differential Evolution; GRASP - Greedy Randomized Adaptive Search Procedure; HS – Harmony Search; IG – Iterated Greedy; ABC – Artificial Bee Colony; SA – Simulated Annealing; SOMA - Self-organising Migrating Algorithm; AIS – Artificial Immune System; MMA – Memetic Algorithm; ILS – Iterated Local Search; VNS – Variable Search Neighborhood; CuckooSA – Cuckoo Search Algorithm; MDGSO - Multi-objective discrete group search optimizer; FFOA – Fruit Fly Optimization Algorithm; VBIH - Variable Block Insertion Heuristic; CRO-VNS – Chemical Reaction Optimization with VNS; CSA – Clustering Search Algorithm; SSA – Scatter Search Algorithm; IBBO - Improved Biogeography-Based Optimization Algorithm; CGLS – Constraint Guided Local Search; DWWO – Discrete Water Wave Optimization; EDA-VNS – Estimation Distribution Algorithm with VNS.

Table 3
Summarization of papers that adopted total flow time-based mono-objective function.

References	Problem	Comments
(Weng, 2000)	$F_m\|block\|\sum_i C_i/n$	TS
(Wang et al., 2010)	$F_m\|block\|\sum_i C_i$	HS
(Deng et al., 2012)	$F_m\|block\|\sum_i C_i$	ABC
(Khorasanian & Moslehi, 2012)	$F_m\|block\|\sum_i C_i$	IG
(Han et al., 2013)	$F_m\|block\|\sum_i C_i$	ABC
(Li et al., 2013)	$F_m\|block\|\sum_i C_i$	Heuristic dynamic programming
(Moslehi & Khorasanian, 2013)	$F_m\|block\|\sum_i C_i$	B&B
(Toumi et al., 2013d)	$F_m\|block\|\sum_i C_i$	B&B
(Toumi et al., 2013c)	$F_m\|block\|\sum_i C_i$	B&B
(Vo et al., 2015)	$F_m\|permu,\beta_i\|\sum_i C_i$	B&B
(Ribas et al., 2015)	$F_m\|block\|\sum_i C_i$	ABC
(Ribas & Companys, 2015)	$F_m\|block\|\sum_i C_i$	GRASP
(Fernandez-viagas et al., 2016)	$F_m\|block\|\sum_i C_i$	Constructive heuristics
(Tasgetiren et al., 2016)	$F_m\|block\|\sum_i C_i$	VBIH
(Ren et al., 2018)	$F_m\|block,r_i\|\sum_i C_i^3$	GA, TS and SA

References to the abbreviations: B&B – branch-and-bound; TS – Tabu Search; GA – Genetic Algorithm; GRASP - Greedy Randomized Adaptive Search Procedure; HS – Harmony Search; IG – Iterated Greedy; ABC – Artificial Bee Colony; VNS – Variable Search Neighborhood; VBIH - Variable Block Insertion Heuristic.

Table 4
Summarization of papers that adopted due dates-based mono-objective function.

References	Problem	Comments
(Armentano & Ronconi, 2000)	$F_m\|block\|\sum_i T_i$	Constructive heuristic and TS
(Ronconi & Armentano, 2001)	$F_m\|block\|\sum_i T_i$	B&B
(Januario, Arroyo, & Moreira, 2009)	$F_m\|block\|\sum_i T_i$	GA
(Ronconi & Henriques, 2009)	$F_m\|block\|\sum_i T_i$	GRASP
(Ribas et al., 2013)	$F_m\|block\|\sum_i T_i$	ILS
(Toumi et al., 2013a)	$F_m\|block\|\sum_i T_i$	B&B
(Toumi et al., 2013b)	$F_m\|block\|\sum_i T_i$	B&B
(Nouha & Talel, 2016)	$F_m\|block\|\sum_i T_i$	PSO
(Nagano et al., 2017)	$F_m\|block\|\sum_i T_i$	CSA
(Shao et al., 2017)	$F_m\|block\|\sum_i T_i$	IWO

References to the abbreviations: B&B – branch-and-bound; TS – Tabu Search; GA – Genetic Algorithm; PSO – Particle Swarm Optimization; Differential Evolution; GRASP - Greedy Randomized Adaptive Search Procedure; ILS – Iterated Local Search; CSA – Clustering Search Algorithm; IWO – Invasive Weed Optimization.

Table 5
Summarization of papers that adopted cycle time mono-objective function.

References	Problem	Comments
(McCormick et al., 1987)	$F_m\|block\|CC$	The problem is proved to NP-complete
(McCormick et al., 1989)	$F_m\|block\|CC$	Constructive heuristic PF is proposed
(Kamoun & Sriskandarajah, 1993)	$F_m\|block\|CC$	$F_3\|no\text{-}wait(1,2),\ block(2,3)\|C_t$ and $F_3\|block\|C_t$ problems are NP-hard in strong sense
(Abadi et al., 2000)	$F_m\|block\|CC$	Constructive heuristic
(Abadi, 2007)	$F_m\|block\|CC$	Constructive heuristic

Table 6
Summarization of papers that adopted multi-objective functions.

References	Problem	Comments
(Prasad et al., 2006)	$F_m\|b_{j,j+1}\|\ \sum_i \bar{C}_{icont} + \sum_i \bar{C}_{iparts} + \sum_i \sigma_i$	GA
(Qian et al., 2009)	$F_m\|b_{j,j+1}\|C_{max}$ and T_{max} $F_m\|b_{j,j+1}\|C_{max}$ and $\sum_j I_j$	DE
(Cohn et al., 2010)	$F_m\|block\|C_{max}$ and CR	Trade-offs between C_{max} and CR can change the solution characteristics substantially
(Emmons & Vairaktarakis, 2013)	General problems with blocking	Literature review on no-wait and blocking machine scheduling problems 2012
(Ventura & Yoon, 2013)	$F_m\|b_{j,j+1}, lot-str\|\sum_i (T_i + E_i)$	GA
(Nouri & Ladhari, 2015)	$F_m\|block\|C_{max}$, $F_m\|block\|\sum_i C_i$ and $F_m\|block\|\sum_i T_i$	GA and ABC
(Guanlong et al., 2016)	$F_m\|block\|C_{max} + \sum_i C_i$	MDGSO
(Nouri & Ladhari, 2017)	$F_m\|block\|C_{max} + \sum_i C_i$	GA
(Lebbar et al., 2018)	$F_m\|block,r_i\|\alpha C_{max} + (\alpha\text{-}1)T_{max}$	MILP
(Gong, Han, & Sun, 2018)	$F_m\|lot\text{-}str,block\|C_{max} + E_{max}$	ABC
(Lebbar et al., 2018)	$F_m\|block\|C_{max}$ and T_{max}	GA
(Shao, Pi, & Shao, 2018a)	$F_m\|block,r_i\|C_{max} + \sum_i T_i$	IWO
(Wang et al., 2018)	$F_m\|block\|C_{max} + E_W$	VNS
(Yang & Liu, 2018)	$F_m\|block\|\tilde{C}_{max} + \widetilde{AI}$	GWO
(Han et al., 2019)	$F_m\|brkdwn,block,\ lot\text{-}str\|C_{max}$ and T_{max}	REMO

References to the abbreviations: GA – Genetic Algorithm; ABC – Artificial Bee Colony; MDGSO - Multi-objective discrete group search optimizer; IWO – Invasive Weed Optimization; VNS – Variable Neighborhood Search; GWO – Grey Wolf Optimization; REMO - Evolutionary Robust Scheduling Algorithm.

Table 7
Summarization of papers that adopted stochastic objective functions.

References	Problem	Comments
(Pinedo, 1982)	$F_m\|block\|E(C_{max})$	Scheduling rule
(Foley & Suresh, 1984)	$F_2\|block\|E(\sum_i C_i)$	Scheduling rule
(Wie & Pinedo, 1986)	$F_m\|block\|E(C_{max})$ and $F_m\|block\|E(\sum_i C_i)$	Scheduling rules
(Jia, 1998)	$F_2\|block\|E(\sum_{i=1}^{n}\sum_{j=1}^{n} \|w_{[i]}^2 - w_{[j]}^2\|)$	Scheduling rule
(Kalczynski & Kamburowski, 2005)	$F_2\|block\|E(C_{max})$	The problem is proved to be equivalent to TSP on a permuted Monge matrix
(Hu et al., 2008)	$F_m\|b_{j,j+1}\|E(C_{max})$	DE
(Han et al., 2016)	$F_m\|block,lot\text{-}str\|E(C_{max}) + E[var(T_{max})]$	MOEA
(Han et al., 2019)	$F_m\|block, lot-str\|E(\bar{C}_{max}) + E[var(C_{max})]$	MBO

References to the abbreviations: DE -Differential Evolution; MOEA – Multi-objective Evolutionary Algorithm; MBO – Migrating Birds Optimization.

phase of NEH is referred as NEH insertion scheme. Computational results showed that, in average, PF reached better results than MM. Regarding the number of best solutions, PFE and MME presented a superior performance in comparison with NEH. Ronconi (2005) developed a B&B for the $F_m|block|C_{max}$ problem. The author proposed a lower bound (LB_{RON}) on C_{max} that takes into account an estimation of the departure times of the non-scheduled jobs set. Computational results showed that the proposed bounding scheme works better than the lower bound proposed by Ronconi and Armentano (2001). Abadi (2007) reviewed the slowing down operation technique from Abadi et al. (2000) to provide a new and faster algorithm for computing C_{max} of m-machine blocking flow shop problems. Companys and Mateo (2007) proposed the LOMnicki PENdular algorithm (LOMPEN) for the $F_m|prmu|C_{max}$ and $F_m|block|C_{max}$ problems. LOMPEN applies, simultaneously, branch-and-bound algorithm to solve the direct and inverse instance problem. A lower bound similar to developed by Ronconi and Armentano (2001) is adapted and incorporated to LOMPEN to solve $F_m|block|C_{max}$ problem. The authors found new upper bounds for the Taillard's instance problems (up to 200 jobs and 10 machines). Grabowski and Pempera (2007) proposed a TS and TS with multimoves local search (TS + M) for the $F_m|block|C_{max}$ problem. TS starts from an initial solution and searches through its neighborhood for a solution to minimize the objective-function. Then, the best solution becomes the initial solution and the process is continued until a stop criterion is met. A memory of the search history is used to prevent cycling as well as to guide the search for promising solution space regions. This list records, for a certain number of iterations, the performed moves, treating them as being forbidden for future moves. The search stops when a given number of iterations without improvement value has been reached. Both TS and TS+M apply the properties of block and anti-block developed by Grabowski and Pempera (2000). TS+M uses multimoves local search, which applies several potential moves simultaneously to a candidate sequence. Computational results showed that both heuristics outperformed GA by Caraffa et al. (2001) and B&B by Ronconi (2005). Pitty and Karimi (2008) classified MILP flow shop scheduling models from the literature, proposing MILP models for the $F_m|block|C_{max}$ and $F_m|block,st_{sd}|C_{max}$ problems. The authors used pairwise slot assignment (PS) and slot-based sequencing models (SA) approaches to model the considered problems. The authors conducted an extensive computational evaluation with MILPs of the literature. Jarboui et al. (2009) proposed an Estimation Distribution Algorithm (EDA) for the $F_m|block|C_{max}$ problem. EDA is a population based evolutionary algorithm that explores the solutions space by sampling a probabilistic model updated according to the promising best solutions so far. EDA works with an initial population of candidate permutations. The population is scored using a fitness population, which gives a numerical ranking for each solution. Based on the ranked population, a subset of the most promising permutations are selected by a selection operator. Then, EDA constructs a probabilistic model which attempts to estimate the probability distribution of each element of the subset. New solutions are generated by sampling the distribution encoded by this model. The new individuals are inserted into the old population, possibly replacing it entirely. The algorithm repeats these procedures until the stop criterion is met. In the proposed H-EDA, initial population is generated by NEH and random method. As selection criterion, Q solutions are ranked and selected based on the lowest C_{max}. A probabilistic distribution model is developed based on the position of jobs in the sequence. The worst individuals are replaced by the new individuals according to its fitness value. A local search is applied to the new individuals, given a probability value of improvement. Computational results showed that the proposed H-EDA improved the upper bounds generated by the B&B by Ronconi (2005) and TS+M by Grabowski and Pempera (2007). Companys, Ribas, and Mateo (2010a) proposed constructive heuristics based on NEH for the $F_m|block|C_{max}$ problem. The authors applied PF by McCormick et al. (1989), Trapezium by Companys (1966), POUR by Pour (2001) and N&M (Nagano & Moccellin, 2002) heuristics as inputs for NEH enumeration scheme. A new tie breaking mechanism is proposed to be used as a criterion to decision on the second phase. Through computational experimentations, the authors recommended PLE2, which consists in applying PF to the inverse instance, as method to solve the problem. Companys, Ribas, and Mateo (2010b) developed improvement heuristics for the $F_m|block|C_{max}$ problem. The authors used NEH, POUR and N&M heuristics as initial sequence for a variation of non-exhaustive descent algorithm (NEDA) called Soft Simulated Annealing (SSA). NEDA tries to improve a solution by swapping any two positions of the sequence. The authors used an auxiliary vector called revolver, which allows exploring the neighborhood choosing a position randomly. Experimental results showed that the developed methods considerably improved the inputs given by NEH and N&M. Duan et al. (2010) proposed a Hybrid Harmony Search (HHS) for the $F_m|block|C_{max}$ problem. HS is a population-based stochastic algorithm which tries to simulate the natural musical performance process when a musician searches for a better state of harmony. Each permutation is represented by a harmony n-dimensional vector. HS requires an initial population of harmony vectors that are stored in a harmony memory. Then, a new candidate harmony is generated by using memory consideration, pitch adjustment and a random selection. Harmony memory is updated by comparing the new candidate with the worst solution of memory consideration. The process continues until the stop criterion is met. The proposed HHS uses a modified version of NEH and random method to generate the initial population of harmony vectors. In the improvisation phase, a new harmony vector is generated according to the memory consideration or to a random selection (i.e., random re-initialization between search bounds). If it is updated by the memory consideration, a pitch adjustment is applied to the new individual. The old population is updated according to the fitness value of the solutions. An insertion local search is applied to each new harmony vector generated. Experimental results showed that the proposed hHS outperformed GA by Caraffa et al. (2001) and TS and TS+M by Grabowski and Pempera (2007). Wang et al. (2010) proposed some versions of Differential Evolution (DE) for the $F_m|block|C_{max}$ problem. DE is a stochastic algorithm based on floating-point representation that uses the differential information among individuals to conduct the global optimization. DE starts from a target population and at each generation, where each individual represents a permutation. During the iteration, it orderly adopts a mutation operator to generate mutant population, a crossover operator to generate trial individuals and a greedy operator to determine a new target population to the next generation. As DE is originally designed for continuous problems, the authors proposed an adapted version of the algorithm, called Discrete DE (DDE), for discrete optimization problem. The second version, denominated Hybrid Discrete DE (HDDE), incorporates a problem dependent local search, which is performed after the crossover phase with a certain probability. Experimental results showed that the proposed metaheuristics outperformed HDDE by Qian et al. (2009), GA by Caraffa et al. (2001), TS and TS+M by Grabowski and Pempera (2007). Liang et al. (2011) developed a Dynamic multi-swarm Particle Swarm Optimization (DMS-PSO) for the $F_m|block|C_{max}$ problem. The basic principle of PSO is to simulate the social behavior of bird flocking or fishing schooling, as well as the means of information exchange between them. A swarm consists of n individuals, called particles, that flies around in an n-dimensional search space. Each particle represents a solution. Each particle is defined by n-dimensional vectors: a) the position of the ith particle in the search space; b) the velocity with which the particle i moves; c) the best position of particle i and; d) the best global particle of the swarm until tth iteration. The position and velocity of the ith particle is adjusted at each iteration t, which is guided by the best position visited by itself and the best position of the best particle of the swarm. The process continues until a stop criterion is met. In order to ensure a swarm with a certain quality level and diversity, the proposed DMS-PSO uses a variant of NEH to generate the initial individuals. Thus, DMS-PSO divides the swarm in small sized sub-swarms, in order to slow down the population's convergence velocity and increase diversity. A local PSO is applied to each sub-swarm. Every R generations, the sub-swarms are randomly regrouped and the search is restarted using a new configuration of small swarms. In order to alleviate the trade-off between larger diversity and faster convergence velocity, a problem-dependent local search is applied to best individuals of each sub-swarm. Experimental results showed that the proposed metaheuristic outperformed TS and TS + M by Grabowski and Pempera (2007) and HDE by Qian et al. (2009). Ribas, Companys, and Tort-Martorell (2011) developed some versions of Iterated Greedy (IG) for the $F_m|block|C_{max}$ problem. IG algorithm starts from an initial solution and then iterates through a main loop in which a partial candidate solution is obtained by removing d jobs from a complete candidate solution. Next, the sequence is reconstructed starting with the partial candidate solution. An acceptance criterion decides if the new solution becomes the new incumbent solution. The process is repeated until the stop criterion is met. In both proposed IG1 and IG2, NEH heuristic is used to generate the initial solution. During the reconstruction phase, the authors proposed some tie-breaking mechanisms to evaluate partial sequences with same C_{max}. A non-exhaustive descent algorithm (NEDA) is applied as the improvement phase. In order to avoid exploring the neighborhood always in the same order, the authors developed a tool called revolver, which is used to randomly select two positions of the sequence to swap during NEDA application. While IG1 uses the basic structure described previously, IG2 uses revolver tool in the local search phase. Computational results showed that both IG1 and IG2 outperformed HDDE by Wang et al. (2010). Wang et al. (2011) proposed three versions of Harmony Search (HS) for the $F_m|block|C_{max}$ problem. The first proposed version of HS is called standard HS and uses NEH-WPT to generate a solution to compose the initial population. NEH-WPT creates the initial list sorting jobs according to non-decreasing values of total processing time. In the second phase, the initial list is used as input to NEH insertion scheme. The second version of HS, denominated modified global-best HS (mgHS), uses a modified pitch adjustment rule. The third version, called Hybrid mgHS (hmgHS), adds an insertion problem-dependent local search to mgHS. Computational results showed that both proposed algorithms outperformed HGA by Wang et al. (2006), TS + M by Grabowski and Pempera (2007), Hybrid PSO by Liu et al. (2008) and IG by Ribas et al. (2011). Davendra et al. (2012) developed a Differential Evolution, named EDE_C for the $F_m|block|C_{max}$. EDE_C uses the principle of clustering to replace old individuals of the population by the new individuals generated. Besides the standard steps of DE, EDE_C also applied a 2-optimal local search to explore the neighborhood of the candidate solution. Through computational tests, the authors conclude that EDE_C outperformed GA by Caraffa et al. (2001), TS and TS+M by Grabowski and Pempera (2007), HDDE by Qian et al. (2009) and HDDE by Wang et al. (2010). Bao, Zheng, and Jiang (2012) proposed an Improved Harmony Search (IHS) for the $F_m|block|C_{max}$ problem. IHS incorporates GA algorithm to improve the population. An insertion neighborhood local search is also proposed to improve the local exploitation ability. Computational results showed the superiority of the proposed metaheuristic against HS, GA and Hybrid HS+GA by Bao, Jing, and Zheng (2011). Bautista et al. (2012) proposed a bounded dynamic programming (BDP) for the $F_m|block|C_{max}$ problem. The BDP combines features of dynamic programming and B&B. The authors adapted lower bounds proposed by Lageweg, Lenstra, and Kan (1978) as a general partial bounds, considering that $F_m|\text{prmu}|C_{max}$ is a relaxation of $F_m|block|C_{max}$. The proposed BDP was compared with the best results for Taillard's instances problems, reported by Ribas et al. (2011). Han et al. (2012) proposed an Improved Artificial Bee Colony (IABC) for the $F_m|block|C_{max}$. ABC is a swarm intelligence-based algorithm that simulates the foraging behavior of employed bees, onlooker bees and scout bees. ABC requires a population of food sources, where each food source corresponds to a solution. Then, these food sources are updated by three kinds of honey bee. Firstly, an employed bee generates a new food source based on the current one. The onlooker bee modifies the food source shared by employed bee. A new food source is produced to update the unchanged ones using a scout bee. In all phases, the fitness value of the food source is calculated. The developed IABC uses MM, NEH, NEH-WPT, a modified version of MM and random method to generate the initial population. In the employed bee stage, four approaches based on insertion and swap operators are applied to generate new solutions. In the onlooker bee stage, an insertion local search, developed by Ruiz and Stützle (2008), is used to try to improve the solutions. Finally, the DDE algorithm by Wang et al. (2010) is applied in the scout bee stage to generate new solutions. The proposed IABC obtained better results than GA by Caraffa et al. (2001), TS and TS + M by Grabowski and Pempera (2007), HDDE by Wang et al. (2010), and Discrete ABC (DABC) by Han et al. (2011). Pan and Wang (2012) develop some constructive heuristics based on the combination of PF, NEH and Referenced Local Search (RLS) for the $F_m|block|C_{max}$ problem. Both proposed wPF and PW use the same steps of PF, with exception of the evaluation function to select the next job of the partial sequence. In order to improve the quality of solution, NEH-insertion scheme is added to PF, wPF and PW, creating PF-NEH(x), wPF-NEH(x) and PW-NEH(x), where x is the number of iterations that all the heuristics are run. Finally, PF-NEH_{LS}(x), wPF-NEH_{LS}(x) and PW-NEH_{LS}(x) are proposed applying RLS to the solution generated by PF-NEH(x), wPF-NEH(x) and PW-NEH(x), respectively. Experimental results evidenced that PF-NEH_{LS}(x), wPF-NEH_{LS}(x) and PW-NEH_{LS}(x) obtained the best results. Wang and Tang (2012) proposed a Discrete PSO (DPSO) for the $F_m|block|C_{max}$ problem. DPSO uses a self-adaptive perturbation to avoid the premature convergence of the solutions. DPSO also applies SVNS, based on Mladenovic and Hansen (1997), to the best global particle so found. Computational results showed that the proposed DPSO outperformed TS+M by Grabowski and Pempera (2007) and both DPSO algorithms by Rameshkumar and Mohanasundaram (2005) and Ponnambalam et al. (2009). For the $F_m|block|C_{max}$ problem, Wang et al. (2012) proposed the Three Phase Algorithm (TPA). In the first phase of TPA, an initial list is composed by jobs sorted according to non-decreasing order of the weighted sum between the average job processing times and the standard deviation of job processing times. In the second phase, NEH insertion scheme is applied to the initial list generated in phase 1. In the third phase, a modified Simulated Annealing is used to improve the solution given by the second phase. Computational results showed that TPA outperformed HDDE by Wang et al. (2010) and IG1 and IG2 by Ribas et al. (2011) in quality of solution. Chowdhury, Ghosh, and Sinha (2013) proposed a novel GA (NGA) to solve the travelling salesman problem (TS) and the $F_m|block|C_{max}$. NGA applies a forced mutation to the worst sequences of the population. A multilevel mutation mechanism is used to mutation phase, which is composed by more than one operator (swap and inversion). In order to ensure a bidirectional quality flow, NGA uses ring-parent topology to ensure bidirectional quality flow. Fitness function value, the order crossover operator and several mutation operators in selection, crossover and mutation phases, respectively. Experimental results showed that the proposed NGA outperformed branch-and-bound by Ronconi (2005), TS + M by Grabowski and Pempera (2007) and HDDE by Wang et al. (2010). Davendra and Bialic-Davendra (2013) proposed a Discrete Self Organism Migrating Algorithm (DSOMA) algorithm for the $F_m|block|C_{max}$ problem. SOMA is an evolutionary algorithm based on competitive and cooperative behavior of intelligent creatures to solve a common problem. SOMA starts from an initial population where each individual corresponds to a solution. A migration phase is applied, where the individuals are modified by moves towards to the best solution of the population. The moves consists of jumps, determined by a jump parameter until the individual reaches the final position of a path length parameter. In DSOMA, initial population is randomly generated by an insertion method. In order to create new individuals, the parameter of minimum jumps that each individual has circumnavigate is used to generate new individuals. A 2-OPT local search is applied to the best new individual generated. Experimental results showed that DSOMA outperformed the results generated by the B&B by Ronconi (2005), TS + M by Grabowski and Pempera (2007) and HDDE by Wang et al. (2010). A DSOMA with lozi map is developed by Davendra et al. (2013) for solving the $F_m|block|C_{max}$ problem. The lozi map is embedded in phases of DSOMA that require the use of a pseudorandom number generator. Besides the basic structure of the algorithm, the algorithm performs a 2-Opt local search to the best new individual with the minimum fitness value in population update. Experimental results showed that the proposed DSOMA performed better than HDDE by Qian et al. (2009), RON by Ronconi (2005) and TS+M by Grabowski and Pempera (2007). Lin and Ying (2013) proposed a Revised Artificial Immune System (RAIS) with SA and a pure AIS (PAIS) for the $F_m|block|C_{max}$. AIS is stochastic algorithm based on the biological immune system (BIS). In the AIS, each antibody of a population represents a solution. An affinity value is associated with each antibody. According to a pre-defined affinity function, a number of clones generated from each antibody is calculated. The clones are mutated to generate new antibodies. The new antibodies are evaluated again, and the worst individuals of the population is replaced by the better mutated clones. The process is repeated until a termination criterion is met. In the proposed RAIS, antibodies are randomly generated to compose the initial population and a determined number of antibodies is subject to be cloned. All the clones are mutated using swap operator to random positions. After the update of the antibody population, a given number of antibodies are mutated again and the new solutions which generate worst affinity values can replace the original individual under a probability function based on SA mechanism. Computational results showed that both metaheuristics outperformed TS + M by Grabowski and Pempera (2007), the B&B by Ronconi (2005), HDDE by Wang et al. (2010) and IG by Ribas et al. (2011). Pan et al. (2013) developed a Modified Memetic Algorithm (MMA) for the $F_m|block|C_{max}$ problem. MMA is a population-based metaheuristic inspired by the natural selection Darwinian principles and Dwakins' notion of meme, which is defined as unit of cultural evolution that is capable of individual refinements. MA repeats evaluation, selection, crossover, mutation and local refinements, until a given stop criterion is met. As method to generate some individuals to compose the initial population, the authors proposed PF+NEH, which uses PF to scheduling n - λ jobs and NEH to scheduling the remaining λ jobs. In the selection, crossover and mutation phases, MMA uses the tournament selection, path relinking technique and shift operators, respectively. As local refinement, a Referenced Local Search (RLS) is applied to new individuals generated after mutation. Experimental results showed that MMA outperformed HDDE by Wang et al. (2010), IG by Ribas et al. (2011), HGA by Wang et al. (2006), HPSO by Liu et al. (2008), DPSO by Wang and Tang (2012) and GAs by Tseng and Lin (2010) and Tseng and Lin (2010). Ribas and Companys (2013) proposed two Variable Neigborhood Search (VNS) for the $F_m|block|C_{max}$. The main strategy of VNS is the systematic change in neighbourhood structure within a local search in order to scape from a local optimum. VNS starts from an initial solution, and, while the termination criterion is not met, local searches are applied to the solution. Both proposed Paralell VNS (PVNS) and series structures VNS (SVNS) use the same initial solution, selection procedure and perturbation mechanism, but they differ in the improvement phase. PW/PW2 by Pan and Wang (2012), applied respectively to directed and reversed instances, are used to generate the initial solution. The authors adopted the destruction/construction procedures proposed by Ruiz and Stützle (2007) as perturbation mechanism. In PVNS, one of the local searches are applied given a percentage of selection. In SVNS both local searches are applied in series. Experimental results showed that both PVNS and SVNS outperformed HDDE by Wang et al. (2010) and the IGs by Ribas et al. (2011). Davendra et al. (2014) applied a scatter search algorithm (SSA) for the $F_m|block|C_{max}$ problem. SSA is an evolutionary metaheuristic that first creates a reference set R of solutions and then, intelligently combines solutions with each other, in order to reach better solutions. At each iteration, the combined solutions are subject to improvement methods and the set R is updated. The process continues until a stop criterion is satisfied. The authors incorporated two-point crossover operator to combine different solutions and included a 2-OPT local search as improvement method. SSA is driven by a set of chaos maps, which are used in phases of SSA that require a pseudorandom number generator. Experimental results evidenced that the proposed SSA outperformed HDDE by Qian et al. (2009), GA by Caraffa et al. (2001) and TS+M by Grabowski and Pempera (2007). Ding et al. (2015) proposed an IG with blocking properties (B-IG) for the $F_m|block|C_{max}$ problem. Based on Grabowski and Pempera (2007), the authors developed some new properties to eliminate "useless moves" of a block of jobs. The proposed B-IG uses a modified NEH as initial solution and applies the proposed block properties during reconstruction phase of IG. Experimental results evidenced that B-IG outperformed HDDE by Wang et al. (2010), IG by Ribas et al. (2011), TPA by Wang et al. (2012), RAIS by Lin and Ying (2013) and MMA by Pan et al. (2013). Han et al. (2015) presented an ABC with Differential Evolution (ABC-DE) for the $F_m|block|C_{max}$ problem. In the proposed ABC-DE, initial population is composed by MME heuristic and random method solutions. A DE is incorporated in the employed bee phase. The tournament selection and a self-adaptive strategy are applied in the onlooker bee phase. In the scout bee phase, several insertion operators are applied to unchanged solutions. The proposed metaheuristic outperformed TS + M by Grabowski and Pempera (2007), HDDE by Wang et al. (2010), DABC by Han et al. (2011), IABC by Pan et al. (2012), $DPSO_{svns}$ by Wang and Tang (2012) and EDE_C by Davendra et al. (2012). Liu et al. (2015) proposed a hybrid metaheuristic, based on the incorporation of Scatter Search Algorithm (SSA) into Variable Neighborhood Search (VNS), for the $F_m|block|C_{max}$ problem. In the proposed hybrid approach, called SSVNS, R is used to store solutions with good quality and diversity. A modified NEH is applied to generate the initial solution. The authors established an index to select each sequence from R to apply the local search. A job block-based local search is used to generate new solutions. Set R is updated for every new solution generated. Experimental results showed that SSVNS outperformed TS+M by Grabowski and Pempera (2000), DMS-PSO by Liang et al. (2011) and achieved the best results for 10 of 12 instance groups in comparison with H-EDA by Jarboui et al. (2009). Sadaqa and Moraga (2015) proposed Meta-Heuristic for Randomized Priority Search (Meta-RaPS) for the $F_m|block|C_{max}$ problem. Meta-RaPS creates an initial sequence based on randomness followed by a construction step that employs NEH insertion scheme to build a new solution. Computational results evidenced that the proposed metaheuristic outperformed mNEH by Wang et al. (2012) in small size problems. Tasgetiren et al. (2015) proposed a novel population local search with differential evolution (DE_PLS) for the $F_m|block|C_{max}$ problem. DE_PLS uses PF-NEH(x), a modified version of GRASP and random method to generate the initial population. The algorithm includes a mechanism that applies IG or ILS to the trial individual. The new individual is subject to a Referenced Local Search, which is guided by the best solution found so far. Computational results showed that the proposed DE_PLS generated better results than all the evaluated algorithms of the literature, except for MMA by Pan et al.

(2013). Wang et al. (2015) proposed a Modified Cuckoo Search Algorithm (CuckooSA) for the $F_m|block|C_{max}$ problem. CuckooSA is a population-based stochastic search algorithm, inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds. CuckooSA requires an initial population of host nest, where a host nest represents a permutation. At each iteration, the host nests are evaluated by their fitness values. The population is updated according to the quality of solution. The initial population of the proposed CuckooSA is composed by solutions generated by a modified NEH. To enhance the exploitation ability, Insertion and swap neighborhood local searches are sequentially used, where the first local search structure to be applied is chosen given a probability. Experimental results showed that the proposed algorithm outperformed DMS-PSO by Liang et al. (2011), TS and TS + M by Grabowski and Pempera (2007), HDDE by Qian et al. (2009), however, in larger computational time. Zhang et al. (2015) proposed a hybrid Variable Search Neighborhood with Simulated Annealing for the $F_m|block|C_{max}$ problem. Five neighborhood structures are considered to be applied: swap, forward insert, afterward insertion, forward swap and inverse. HVNSSA uses random method to generate an initial solution. A Simulated Annealing, which uses the construction/destruction mechanism, is also integrated to HVNS. Computational results evidenced that the proposed HVNSSA is competitive in comparison with the evaluated metaheuristics. Eddaly, Jarboui, and Siarry (2016) proposed some versions of PSO for the $F_m|block|C_{max}$ problem. In PSO, each sequence is represented by a particle of the swarm. In the proposed Combinatorial PSO (CPSO), initial swarm is composed by one particle provided by NEH and the remaining is randomly generated. New particles are created according to a procedure that applies swap operator. In Hybrid CPSO (HCPSO), an ILS is applied to the current best solution, given a probability. Experimental results showed that both CPSO and CPSO_{LS} outperformed GA by Caraffa et al. (2001), TS by Grabowski and Pempera (2007) and HDDE by Wang et al. (2010). Han et al. (2016) proposed a modified Fruit Fly Optimization Algorithm (MFFOA) for the $F_m|block|C_{max}$ problem. FFO is a swarm intelligence approach, which simulates the intelligent foraging behavior of fruit flies. FFO is composed by three stages: initialization stage, smell-based search stage and vision-based stage. The proposed MFFOA uses MME heuristic and random method to create the initial population. In the smell-based search stage, a neighborhood local search strategy, based on insertion and swap operators is applied to the current solutions. Then, an insertion local search is applied to the new solutions with a certain probability. In visitation-based search, the best solution found so far replaces the worst solution of the population. Experimental results showed that the modified MFFOA outperformed HDDE by Wang et al. (2010), IG by Ribas et al. (2011), EDEc by Davendra et al. (2012), DABC by Han et al. (2011), IABC by Han et al. (2012) and DE-ABC by Han et al. (2015). Tasgetiren et al. (2017) proposed some heuristics methods for the $F_m|block|C_{max}$ problem. The developed PFT-NEH(x) is similar to PF-NEH(x). In PFT, initial list is established based on the data of the processing times. In the iterative phase of PFT, job assigned to position i of initial list is assigned to the first position of the new sequence. The remaining jobs are scheduled according to a similar way of PF. In total, x sequences are generated by PFT, where x is set according to the number of jobs n. In PFT-NEH(x), NEH insertion scheme is applied for the last δ jobs of sequence generated by PFT. Experimental results showed that the proposed heuristic, in general, outperformed the heuristics by Pan and Wang (2012). The authors also proposed some versions of IG with iteration jumping (IG_IJ) and IG with referenced insertion local search (IG_RIS). Both methods use a mechanism that chooses between RIS or swap neighborhood search to be performed. Additionally, IG_IJ uses the Simulated Annealing (SA) acceptance criterion to the new solution. Computational results evidenced that IG_RIS and IG_IJ outperformed all the versions of IG by Ribas et al. (2011), HDDE by Wang et al. (2010) and some versions of VNS by Ribas and Companys (2013). Toumi et al. (2017) proposed a B&B for the $F_m|block|C_{max}$ problem. The authors designed lower bounds based on Gilmore and Gomore Algorithm (LB_{GG}) and (Ronconi, 2005) (LB_{RO}). The authors established a number of nodes to use both lower bounds during the running of B&B. Computational results showed the superiority of the Gilmore and Gomory algorithm based lower bound against the other existing in the literature. Liu, Wang, and Zhang (2018) developed an Improved Biography-Based Optimization Algorithm (IBBO) for $F_m|block|C_{max}$. BBO is based on the biographic model, which describes how species are produced, migrated and destroyed. In BBO, each individual is called habitat with a Habitat Suitability Index (HIS), which is associated with a suitability index variables (C_{max}). In the proposed IBBO, authors applied NEH and random method to generate the initial population. In the next phase of IBBO, solutions are sorted in descending order of quality of solution. In migration operation, worst solutions are enabled to obtain additional information from the best solutions immigration and emigration rates. Mutation phase is composed by a swap move. Finally, an insertion neighborhood is used to improve the best solution of the population. Computational results showed that IBBO outperformed the B&B proposed by Ronconi (2005). Panwalkar and Koulamas (2018) reviewed existing network schematics for flow shop scheduling problems and proposed new schematics for the flow shops with blocking, synchronous transfers, no-wait, no idle and limited buffers. Given that $F_m|block|C_{max}$ is represented with processing times on arrows in most part of the literature, the authors proposed a new blocking flow shop schematic with processing times on nodes. Shao, Pi, and Shao (2018b) developed an Estimation Distribution Algorithm with Variable Search Neighborhood (P-EDA) for the $F_m|block|C_{max}$ problem. Initial population is generated through NEH-based heuristic and random method. Selection phase uses a modified linear rank selection to select the superior individuals from the population. A probabilistic method is proposed to guide the algorithm to promising solution space. A path relinking method is used to enhance the convergence property of EDA. To enhance the exploitation capability, a modified referenced local search is applied to the individuals. Finally, a diversity-maintaining scheme is adopted to avoid the deterioration of the population. Experimental results showed that the proposed EDA-VNS generated better results in comparison with HDDE (Wang et al., 2010), hmgHS (Wang et al., 2011), IABC (Han et al., 2012), DE-ABC (Han et al., 2015), MFFO (Han et al., 2016), IG (Ribas et al., 2011), TPA (Wang et al., 2012), RAIS (Lin & Ying, 2013), SVNS (Ribas & Companys, 2013), MMA (Pan et al., 2013) and B-IG (Ding et al., 2015). Ozolins (2019) proposed a bounded dynamic programming for the $F_m|block|C_{max}$ problem. Two lower bounds are proposed, based on Bautista et al. (2012) and Gilmore and Gomory (1964). Experimental results showed that, on average, the proposed lower bound outperformed the lower bound by Ronconi (2005).

3.1.1.2. $F_m|block,\beta_i|C_{max}$. Logendran and Sriskandarajah (1993) studied the $F_2|block, st_{ant}|C_{max}$ problem. The authors proved that the problem is NP-hard in strong sense. The proposed Algorithm H applies Gilmore and Gomory Algorithm to find a schedule for the problem. Algorithm H was compared with H_R, version of H that generates random sequences. The authors concluded that, in the average case, Algorithm H outperformed H_R. Levner, Kogan, and Maimon (1995) represents a robotic cell as two-machine blocking flow shop with setup time effects, several automated storages and retrieval stations with C_{max} minimization. The authors showed that the studied scheduling problem with job-dependent transportation times may be solved in cubic time using a special structuring of the Gilmore–Gomory algorithm. For the $F_m|permu,no\text{-}wait,block|C_{max}$ problem, Grabowski and Pempera (2000) proposed a version of Tabu Search. The authors presented two properties of the problem, based on block theory, with the aim to eliminate moves which do not improve C_{max}. In the proposed TS, NEH is used to generate the initial solution. The neighborhood is defined according to the application of insertion neighborhood local search, which generates $(n-1)^2$ neighbors. A reduction of the neighborhood is carried out applying the proposed properties. Besides the application of TS in a study case, computational results showed that TS outperformed SPIRIT by Widmer and Hertz (1989). Soukhal et al. (2005) provided some complexity results for the two-machine flow shop with transportation times. The problem is denoted as $F_2 \rightarrow D|block,v,c_e,F_i|C_{max}$, where $F_2 \rightarrow D$ means that the jobs are processed firstly on two machine flow shop and after delivered to the costumer, v is the number of identical trucks, c_e is the capacity of the trucks and F_i is the elapsed time to transport job i from the output system to the costumer and to come back. The authors proved that $F_2 \rightarrow D|block,v=1,c_e=2|C_{max}$, $F_2 \rightarrow D|block,v=1,c_e=2,F_i=h|C_{max}$ and $F_2 \rightarrow D|block,v=1,c_e=3|C_{max}$ problems are strongly NP-hard. Yuan et al. (2007) extended some complexity results from Soukhal et al. (2005) for the $F_2 \rightarrow D|block,v,c_e,F_i|C_{max}$. The authors proved that $F_2 \rightarrow D|block,v=1,c_e=2|C_{max}$ is binary NP-hard and $F_2 \rightarrow D|block,v=1,c_e=3,F_i=h|C_{max}$ is strongly NP-hard. Liu and Kozan (2009) proposed a constructive heuristic, called Liu–Kozan algorithm, for the $F_m|prmu, no\text{-}wait, block, b_{j,j+1}|C_{max}$ problem. The authors developed some procedures to compute the completion time of each buffer conditions (*prmu*, *no-wait*, *block* and $b_{j,j+1}$) and a *tune-up* procedure to adjust the start time, blocking time, no-wait time, storing time, departure time and completion time of jobs to each buffer condition. The Liu–Kozan Algorithm constructs the sequence according to the Best Insertion Heuristic (BIH) scheme, where, all jobs are inserted in all positions of the partial sequence. The partial sequence with the ith job that causes the minimum C_{max} is chosen to the next iteration and job i is removed from the non-scheduled set of job. The heuristic finishes when all jobs are scheduled. Computational results showed the performance of the algorithm under a flow shop with different buffer conditions. Carlier et al. (2010) proposed a branch-and-bound and a GA for the scheduling problems of jobs in a robotic cell, specified as $F_m|block,t_i|C_{max}$, where t_i is defined as the time that a robot takes to transfer job i from the current machine to the next machine. The authors proposed one-machine based and two-machine based lower bounds to be applied on the proposed branch-and-bound. For the developed GA, the similar job order operator and interchange of jobs operators are applied in crossover and mutation phases, respectively. Experimental results showed that GA requires less computational time and is useful for solving large instance problems. Lee et al. (2010) studied the $F_2|block, \alpha p_{ij}|C_{max}$ problem. The authors developed some dominance rules for node elimination and a lower bound to the proposed branch-and-bound algorithm. Four heuristics are proposed for the problem. The first heuristic (H_1) is based on Johnson's rule. The second algorithm, H_2, aims to minimize the impact of the idle time of the next job scheduled. The third method, H_3, aims to minimize C_{max} of the partial sequence. The fourth method, H_4, aims to reduce the blocking times of the first machine or the idle time of the second machine. Experimental results indicated that the branch-and-bound algorithm could solve problems with up to 32 jobs in a reasonable computational time. Regarding the heuristics methods, H_4 was the best evaluated. Tang and Tang (2011) proposed a MILP and a Particle Swarm Optimization (PSO) for the $F_2|b_{j,j+1}, batch, st_{si}|C_{max}$ problem. A standard version of PSO is proposed and a local search is incorporated to the algorithm to try to improve the position of the particles. Computational results showed that both proposed solution approaches reached effective solutions. Chen et al. (2014) developed a Hybrid Discrete DE (HDDE) for the $F_2|block, batch(1), r_j|C_{max}$ problem, where $batch(1)$ denotes that the first machine processes the jobs in batch. The initial population is established according to LPT, SPT, Johnson's rule and random method. In order to form the batches, First-Fit (FF) heuristic is proposed and the least idle/blocking time (LIBT) is used to scheduling the batches. An insertion local search is used to improve the quality of the solution. Experimental results showed that HDDE outperformed SA by Manjeshwar, Damodaran, and Srihari (2009) and GA by Manjeshwar, Damodaran, and Srihari (2011). Al-Salem and Kharbeche (2016) studied the application of heuristic methods for the two-machine robotic cells with controlled processing times scheduling problem, denoted as $F_2|batch(1), block, p_{i,j}^{min} \le p_{i,j} \le p_{i,j}^{max}|C_{max}$, where $p_{i,j}^{min} \le p_{i,j} \le p_{i,j}^{max}$ is the interval duration of processing times. Given the proposed MILP of the problem, the authors developed a MIP-based heuristic, which consists of solve the LP relaxation of the MILP and scheduling the sequence of jobs and the set of robot moves. The authors also adapted GA, which was designed to scheduling jobs and robot moves. Experimental results showed the effectiveness and efficiency of the proposed MILP and heuristic methods. Fu et al. (2017) proposed a Chemical Reaction Optimization with Variable Neighborhood Search (CRO-VNS) for the $F_m|block, \alpha p_{ij}|C_{max}$ problem. CRO algorithm is a population-based algorithm that simulates a set of molecules microscopically and tries to capture the energy during the reaction process. Each solution is represented by a molecule. Four elementary reactions are used to try to optimize the problem: on-wall ineffective collision, decomposition, inter-molecular ineffective collision and synthesis reactions. In the proposed CRO-VNS, in on-wall ineffective collision phase, molecules are modified through insertion and swap local searches. In decomposition one molecule is broken into two or more by swapping randomly jobs. In synthesis reaction, two molecules are combined to form a new molecule by distance preserving crossover. Finally, a VNS is applied to the best solution found. Computational results showed that CRO-VNS outperformed the standard version of CRO and the classic version of GA. Khorasanian and Moslehi (2017) provides some results for $F_2|block,mtflx(1), prmp|C_{max}$. A lower bound on the makespan is proposed to lessen solutions space. The authors proved that the problem is strongly NP-hard. Two MILP are proposed, using the developed lemas. A basic VNS and a Dynamic VNS (DVNS) are proposed for the problem. In both VNSs, initial solution is given by a modified version of NEH. In shaking phase, five neighborhood structures based on insertion and swap are tuning to be performed in randomly selected jobs. In the local search phase, an insertion neighborhood structure is used. Computational experiments showed that DVNS outperformed hmgHS by Wang et al. (2011), TPA by Wang et al. (2012) and HVNS by Moslehi and Khorasanian (2014) in $n \ge 75$ jobs. Takano and Nagano (2017) developed a B&B and a mixed integer linear programming (MILP) for the $F_m|block, st_{sd}|C_{max}$ problem. The proposed B&B follows the algorithm structure of Ronconi (2005) and four lower bounds were developed. Both methods were implemented and tested against each other. Computational results evidenced that, in general, the proposed B&B outperformed MILP. An index rule was developed by Koulamas and Panwalkar (2018) for the $F_m|no\text{-}wait,ord|C_{max}$. The authors extended the application of the index rule to the $F_m|block,ord|C_{max}$ problem. Based on previous results of slowing down operations by Abadi et al. (2000), the index rule can obtain good solutions for the problem. (Shao, Pi, and Shao (2018a), (2018b)) proposed a Discrete Water Wave Optimization (DWWO) for the $F_m|block, st_{sd}|C_{max}$ problem. Two heuristics are proposed by the authors: PFT+NEH(x) and PFT+RZ(x), based on PFT, NEH and RZ local search, developed by Rajendran and Ziegler (1997). WWO is a population-based optimization algorithm inspired by shallow wave model. The searching process of WWO is viewed as wave motions including propagation, breaking and refraction. In the proposed DWWO, each permutation is represented by a wave (individual). In order to compose the initial population, PFT+NEH(x), PFT+RZ(x) and a swap mechanism perturbation are used. In propagation phase, a perturbation scheme is used to generate new waves and RIS is applied to each new generated wave. In the refraction phase, path relinking method with a swap perturbation mechanism is proposed. Finally, a VNS is applied in the breaking phase. According to the experimental results, DWWO outperformed 23 metaheuristics designed for $F_m|block|C_{max}$, $F_m|st_{sd}|C_{max}$ and $1|st_{sd}|C_{max}$ problems. Takano and Nagano (2019) evaluated the performance of the main constructive heuristics of the literature to solve the $F_m|block, st_{sd}|C_{max}$ problem. The authors adapted methods designed for the $F_m|block, st_{sd}|C_{max}$ problem, proposed by McCormick et al.(1989), Ronconi (2004) and Pan and Wang (2012). Among the 128 evaluated methods, computational results showed that the adapted version of PF by McCormick et al. (1989), denominated PF1, obtained the best performance.

3.1.1.3. F_m| RCB,block or/and RCb, $\beta_i|C_{max}$.* Martinez et al. (2006) provided complexity results for the *m*-machine flow shops with *block* and *RCb* conditions to minimize C_{max}. The authors proved that $F_2|RCb|C_{max}$, $F_3|RCb|C_{max}$, $F_4|RCb(1,2), block(2,3), RCb(3,4)|C_{max}$, $F_3|RCb(1,2), block(2,3)|C_{max}$ and $F_3|block(1,2), RCb(2,3)|C_{max}$, are polynomial problems. The authors also showed that $F_5|RCb|C_{max}$ $F_4|RCb(1,2), RCb(2,3), block(3,4)|C_{max}$, $F_4|block(1,2), RCb(2,3), RCb(3,4)|C_{max}$ and $F_4|block(1,2), RCb(2,3), block(3,4)|C_{max}$ are NP-hard problems. Sauer and Sauvey (2012) proposed a GA for the $F_m|RCb|C_{max}$ problem. The developed GA uses a meta-gene association, which detects potential genes of the best individuals of the population with the purpose of identify "good" association patterns. GA was compared with adapted NEH by Martinez (2005), HSS2 heuristic by Sauvey and Sauer (2009), SA by Martinez et al. (2004) and Electro-Magnetism like optimization heuristic by Yuan and Sauer (2007). Trabelsi et al. (2011) provided some complexity results for F_m| RCB,block,RCb*$|C_{max}$ problems. The authors proved that $F_4|RCb(1,2), block(2,3), RCb(3,4)|C_{max}$ and $F_3|RCb^*|C_{max}$ are polynomial problems. The authors developed a linear mathematical model for the F_m| RCB,block, RCb*$|C_{max}$ problem. Computational results showed that flow shop problems with mixed blocking constraints are at most less complicated, so fast to find optimal solutions, than problems with only one blocking constraint. Trabelsi et al. (2012) proposed constructive heuristics and a genetic algorithm (GA) for the $F_m|prmu,block,RCb^*,RCb|C_{max}$ problem. The authors proposed TSS heuristic and some local searches based on TSS and NEH. TSS chooses the next job of the sequence according to the minimum sum of partial C_{max}, the sum of inactive times and the sum of jobs already scheduled. Computational results showed that both heuristics are competitive and the proposed GA was efficient to a small number of jobs. Khorramizadeh and Riahi (2015) proposed an ABC for the $F_m|prmu,block,RCb^*,RCb|C_{max}$ problem. The initial population, with *p* solutions, was generated by NEH and random method. In the employed bee phase, all the *p* solutions are perturbed by inserting a job randomly generated to other position, generating *p'*. Thus, path relinking is applied to *p'* in order to generate a certain number of new solutions. A local search is applied to the best solutions generated. In the onlooker bee phase, a local search based on insertion and exchange neighborhoods are used at a determined number of iterations. A combined method based on Taguchi orthogonal arrays are used in the scout bee phase. Experimental results showed that the proposed ABC outperformed GA by Trabelsi et al. (2012). Riahi et al. (2017) proposed a Scatter Search Algorithm (SSA) for the $F_m|prmu,block,RCb^*,RCb|C_{max}$ problem. In the proposed SSA, initial population is composed by a solution generated by a modified version of NEH-WPT, proposed by Wang et al. (2011). An Iterated Local Search (ILS) with NEH-perturbation scheme and insertion and swap local searches are applied in the improvement phase. In the combination phase, SSA applies an adapted version of path relinking proposed by Khorramizadeh and Riahi (2015). Computational results showed that SSA outperformed GA by Trabelsi et al. (2012) and ABC by Khorramizadeh and Riahi (2015). Riahi et al. (2018) developed three versions of the constraint guided local search (CGLS) for the $F_m|st_{sd}, RCb, RSb|C_{max}$ problem. Basically CGLS(1), CGLS(2) and CGLS(3) start from an initial solution and a VNS is used to improve solutions until a stop criterion is reached. For CGLS(1), a variant NEH, called NNEH, proposed by Riahi et al. (2017) is used as initial sequence. In CGLS(2) and CGLS(3), initial sequence is generated by a proposed modified version of NEH. A Variable Neighborhood Descent (VND) is used as local search in all proposed methods. Experimental results showed that all local searches proposed significantly outperformed IG by Ruiz and Stützle (2008) and GRASP by Ribas and Companys (2015).

3.1.1.4. $F_m|b_{j,j+1},\beta_i|C_{max}$. Dutta and Cunningham (1975) developed a dynamic programming for the $F_2|b_{j,j+1}|C_{max}$ problem with $0 \leq b < \infty$. For $b=0$ or 1, only relatively poor solutions were obtained, given C_{max} of 15% greater than the optimal. Papadimitriou and Kanellakis (1978) provided some complexity results for two-machine flow shop with limited buffers. The authors proved that $F_2|b_{j,j+1}|C_{max}$ with $0 < b < \infty$ is NP complete. Leinsten (1990) proposed two constructive heuristics for the $F_m|b_{j,j+1}|C_{max}$ with $0 < b < \infty$. The first heuristic (BFSE) is designed to maximize the usage of the buffer and to avoid the blocking on the first machine. The second heuristic, denominated BFSP, generates two sequences, one from the start of the sequence *J* and other from the end sequence *J'*. In both sequences, jobs are selected according to the minimum difference between the completion time of adjacent jobs. Results showed that both BFSE and BFSP are competitive with NEH by Nawaz et al. (1983). Norman (1999) proposed a Tabu Search (TS) for the $F_m|b_{j,j+1},st_{sd}|C_{max}$ problem. The author considered as neighborhood all the $(n-1)^2$ sequences generated by insertion local search. Computational results showed that TS outperformed NEH by Nawaz et al. (1983), Profile Fitting (PF) by McCormick et al. (1989) and BFSE by Leinsten (1990) in all $0 \leq b_{j,j+1} \leq b$ scenes evaluated. Nowicki (1999) proposed a TS for the $F_m|b_{j,j+1}|C_{max}$ problem. The author described some block properties, which is applied to insertion neighborhood local search to remove "useless moves". The proposed algorithm uses a generalization version of NEH as initial solution, $(n-1)^2$ moves generated by insertion local search to compose the neighborhood and memories of short-term and long-term search to store moves. Computational results showed that TS outperformed SA and fast SA by Van, Cummings, and Le Van (1990) when $b > 0$. Li and Tang (2005) extended the studies from (Grabowski & Pempera, 2000) and present new block properties for the $F_m|b_{j,j+1}|C_{max}$ problem. The authors also proposed a speed up method to compute C_{max} for insertion neighborhood moves. The properties are applied in TS, which uses NEH as initial solution, neighborhood based on insertion moves and reduced according to the developed properties. Three version of TS are tested, TS^N, which uses the classic neighborhood, TS^{H1} with the new neighborhood and TS with the classical neighborhood and with the speed up method. Computational results evidenced that TS^{H1} outperformed the other proposed versions of TS. Wang et al. (2006) developed an hybrid GA (HGA) for the $F_m|b_{j,j+1}|C_{max}$ problem. The authors used the properties presented by Nowicki (1999) to the proposed algorithm to eliminate "useless moves" in the neighborhood created by an insertion local search. HGA uses multiple crossover operators in a hybrid sense and parallel, i.e., one operator is picked at random to perform the crossover. In the next step of the algorithm, mutation phase or the Nowicki's local search is applied given a probability d_p. The authors also apply different mutation operators (Swap, Inverse and Insert) to the problem. Computational results showed that HGA outperformed TS by Nowicki (1999) and Simulated Annealing (SA) by Ogbu and SMith (1990). Liu et al. (2008) proposed a Hybrid Particle Swarm Optimization (HPSO) for the $F_m|b_{j,j+1}|C_{max}$ problem. HPSO incorporates the block-based local search proposed by Nowicki (1999) to compose the algorithm structure. A Simulated Annealing (SA) with multiple local searches and adaptive meta-Lamarckian learning strategy is also incorporated to the proposed HPSO. The proposed adaptive meta-lamarckian strategy determines which local search (insertion, swap or inverse) is chosen to be performed, in order to maintain diversity of the population. Experimental results showed that HPSO is competitive compared with TS by Nowicki (1999), PSO with Variable Neighborhood Search (PSO_{VNS}) by Tasgetiren et al. (2004) and HGA by Wang et al. (2006). Qian et al. (2009) proposed a hybrid DE for the $F_m|b_{j,j+1}|C_{max}$ problem. The proposed HDE uses the same DE-based configurations by Storn and Price (1997). Additionally, a block insertion local search is applied to the best solution found. Computational results showed that, in the case $b=0$, the proposed HDE outperformed RON by Ronconi (2005). Pan et al. (2011) proposed a Hybrid Discrete Differential Evolution (HDDE) for the $F_m|b_{j,j+1}|C_{max}$ problem. A procedure to create the initial population is developed, which uses NEH to proportionate a higher diversity and a certain individuals quality level. A local search composed by a mechanism that uses insertion and swap neighborhood search to improve the sequence is applied after the crossover phase. Computational results showed that HDDE outperformed Hybrid GA (HGA) by Wang et al. (2006), Hybrid PSO (HPSO) by Liu et al. (2008), P_DDE by Pan et al. (2008) and W_DDE by Wang, et al. (2010). Pan et al. (2011) proposed a Chaotic HS (CHS) for the $F_m|b_{j,j+1}|C_{max}$. CHS applies a chaotic local search to the selected solution. Additionally, a Simulated Annealing-based (SA-based) mechanism is used to accept a solution with inferior quality. Computational results showed that, in the case $b=0$, CHS outperforms the Improved HS by Mahdavi, Fesanghary, and Damangir (2007), HGA by Wang et al. (2006), Best Global HS by Omran (2008) and HPSO by Liu et al. (2008). Moslehi and Khorasanian (2014) proposed a hybrid VNS (HVNS) for the $F_m|b_{j,j+1}|C_{max}$ problem. HVNS uses the NEH-WPT to generate the initial solution. In order to generate a new neighbor, a shaking phase applies insertion, edge-insertion or swap local search in the initial solution. After, a Simulated Annealing (SA) insertion based local search is used to explore new neighborhoods. Experimental results evidenced that, in the case $b=0$, HVNS outperformed HGA by Wang et al. (2006), HPSO by Liu et al. (2008), CHS by Pan et al. (2011), hmgHS by Wang et al. (2011), TPA by Wang et al. (2012) and RAIS by Lin and Ying (2013). Zhao et al. (2014) proposed an improved PSO (LDPSO) for the $F_m|b_{j,j+1}|C_{max}$ problem. A linearly decreasing disturbance term to update the velocity formula is developed to solve the slow convergence rate of the system in the later processes. Experimental results showed that the proposed LDPSO was superior in comparison with other algorithms.

3.1.2. Total Flow Time ($\sum_i C_i$)

From the total of papers covered, 11% adopted Total Flow Time-based ($\sum_i C_i$) as mono-objective function minimization. As shown in Table 3, all the listed papers dealt with the development of solution methods. Weng (2000) proposed a Tabu Search (TS) for the $F_m|block|\sum_i C_i/n$ problem. The author applied a modified version of NEH, designed to minimize the mean total flow time, to generate the initial solution. The neighborhood considered is the pairwise exchange. Computational results showed that as the number of buffers or machines increased, TS improvement decreased. Wang et al. (2010) proposed some versions of Harmony Search (HS) for the $F_m|block|\sum_i C_i$ problem. In the proposed basic Harmony Search (HS), harmony memory is composed by solutions generated by NEH-WPT and random method. In NEH-WPT, initial list is set by jobs sorted in non-decreasing order of total processing times. Thus, NEH insertion scheme are applied to the initial list. In memory consideration, a harmony is randomly selected in the harmony memory. A pitch adjustment is applied to the harmony given a probability. In the Global HS algorithm (gHS), pitch adjustment functions from the literature are used. In Modified global best HS (mgHS), a new pitch adjustment function is developed. Hybrid versions of HS (hHS), gHS (hGHS) and mgHS (hmgHS) apply an insertion neighborhood local search to stress exploitation. Computational results showed that hmgHS obtained the best performance. Deng, Xu, and Gu (2012) proposed a Discrete Artificial Bee Colony (DABC) for the $F_m|block|\sum_i C_i$ problem. Initial population is generated by NEH-WPT and random method. Employed bee phase applies a procedure based on the best random insertion. A combined local search with insertion and swap neighborhood search is used in the onlooker bee phase. In scout bee, the best random insertion is applied to the best solution of the population. Finally, an insertion neighborhood local search is used to the new food. Experimental results showed that DABC outperformed DABC by Tasgetiren et al. (2011), HDDE by Wang et al. (2010) and Iterated Greedy (IG) by Ruiz and Stützle (2007). Khorasanian and Moslehi (2012) proposed an NEH-MK constructive heuristic and two versions of Iterated Greedy (IG) for the $F_m|block|\sum_i C_i$ problem. In the proposed NEH-MK heuristic, jobs are initially sorted in non-decreasing order of total processing times. Thus, NEH-insertion scheme is applied to the initial list. At each job inserted, all jobs already scheduled are reinserted in the best position, one by one, except the new job. Finally, a pair wise local search tries to improve the final sequence. The solution given by NEH-MK is used as input for the proposed two versions of IG for the $F_m|block|\sum_i C_i$ problem. Experimental results showed that both IGs outperformed hmgHS by Wang et al. (2010). Han et al. (2013) developed some versions of Artificial Bee Colony (ABC) for the $F_m|block|\sum_i C_i$ problem. In the proposed Discrete ABC, initial population of food sources is given by random method. In the employed bee stage, firstly, insertion and swap operators are used to generate a neighbor list (NL) and a self-adaptive strategy are applied to NL. An Estimation Distribution Algorithm (EDA) is applied in the onlooker be stage. Three versions of hybrid DABC (hDBAC) are also proposed. The authors applied both MME-A and MME-B to generate initial solutions for the three versions of hDABC. In hDBAC1, hdABC2 and hABC3. A referenced local search is applied in employed bee stage, onlooker bee stage and scout bee stage in hDBAC1, hdABC2 and hABC3, respectively. Computational results showed that the proposed metaheuristics outperformed HS by Wang et al. (2010). Li et al. (2013) proposed a mathematical model using Max-algebra theory for the $F_m|block|\sum_i C_i$ problem and settled it with a dynamic programming heuristic. Computational experiments showed that the proposed HDP produced near optimal practical solutions. Moslehi and Khorasanian (2013) developed some solution methods for the $F_m|block|\sum_i C_i$ problem. Two MILP are proposed, where the first model is based on the departure times and the second model is based on idle and blocking times. NEH-MK, proposed by Khorasanian and Moslehi (2012), is used to generate an initial upper bound for the B&B. Three lower bounds are developed, based on the underestimate of the departure time of all non-scheduled jobs. Computational results showed the superiority of the proposed lower bounds compared to the lower bound by Ronconi and Armentano (2001). Toumi et al. (2013d) and Toumi et al. (2013c) developed two B&B's for the $F_m|block|\sum_i C_i$ problem. The authors proposed lower bounds based on Ronconi and Armentano

(2001). Both authors generated the best results found for the Taillard's instance problems (up to 200 jobs and 20 machines). Constructive heuristics and some versions of Greedy Randomized Adaptive Search Procedure (GRASP) for the $F_m|block|\sum_i C_i$ problem are designed by Ribas and Companys (2015). The constructive heuristics are based on PF by McCormick et al. (1989). PF, HPF1and HPF2 uses an index evaluation regarded the total flow time to select the next job to be inserted. HPF2 uses a different index, based on the data of the problem, to select the first job of the sequence. In NPF, NHPF1 and NHPF2, NEH insertion scheme is applied to the solutions generated by PF, NHPF1 and NHPF2, respectively. GRASP is a multi-start or iterative process, which consists of two phases: construction and local search. In the construction phase, a feasible solution is built, adding one element at a time. Thus, the neighborhood of the given solution is evaluated until a local minimum is found during the local search phase. The best found solution is kept as the result. The algorithm restarts the search until a stopping criterion is reached. Four versions of GRASP are proposed and uses HPF1, HPF2, NHPF1 and NHPF2 as initial solutions. Experimental results showed that all the heuristics presented similar performances, being HPF2 and NHPF2 the best heuristics for the problem. Regarding the metaheuristics, the authors recommended the use of GRASP(HPF2). Ribas, Companys, and Tort-Martorell (2015) proposed a DABC for the $F_m|block|\sum_i C_i$ problem. The initial population of the proposed DABC is composed by solutions given by HPF2 and random methods. In the employed phase, three operators based on insertion and swap are applied to food sources. In the onlooker bee phase, authors used path relinking method to improve the food sources. In scout bee phase, population is updated if the new generated solutions are better than the worst solutions of the population. Computational results showed that DABC outperformed HDDE by Wang et al. (2010), DABC by Deng et al. (2012) and IG by Khorasanian and Moslehi (2012). Fernandez-viagas et al. (2016) proposed Beam-Search-Based (BS(x)) for the $F_m|block|\sum_i C_i$ problem. The heuristic adopts a beam-search-based strategy that combines the diversification of the based-population algorithms and the efficiency of constructive heuristics. Some versions of the BS(x) outperformed the main heuristics designed for $F_m|block|\sum_i C_i$, $F_m|block|C_{max}$ and $F_m|\text{prmu}|C_{max}$ problems. Vo et al. (2015) proposed some general lower bounds for the $F_m|permu,\beta_i|\sum_i C_i$, where $\beta_i = block, RCb^{*}, RCb, delay, no\text{-}wait, st_{sd}, R_{nsd}$. The author developed lower bounds for the total completion time, based on the solution of two sub-problems: one problem similar to the single machine total completion time minimization and other similar to a travelling salesman problem. The MaxPlus theory is used to model lower bounds to the proposed branch-and-bound. Experimental evaluation was conducted to solve the $F_m|permu|\sum_i C_i$ and $F_m|permu, st_{sd}|\sum_i C_i$, but the authors suggested the use of the lower bounds for problems that lead with *block*, RCb^{*}, *RCb* constraints. Tasgetiren et al. (2016) developed tPF, tPF_NEH(x), PFT and PFT_NEH(x), based on PF_NEH(x) by Pan and Wang (2012) and a metaheuristic called Variable Block Insertion Heuristic (VBIH) for the $F_m|block|\sum_i C_i$ problem. In VBIH algorithm, a block of jobs with size BS is removed from the current sequence, and then, it makes a number of block insertion moves randomly in the partial sequence. The best sequence evaluated is chosen to go under a variable local search, based on insertion and swap neighborhood. As long as the solution improves, the block size keeps the same value, otherwise, the block size is incremented by one. A SA-based acceptance criterion is used to accept inferior solutions. The process stops when the block size reaches the maximum block size. Computational results showed that tPF_NEH(x) outperformed HPF2 and NHPF2 by Ribas and Companys (2015). VBIH outperformed DABC by Ribas et al. (2015). Ren et al. (2018) proposed a GA, TS and SA for the $F_m|block,r_i|\sum_i C_i^3$ problem. The authors developed some properties and dominance rules for the $F_2|block,r_i|\sum_i C_i^3$ and applied them to the proposed TS to reduce the search space and accelerate the convergence of the algorithm. Experimental results showed that the performance of SA applied to an industrial data tests dominated TS and GA.

3.2. Due date based measures

In this subsection, all the single performance measures related to due date of jobs are included. In total, 7% of the papers adopted total tardiness ($\sum_i T_i$) or weighted total tardiness ($\sum_i W_i T_i$) as mono-objective function. Table 4 lists publications which dealt with due date-based measures. All of them proposed solution methods for the problem. Armentano and Ronconi (2000) developed a constructive heuristic called LB-NEH and some versions of Tabu Search (TS) for the $F_m|block|\sum_i T_i$ problem. LB index is applied in the first phase of NEH and aims to order jobs according to the difference between due date and the total processing times of jobs. In the proposed Tabu Search (TS), LB-NEH is used as initial solution. As a strategy of neighborhood generation, insertion neighborhood local search is used. The authors compared all the version of TS with a B&B in order to verify their effectiveness. Ronconi and Armentano (2001) proposed a B&B for the $F_m|block|\sum_i T_i$ problem. The authors developed a lower bound on $\sum_i T_i$ based on the departure time of non-scheduled jobs. New upper bounds for the Taillard's instance problems were found by the proposed B&B. Ronconi and Henriques (2009) developed heuristic methods for the $F_m|block|\sum_i T_i$ problem. In FPD, the permutation is built according to a priority measure F_k, which considers the processing time profile of the last fixed job in the partial sequence as well as the due dates of candidates jobs. NEH insertion scheme is applied to the initial sequence generated by FPD. FPDNEH is used as input to the proposed GRASP, which uses three strategies to define the RCL length and insertion neighborhood local search as an improvement phase. Another version of GRASP applies path relinking method to intensify each local optimum of the current solution. The authors compared both GRASP versions with the solutions given by FPDNEH and B&B proposed by Ronconi and Armentano (2001). Januario, Arroyo, and Moreira (2009) proposed a GA for the $F_m|block|\sum_i T_i$ problem. Initial population is composed by adapted NEH and random method solutions. The authors used tournament selection, one-point and similar job order operators and insert operator in selection, crossover and mutation phases, respectively. An insertion local search is applied to the individuals generated in crossover and mutation. After GA optimization, two elite solutions are tried to be improved by path relinking. Computational results showed that the proposed GA outperformed GRASP by Ronconi and Henriques (2009). Ribas, Companys, and Tort-Martorell (2013) developed an Iterated Local Search with Variable Neighborhood Search (ILS_{VNS}) for the $F_m|block|\sum_i T_i$ problem. In the proposed ILS_{VNS}, the initial solution is given by NEH with EDD and NEH with FPD rule as initial sequences. In the local search phase, a VNS with two neighborhood structures, swap and insertion, is applied. The perturbation mechanism is used d times through the application of three operators (Pairwise Exchange, Forward Insertion and Backward Insertion). Computational results showed that ILS_{VNS} performed better than IG by Ribas et al. (2011) as the problem becomes more complex. Toumi et al. (2013a) and Toumi et al. (2013b) proposed some B&B's for the $F_m|block|\sum_i T_i$ problem. Toumi et al. (2013b) also developed lower bounds for the $F_m|block|\sum_i W_i T_i$. Both papers proposed lower bounds, based on Armentano and Ronconi (2000), and compared the lower bound with the lower bound proposed by Ronconi and Armentano (2001). In addition, the authors updated the upper bounds of (Ronconi & Armentano, 2001) instance problems. Nouha and Talel (2016) proposed a PSO for the $F_m|block|\sum_i T_i$ problem.

In order to generate the initial swarm, the seed sequences are subject to a revised NEH. A two-point crossover operation and referenced local search is used to update the position and velocity of the particles. Experimental results showed that the proposed PSO is competitive with GRASP by Ronconi and Henriques (2009). Nagano et al. (2017) developed an Evolutionary Clustering Search (ECS) for the $F_m|block|\sum_i T_i$ problem. ECS is a hybrid metaheuristic, which works in conjunction with evolutionary algorithms, in order to cluster solutions to detect regions supposing promising in the search space. The proposed ECS is composed by four independent components: a) Search metaheuristic (SM), which works as a solution generator along the iterations. Genetic Algorithm is adopted to represent SM; b) iterative clustering that aims to clustering similar solutions, in order to identify a representative cluster to minimize the problem; c) module analyzer, used to evaluate each cluster in regular intervals, in order to indicate the best promising cluster and d) A local search, which is applied in the promising cluster. Computational results showed that ECS outperformed ILS by Ribas et al. (2013). Shao et al. (2017) proposed seven improved versions of FPDNEH and Invasive Weed Optimization (IWO) for the $F_m|block|\sum_i T_i$ problem. All the proposed versions of FPDNEH are composed by different tie-breaking mechanisms, used during the NEH insertion phase. Experimental results showed that the proposed heuristics outperformed tPF_NEH(x) by Tasgetiren et al. (2016). IWO is an evolutionary optimization algorithm that tries to imitate the behavior of weeds in growth and colonizing. A weed represents a permutation of jobs. IWO is composed by initialization, reproduction, spatial dispersal and competitive exclusion. In the proposed self-adaptive discrete IWO (SaDIWO), one seed is generated by FPDNEH in initialization. In the reproduction phase, the weed produces a number of seeds according to its fitness value. In spatial dispersal phase, an IG is applied to all solutions and a self-adaptive mechanism controls the range of the neighboring solutions. In competitive exclusion, a distance-based competitive exclusion is used to select the weeds for the next generation. Finally, a VNS is applied to solutions generated in the spatial dispersal phase. Experimental results showed that the proposed IWO outperformed GRASP by Ronconi and Henriques (2009), GA by Januario, Arroyo, and Moreira (2009), ILS and IG by Ribas et al. (2013), steady state GA by Kellegöz, Toklu, and Wilson (2010), IG with Referenced Local Search (IG_RLS) by Karabulut (2016) and bi-population EDA by Shen, Wang, and Wang (2015).

3.3. Cycle time

The minimization of cycle time in production scheduling can be seen in papers that adopts repetitive production systems, as assembly flow lines. Proposition of solution methods and problem complexity analysis sums 3% from the total papers covered and it can be seen in Table 5. McCormick, Pinedo, and Wolf (1987) provided some complexity results for a repetitive production line with cycle time minimization. The authors proved that $F_m|block|CC$ problem is NP-Complete. McCormick et al. (1989) extended the studies of the previous authors. Profile Fitting heuristic (PF) is proposed for the $F_m|block|CC$ problem. In sum, PF selects the next job of the sequence according to the minimum idle time between the last job scheduled in the partial sequence and the current job to be inserted. Kamoun and Sriskandarajah (1993) studied the complexity of scheduling jobs in repetitive production systems. The authors proved that $F_3|no\text{-}wait(1,2), block(2,3)|C_t$ and $F_3|block|CC$ problems, respectively, are NP-hard in strong sense. Abadi et al. (2000) proposed the sequential slowing down (SSD) and parallel slowing down (PSD) heuristics for the $F_m|block|CC$ problem. SSD is based on GENIUS heuristic by Gendreau, Hertz, and Laporte (1992). The authors proved that for any sequence δ, $C_tB(\delta)$, calculated according to blocking aspects is equal to $C_tNWS(\delta)$, where NWS is the no-wait flow shop problem, if the processing times of some operations is slowed down. PSD and SSD were tested with a lower bound and two decomposition approaches.

3.4. Multi-objective based measures

As shown in Table 6, 12% of the papers treated the multi-objective functions minimization. Most of them proposed solution methods, which includes MILP and metaheuristics. Since some objective-functions often conflict with themselves, a set of optimal solutions namely Pareto-optimal are provided. Thus, a multi-objective optimization problem lies in identify all the elements that belong to Pareto set, which contains all solutions which are not dominated by any other solution. Prasad et al. (2006) considered the Kanban permutation flow shop scheduling problem with intermediate storages. The dual blocking occurs by the part type and queue size on action machine. The objectives of the problem are the minimization of mean completion time of containers ($\sum_i \bar{C}_{icont}$), mean completion time of part types ($\sum_i \bar{C}_{iparts}$) and standard deviation of mean completion time of part types ($\sum_i \sigma_i$). The authors proposed a modified version of Genetic Algorithm, called "non-dominated and normalized distance ranked sorting multi-objective genetic algorithm" (NDSMGA). Initial generation of GA included the generation of different individuals for each objective function. The sorting of the population is based on Pareto ranking. Thus, NDSMGA performs crossover or mutation according to a given probability. An additional phase applies pairwise exchange neighborhood local search to all individuals. Computational results showed that NDSMGA outperformed the other methods evaluated in most of the problems. Gong and Tang (2008) proposed a greedy heuristic for the $F_m|block, t_i, batch(1)|C_{max} + \sum_i B_i$ problem, where $batch(1)$ defines that the first machine processes a set of jobs in batch. Authors proved that the considered problem is strongly NP-hard even when the transfer times are equal to zero. The authors proposed greedy heuristic H, which uses LPT dispatching rule to create the batches. A worst-case rate analysis of Heuristic H is theoretically presented. Qian et al. (2009) proposed an hybrid DE (HDE) for both $F_m|b_{j,j+1}|C_{max}$ and T_{max} and $F_m|b_{j,j+1}|C_{max}$ and $\sum_j I_j$ problems. In HDE, the initial population is randomly generated. Besides the basic structure of DE, a problem-dependent local search with insertion neighborhood is applied to the individuals. Computational results showed that, in the case $b=0$, HDE outperformed IMMOGLS$_2$ by Ishibuchi, Yoshida, and Murata (2003) in both $F_m|b_{j,j+1}|C_{max}$ and T_{max} and $F_m|b_{j,j+1}|C_{max}$ and $\sum_j I_j$ problems. Cohn et al. (2010) studied the relationship between makespan and customer responsiveness (CR) in the blocking flow shop sequencing, denoted as $F_m|block|C_{max}$ and CR. Computational results showed how different approaches to making trade-offs between these two metrics can change the solution characteristics substantially. Gong et al.(2010) discussed some complexity results and proposed some approximation algorithms for the $F_2|\beta \rightarrow \delta,\ block,\ sharedst\ |\lambda C_{max} + (1-\lambda)\sum_j B_j$ problem. The authors showed that the considered problem is proved to be strongly NP-hard. The $F_2|\beta \rightarrow \delta,\ sharedst\ |\lambda C_{max} + (1-\lambda)\sum_j B_j$ problem can be optimally solved by Longest Processing Time-Last Only Not Full (LPT-LONF). Two approximation algorithms and a mathematical model are developed for the problem of sequencing and batching jobs. The authors concluded that the two approximation algorithms are useful to generate schedules with good quality. Maleki-Darounkolaei et al. (2012) proposed a MILP for the $3AF|block, s_{i,k,j}|\alpha(\sum_i W_iC_i/W)+(\alpha-1)C_{max}$ problem. The authors also proposed a basic version of Simulated Annealing (SA). Because of over much number of variables and constraints, SA performed better in computational time than MILP. Additionally, SA was able to solve instance problems in a more reasonable time in comparison with the MILP. Ventura and Yoon (2013) proposed a Novel Genetic Algorithm (NGA) for the $F_m|b_{j,j+1}, lot-str|\sum_i(T_i+E_i)$ problem with equal sublots. In the proposed Novel GA (NGA), initial population is randomly generated. Roulette wheel selection is used to pick a couple of individuals in selection phase. The partial matched chromosome operator (PMX) with inter-chromosomal interchange is used in crossover phase. In the mutation phase, a swap move is performed in the individual generated in crossover phase, given a fixed mutation rate. Computational results showed that the proposed NGA significantly improved the basic version of GA in all instances evaluated. Nouri and Ladhari (2015) proposed GA and ABC to $F_m|block|C_{max}$, $F_m|block|\sum_i C_i$ and $F_m|block|\sum_i T_i$. In the proposed Blocking GA (BGA), initial population is composed by solutions generated by NEH-WPT, mod-NEH-MK and random method. Roulette wheel selection and two-point operator are used in selection and crossover phases. A multi-parent crossover is used after mutation phase to clean population. In Blocking ABC (BABC), initial food source is generated by PF-NEH(x) and random method. In employed bee phase, a path relinking is performed to produce new food sources. In the onlooker bee phase, a referenced local search is applied to a sequence selected according to the roulette wheel selection operator. In scout bee phase, the worst solutions are replaced for new food sources by applying the insertion local search to the best solution. Computational results showed that both BGA and BABC outperformed MMA by Pan et al. (2013), DPSO by Wang and Tang (2012), DABC by Deng et al. (2012), IG by Ribas et al. (2011), HDDE by Wang et al. (2010), RAIS by Lin and Ying (2013), hmghs by Wang et al. (2011), hDABC by Han et al. (2013), IG by Khorasanian and Moslehi (2012) and GRASP by Ronconi and Henriques (2009). Guanlong, Shuning, and Mei (2016) proposed a Multi-Objective Discrete Group Search Optimization (MDGSO) for the $F_m|block|C_{max}+\sum_i C_i$ problem. GSO is a stochastic population-based that consists of three roles: Producers, Scroungers and Rangers. The purpose of producers is to explore the neighborhood region of a relatively better solution. The scrounger is designed by introducing the crossover operator in genetic algorithm. Ranger is designed to perform random search. In the proposed MDGSO, initial population is constituted by individuals generated by NEH, NEH-WPT and random method. In this case, the relatively better solutions are non-dominated solutions. In producer procedure, insertion-based pareto local search is performed to the set of non-dominated solutions (NS). If a random number is less than a probability value, MDGSO chooses scrounger phase to be performed. In this procedure, partially mapped crossover is performed to a random selected solution of set NS. Otherwise, the algorithm executes ranger procedure. Here, a random solution is selected from NS set and an insertion neighborhood local search is applied to it until no better solution is found. Computational results showed that MDGSO outperformed Non-Dominated Sorting GA by Deb et al. (2002) and bi-objective Multistart SA by Lin and Ying (2013). Nouri and Ladhari (2017) proposed a GA for the $F_m|block|C_{max}+\sum_i C_i$ problem. NEH-WPT, mod-NEH-MK and random method generates the initial population. The sorting technique for non-dominated solutions is used to sort the members of the population. In selection phase, a binary operator tournament is adopted. In order to generate new individuals, the two point operator is applied in the crossover phase. Elitism is performed to update the next generation. Experimental results showed that the proposed GA outperformed Strength Pareto Evolutionary Algorithm II (SPEA-II) by Zitzler, Laumanns, and Tzhiele (2001). Gong, Han, and Sun (2018) developed an artificial bee colony (ABC) algorithm for the $F_m|lot\text{-}str,block|C_{max}+E_{max}$. In the proposed multi-objective ABC (MOABC), initial population is obtained by the application of variants of NEH and MME and random method. In the employed bee phase, MOABC uses crossover or mutation operators to generate new off-springs with a certain probability. A construction/destruction operator is performed on scout bee stage. To improve the individuals in the onlooker bee stage, a pareto local search is applied to p_i individuals generated in employed bee phase. Elitism method is used to select food sources to update the old population. Computational results showed that the proposed MOABC performed better than other evaluated algorithms designed for the flow shop lot-streaming scheduling problem literature. Lebbar et al. (2018) proposed a MILP for the $F_m|block,r_i|\alpha C_{max}+(\alpha\text{-}1)T_{max}$. Experiment results showed that, overall, the proposed model is computationally avaricious to solve the considering problem. Lebbar et al. (2018) proposed an improved version of Non-Dominated Sorting Genetic Algorithm II (NDGAII) for the $F_m|block|C_{max}$ and T_{max} problem. The proposed NDGAII uses NEH-WPT and random method to generate the initial population. Binary tournament, two point and shift change operators are applied in selection, crossover and mutation phases, respectively. Elitism is used to update population. Experimental results showed that the proposed NDGAII can provide more competitive non-dominated solutions than HDDE by Qian et al. (2009). Shao, Pi, and Shao (2018a) developed an IWO for the $F_m|block,r_i|C_{max}+\sum_i T_i$ problem. In the proposed multi-objective discrete IWO (MODIWO), initial population is composed by weeds generated by PFT and FPD. In reproduction phase, a reference-based line is performed to generate new seeds. In dispersal spatial phase, d insertions are carried out in the current seed, where d is a parameter to be tuned. In competitive exclusion phase, the fast non-dominated sorting strategy and crowding distance technique, presented by Deb et al. (2002), are applied to determine the next generation. Self-adaptation phase performs a local search to the seeds to try to improve their adaptability. Experimental evaluations showed that the proposed MODIWO outperformed the state-of-art algorithms for multi-objective scheduling problems. Wang et al. (2018) proposed a multi-objective Parallel Variable Neighborhood Search (MOVNS) for the $F_m|block|C_{max}+E_W$. A modified version of NEH is used to generate the initial solution of MOVNS. Insertion and swap neighborhood local searches are adopted to be part of the structure of the VNS. Additionally, an insertion-based local search is proposed to compose the algorithm. Computational results showed that MPVNS outperformed NSGA-II by Deb et al. (2002) and BMSA by Lin and Ying (2013). Yang and Liu (2018) considered the m-machine blocking flow shop with fuzzy processing times and fuzzy due dates to minimize the fuzzy makespan and maximize the average agreement index. The problem can be denoted as $F_m|block|\tilde{C}_{max}+\widetilde{AI}$, where AI indicates the degree of the compliance between $\tilde{C}_i$ and $\tilde{d}_i$. Grey wolf optimizer is a population metaheuristic that tries to mimic social hierarchy of wolves. In the proposed HGWO, an n-dimensional vector that represents the position of a gray wolf represents a permutation. A modified version of PFE and random method generate the initial population. Three leaders are selected according to their fitness value-based to guided the remaining population. Two local searches based on insertion are performed to improve each gray wolf. Computational results showed that the proposed HGWO outperformed NSGAII by Deb et al. (2002). Han et al. (2019) proposed an evolutionary algorithm for the $F_m|brkdwn,block,lot\text{-}str|C_{max}$ and T_{max} problem. In the proposed evolutionary algorithm, called evolutionary robust scheduling algorithm (REMO), a variant of NEH and random method are used to generate the initial population. Two improved crossover operators based on similar block order crossover and artificial chromosome similar job order crossover are proposed. Insertion or swap is randomly selected to perform mutation. Experimental results showed that the proposed REMO outperformed the main metaheuristics of the literature.

3.5. Stochastic objective functions

In stochastic scheduling problems, processing times are assumed to obey a known probability distribution, and under the circumstances, the stochastic processing times can be converted into a deterministic counterpart. Table 7 summarizes publications that adopted stochastic measures, which corresponds to 6% of the covered papers. Pinedo (1982) provided a rule to scheduling jobs to solve the $F_m|block|E(C_{max})$ problem. Based on developed theorems, the authors recommended scheduling jobs with smaller expected processing times and larger variances in the processing times toward the beginning and the end of the sequence and schedule jobs with longer expected processing times and smaller variances in the processing times toward the middle of the sequence. Foley and Suresh (1984) showed that the Shortest Expected Processing Times rule minimizes the $F_2|block|E(\sum_i C_i)$. Wie and Pinedo (1986) showed complexity problem results of the $F_m|block|E(C_{max})$ and $F_m|block|E(\sum_i C_i)$ problems. The authors considered that processing time of job i on each one of the m machines is equal to the random variable X_j and it is distributed according to F_j. Thus, processing time are stochastically ordered, such that, $F_1 \leq \ldots \leq F_n$. The authors proved that sequence $J_1 J_3 J_5,\ldots J_{n-1} J_n J_{n-2},\ldots J_6 J_4 J_2$ when n is even and the sequence $J_1 J_3 J_5,\ldots J_{n-2} J_n J_{n-1},\ldots J_6 J_4 J_2$ when n is odd minimizes $E(C_{max})$. The authors also prove that sequence $J_1,\ldots J_n$ minimizes the $E(\sum_i C_i)$. Jia (1998) presented some results for the $F_2|block|E(\sum_{i=1}^{n}\sum_{j=1}^{n}|w_{[i]}^2 - w_{[j]}^2|)$. The author aimed to determine the optimal sequence that minimizes the expected total absolute differences in waiting times until processing is started on the second machine. The authors proved that when n is odd, the optimal sequence that minimizes the objective function is $J_n J_{n-2} J_3,\ldots J_1 J_2,\ldots J_{n-3} J_{n-1}$ and its reverse. When n is even, sequence $J_n J_{n-2},\ldots J_2 J_1 J_3,\ldots J_{n-3} J_{n-1}$ minimizes the objective function. Kalczynski and Kamburowski (2005) provided some complexity results of the $F_2|block|E(C_{max})$ problem. The authors showed that the problem is equivalent to TSP on a permuted Monge matrix. The authors also identified a new class of efficiently solvable cases. Hu et al. (2008) proposed a differential evolution with optimal computing budget allocation (OCBA) and test hypothesis (OHTDE) for the $F_m|b_{j,j+1}|E(C_{max})$ problem. In DE, a special structure of crossover, based on interchange moves, are incorporated in order to enhance its exploitation ability. OCBA is applied to allocate limited sampling budgets to provide evaluation and identification for good solutions and the hypothesis test is used to estimate the solution performance and to reduce the repeated search. Computational results showed the effectiveness of the adopted mechanisms. Han et al. (2016) proposed a multi-objective evolutionary optimization algorithm (MOEA) for the $F_m|block,lot\text{-}str|E(C_{max}) + E[var(T_{max})]$ problem, where var corresponds to the variance. The problem is formulated as a multi-objective optimization problem, where each interval objective is converted into a real-valued one using a dynamically weighted sum of its midpoint and radius. The proposed MOEA is based on the selection mechanism of NSGA-II. The initial population is generated by a modified version of MME and random method. The authors proposed a crossover operator, which considers non-dominated solutions information and differences among the parents performing crossover. In mutation, a local search based on insertion neighborhood is applied. Each mutant generated by the local search is evaluated by the distance from its parents. Computational results showed that the proposed MOEA outperformed the improved version of NSGA-II by Deb et al. (2002) and Pareto Based EDA (PBEDA) by Tiwari et al. (2015). Han et al. (2019) proposed a Migrating Birds Optimization (MBO) algorithm for the $F_m|block, lot-str|E(\bar{C}_{max}) + E[var(C_{max})]$, where $\bar{C}_{max}$ is the mean C_{max} of n jobs and var corresponds to the variance. MBO is a swarm-based metaheuristic which tries to imitate the V flight formation of birds and consists of four main parts: a) Initialization, which aims to generate the initial population; b) improving the leading solution; c) improving the remaining solutions and; d) replacing the leading solution. In the proposed modified MBO (MOBMO), solutions are generated through a modified version of PFE and random method. Insertion swap or conduct inverse is randomly selected and applied to the leading sequence. A probabilistic model is developed and through its application, solutions are sampled and generated. Additionally, a reference-point-assisted local search, which is based on insertion local search, is proposed to a random selected solution of the population. Computational results showed that MOBMO outperformed NSGA-II by Deb et al. (2002), the basic version of MOB, evolutionary multi-objective robust scheduling algorithm by Han et al. (2016), modified Multi-Objective Evolutionary Algorithm based on Decomposition by Shen, Han, and Fu (2017) and hybrid Multi-Objective Evolutionary Algorithm by Fu et al. (2018).

4. Results and discussions

Motivated by the results shown in Section 3, this section presents some quantitative metrics of the efforts carried out from the first paper found in 1969 up to the most recent paper found so far (May 2019). The analysis of the results discussed here are similar to the quantitative analysis carried out by Mutlu and Yagmahan (2014) and Rossit et al. (2018). First, Fig. 4 shows the distribution of papers according to the performance measure adopted. As evidenced, most of publication adopted blocking-machine problems with classic objective functions.

Completion time-based mono-objective functions are predominant in comparison with other performance measures. Inside this category, makespan (C_{max}) is the most studied mono-objective function. The adoption of total flow time, due date-based objective, cycle time and multi-objective functions are significantly less. Fig. 5 evidences the distribution of the papers according to the objective of the paper.

Papers that proposed constructive heuristics to generate an input for another algorithm were labeled inside the “metaheuristics and local searches” category. Publications that exclusively dealt with the development of constructive heuristics was categorized with this label. Exact methods included papers that developed MILP, branch-and-bound and bounded dynamic programming. Finally, the category “Analysis of the problem” included the literature reviews and papers that describe the complexity of blocking problems. Papers whose objective were the conception of solution methods corresponded to 92% of the total covered. In the relation between objective functions and solution methods, 74% of papers proposed algorithms to solve completion time-based measures, being C_{max} and total flow time-based objectives corresponding to 63% and 11%, respectively. The remaining of the papers corresponds to 12%, 8%, 5% and 1%, with multi-objective, tardiness based, stochastic based and cycle time objective-functions, respectively. Because of the complexity of the flow shop scheduling problem, the predominance of heuristic methods proposition is observed, mainly, metaheuristics and local searches. Table 8 shows the summarization of the papers according to the solution method adopted. Regarding 129 papers that proposed solution methods, the most used metaheuristics were Genetic Algorithm (9%), Artificial Bee Colony (5%), Differential Evolution (6%), Tabu Search (5%), Variable Search Neighborhood (5%), Particle Swarm Optimization (5%), Harmony Search (4%) and Iterated Greedy (3%). Although there are several methods to solve the problem, the performance of metaheuristics and local searches are dependent of a series of parameters. Thus, a design of experiments to define the parameters values that generates the best performance of the algorithm is

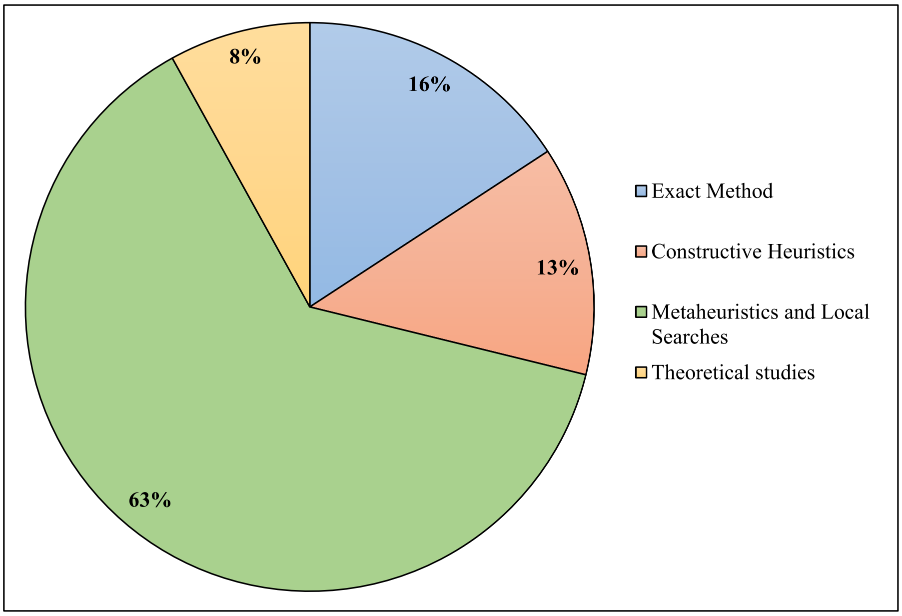

Fig. 4. Percentage of papers according to the objective of the paper.

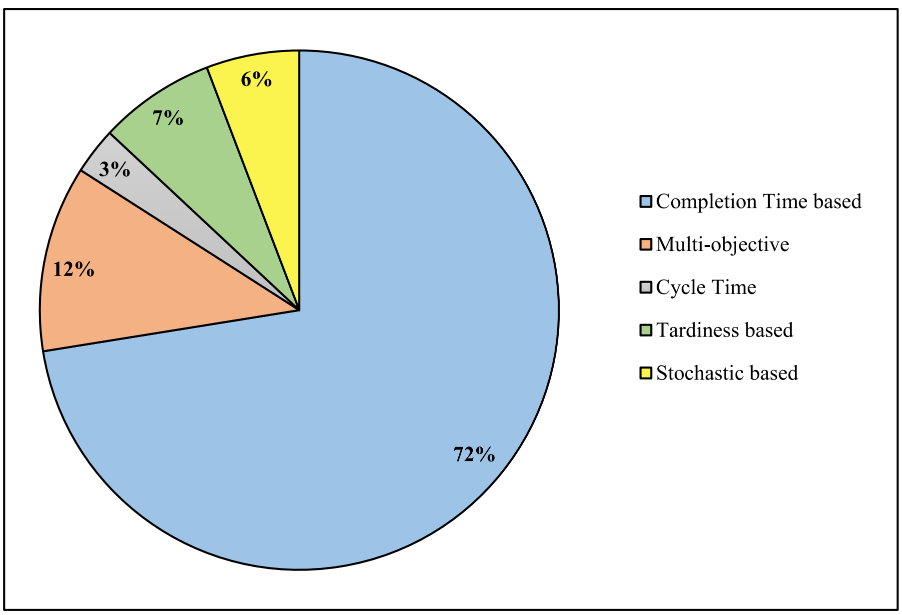

Fig. 5. Percentage of papers according to the objective-function adopted.

required. Another factor that affects the metaheuristic performance is the initial solution. In most publications, constructive heuristics and local searches have been used to generate an input for the improvement heuristic. Papers that dealt exclusively with the development of constructive heuristics correspond to 15% of 129 papers that proposed solution methods. Most of the constructive heuristics used PF and NEH insertion-schemes as mechanisms to build solutions, as in Ronconi (2004), Wang et al. (2010), Pan and Wang (2012), Han et al. (2013), Ribas and Companys (2015), Tasgetiren et al. (2017) and Takano and Nagano (2019). In most of the papers, authors established new criterions for the selection of the first job and new indexes to evaluate the next job to be inserted. The adoption of both mechanisms as part of the proposed method have been effective to generate solutions with high quality in reasonable computational time. Regarding the exact methods, which correspond to 17% of the 129 papers, branch-and-bound is the most adopted technique, followed by bounded dynamic programming and MILP.

Fig. 6 evidences that blocking flow shop scheduling problems have gained more attention mainly from 2000 on. Until 1995, the mainly focus of the papers were concentrated in the *m*-machine flow shop with *block* condition and minimization of makespan or cycle time. In this range (up to 1995), the literature are divided into analysis of problems and proposition of exact methods and constructive heuristics. Since the survey of Hall and Sriskandarajah (1996), the minimization of other objective-functions than makespan has grown significantly from 2008 on. Over 33% of the 53 papers included in 2008–2013 proposed solution methods for other objective methods. The proportion has grown in the range 2014–2019, since 42% of the 50 papers adopted other non-makespan objective functions. The proposition of different solution approaches also has modified throughout the years. The

Table 8
Summarization of papers according to the adopted approach.

Approaches	Problem	References
Exact methods		
B&B	$F_m\|block\|C_{max}$	(Levner, 1969); (Suhami & Mah, 1981); (Ronconi, 2005); (Companys & Mateo, 2007); (Toumi et al., 2017);
	$F_2\|block, \alpha p_{ij}\|C_{max}$	(Lee et al., 2010)
	$F_m\|block\|\sum_i C_i$	(Toumi et al. 2013c); (Toumi et al., 2013d)
	$F_m\|block\|\sum_i T_i$	(Ronconi & Armentano, 2001); (Toumi et al., 2013a)
	$F_m\|block\|\sum_i W_i T_i$	(Toumi et al., 2013b)
	$F_m\|permu,\beta_i\|\sum_i C_i$	(Vo et al., 2015)
MILP	$3AF\|block,s_{i,k,j}\|\alpha(\sum_i W_i C_i/W)+(\alpha-1)C_{max}$	(Maleki-Darounkolaei et al., 2012)
	$F_m\|block\|C_{max}$ and $F_m\|block,st_{sd}\|C_{max}$	(Pitty & Karimi, 2008)
	$F_m\|$ RCB,block, RCb*$\|C_{max}$	(Trabelsi et al., 2011)
	$F_m\|block,r_i\|\alpha C_{max} + (\alpha\text{-}1)T_{max}$	(Lebbar et al., 2018)
MILP and B&B	$F_m\|block, s_{i,j,k}\|C_{max}$	(Moslehi & Khorasanian, 2013)
	$F_m\|block\|\sum_i C_i$	(Takano & Nagano, 2017)
BDP	$F_m\|block\|C_{max}$	(Bautista et al., 2012); (Ozolins, 2019)
	$F_m\|block\|\sum_i C_i$	(Li et al., 2013)
GGA	$F_2\|block\|C_{max}$	(Reddi & Ramamoorthy, 1972)
Constructive Heuristics		
HDP	$F_2\|b_{j,j+1}\|C_{max}$	(Dutta & Cunningham, 1975)
	$F_m\|block\|E(C_{max})$	(Pinedo, 1982)
	$F_2\|block\|E(\sum_i C_i)$	(Foley & Suresh, 1984)
	$F_m\|block\|E(C_{max})$ and $F_m\|block\|E(\sum_i C_i)$	(Wie & Pinedo, 1986)
	$F_m\|block\|CC$	(McCormick et al., 1989); (Abadi et al., 2000); (Abadi, 2007)
	$F_m\|b_{j,j+1}\|C_{max}$	(Leinsten, 1990)
	$F_2\|block, st_{ant}\|C_{max}$	(Logendran & Sriskandarajah, 1993)
		(Levner et al., 1995)
	$F_2\|block\|E(\sum_{i=1}^{n}\sum_{j=1}^{n}\|w_{[i]}^2-w_{[j]}^2\|)$	(Jia, 1998)
	$F_m\|block\|C_{max}$	(Ronconi, 2004); (Companys et al., 2010a); (Pan & Wang, 2012)
	$F_m\|prmu, no\text{-}wait, block,b_{j,j+1}\|C_{max}$	(Liu & Kozan, 2009)
	$F_2\|no\text{-}wait, block, batch, sharedst\|C_{max}$	(Gong et al., 2010)
		(Fernandez-viagas et al., 2016)
	$F_m\|block,ord\|C_{max}$	(Koulamas & Panwalkar, 2018)
	$F_m\|block, st_{sd}\|C_{max}$	(Takano & Nagano, 2019)
Metaheuristics and local searches		
GA	$F_m\|block\|C_{max}$	(Caraffa et al., 2001); (Chowdhury et al., 2013)
	$F_m\|block, t_j\|C_{max}$	(Carlier et al., 2010)
	$F_m\|RCb\|C_{max}$	(Sauer & Sauvey, 2012)
	$F_m\|prmu,block, RCb^*, RCb\|C_{max}$	(Trabelsi et al., 2012)
	$F_m\|b_{j,j+1}\|C_{max}$	(Wang et al., 2006)
	$F_m\|block\|\sum_i T_i$	(Januario, Arroyo, & Moreira, 2009)
	$F_m\|b_{j,j+1}\|\sum_i \bar{C}_{icont} + \sum_i \bar{C}_{iparts} + \sum_i \sigma_i$	(Prasad et al., 2006)
	$F_m\|b_{j,j+1}, lot-str\|\sum_i (T_i + E_i)$	(Ventura & Yoon, 2013)
	$F_2\|batch(1), block, p_{i,j}^{min} \le p_{i,j} \le p_{i,j}^{max}\|C_{max}$	(Al-Salem & Kharbeche, 2016)
	$F_m\|block\|C_{max}+\sum_i C_i$	(Nouri & Ladhari, 2017)
	$F_m\|block\|C_{max}$ and T_{max}	(Lebbar et al., 2018)
GA and ABC	$F_m\|block\|C_{max}$, $F_m\|block\|\sum_i C_i$ and $F_m\|block\|\sum_i T_i$	(Nouri & Ladhari, 2015)
GA, TS and SA	$F_m\|block,r_i\|\sum_i C_i^3$	(Ren et al., 2018)
TS	$F_m\|block\|C_{max}$	(Grabowski & Pempera, 2007)
	$F_m\|prmu,no\text{-}wait, block\|C_{max}$	(Grabowski & Pempera, 2000)
	$F_m\|b_{j,j+1}, st_{sd}\|C_{max}$	(Norman, 1999)
	$F_m\|b_{j,j+1}\|C_{max}$	(Nowicki, 1999); (Li & Tang, 2005)
	$F_m\|block\|\sum_i C_i/n$	(Weng, 2000)
	$F_m\|block\|\sum_i T_i$	(Armentano & Ronconi, 2000)
EDA	$F_m\|block\|C_{max}$	(Jarboui et al., 2009); (Shao, Pi, & Shao, 2018b)
SA	$F_m\|block\|C_{max}$	(Companys et al., 2010b)
DE	$F_m\|block\|C_{max}$	(Wang et al., 2010); (Davendra et al., 2012); (Tasgetiren et al., 2015)
	$F_2\|block, batch, r_j\|C_{max}$	(Chen et al., 2014)
	$F_m\|b_{j,j+1}\|C_{max}$	(Qian et al. 2009), (Pan et al., 2011)
	$F_m\|b_{j,j+1}\|C_{max}$ and T_{max} and $F_m\|b_{j,j+1}\|C_{max}$ and $\sum_j I_j$	(Qian et al., 2009)
	$F_m\|b_{j,j+1}\|E(C_{max})$	(Hu et al., 2008)
PSO	$F_m\|block\|C_{max}$	(Liang et al., 2011); (Wang & Tang, 2012); (Eddaly et al., 2016)
	$F_2\|b_{j,j+1}, batch, st_{si}\|C_{max}$	(Tang & Tang, 2011)
	$F_m\|block\|\sum_i T_i$	(Nouha & Talel, 2016)
	$F_m\|b_{j,j+1}\|C_{max}$	(Liu et al., 2008); (Zhao et al., 2014)
IG	$F_m\|block\|C_{max}$	(Ribas et al., 2011); (Ding et al., 2015); (Tasgetiren et al., 2017)
	$F_m\|block\|\sum_i C_i$	(Khorasanian & Moslehi, 2012)
HS	$F_m\|block\|C_{max}$	(Duan et al., 2010); (Bao et al., 2012); (Wang et al., 2011)
	$F_m\|b_{j,j+1}\|C_{max}$	(Pan et al., 2011)
	$F_m\|block\|\sum_i C_i$	(Wang et al., 2010)
ABC	$F_m\|block\|C_{max}$	(Han et al., 2012); (Han et al., 2015); (Khorramizadeh & Riahi, 2015)
	$F_m\|block\|\sum_i C_i$	(Deng et al., 2012); (Han et al., 2013); (Ribas et al., 2015)
	$F_m\|lot\text{-}str,block\|C_{max} + E_{max}$	(Gong, Han, & Sun, 2018)

(continued on next page)

Table 8 (*continued*)

Approaches	Problem	References
SA	$F_m\|block\|C_{max}$	(Wang et al., 2012)
SOMA	$F_m\|block\|C_{max}$	(Davendra & Bialic-Davendra, 2013); (Davendra et al., 2013)
AIS	$F_m\|block\|C_{max}$	(Lin & Ying, 2013)
MMA	$F_m\|block\|C_{max}$	(Pan et al., 2013)
VNS	$F_m\|block\|C_{max}$	(Ribas & Companys, 2013); (Liu, Li, & Ren, 2015); (Zhang et al., 2015)
	$F_2\|block,mtflx_{(1)},prmp\|C_{max}$	(Khorasanian & Moslehi, 2017)
	$F_m\|b_{j,j+1}\|C_{max}$	(Moslehi & Khorasanian, 2014)
	$F_m\|block\|C_{max}+\ \mathrm{E_W}$	(Wang et al., 2018)
	$F_m\|block\|C_{max}$	(Sadaqa & Moraga, 2015)
SSA	$F_m\|block\|C_{max}$	(Davendra et al., 2014)
	$F_m\|prmu,block,\ RSb,\ RCb^{*},\ RCb\|C_{max}$	(Riahi et al., 2017)
CuckooSA	$F_m\|block\|C_{max}$	(Wang et al., 2015)
FFO	$F_m\|block\|C_{max}$	(Han et al., 2016)
BBO	$F_m\|block\|C_{max}$	(Liu, Wang, & Zhang, 2018)
CRO	$F_m\|block,\ \alpha p_{ij}\|C_{max}$	(Fu et al., 2017)
WWO	$F_m\|block,\ st_{sd}\|C_{max}$	(Shao, Pi, & Shao, 2018)
CGLS	$F_m\|\mathrm{st_{sd}},RCb,\ RSb\|C_{max}$	(Riahi et al., 2018)
GRASP	$F_m\|block\|\sum_i C_i$	(Ribas & Companys, 2015)
	$F_m\|block\|\sum_i T_i$	(Ronconi & Henriques, 2009)
VBIH	$F_m\|block\|\sum_i C_i$	(Tasgetiren et al., 2016)
ILS	$F_m\|block\|\sum_i T_i$	(Ribas et al., 2013)
ECS	$F_m\|block\|\sum_i T_i$	(Nagano et al., 2017)
IWO	$F_m\|block\|\sum_i T_i$	(Shao et al., 2017)
	$F_m\|block,r_i\|C_{max}+\sum_i T_i$	(Shao, Pi, & Shao, 2018a)
GSO	$F_m\|block\|C_{max}+\sum_i C_i$	(Guanlong et al., 2016)
MOEA	$F_m\|block,lot\text{-}str\|E(C_{max})+E[var(T_{max})]$	(Han et al., 2016)
GWO	$F_m\|block\|\tilde{C}_{max}+\widetilde{AI}$	(Yang & Liu, 2018)
MBO	$F_m\|block,\ lot-str\|E(\bar{C}_{max})+E[var(C_{max})]$	(Han et al., 2019)
REMO	$F_m\|brkdwn,block,\ lot\text{-}str\|C_{max}$ and T_{max}	(Han et al., 2019)
Other results		
CS	$F_2\|b_{j,j+1}\|C_{max}$	(Papadimitriou & Kanellakis, 1978)
CS	$F_m\|block\|CC$	(McCormick et al., 1987)
CS	$F_3\|no\text{-}wait(1{,}2),\ block(2,\ 3)\|CC$ and $F_3\|block\|CC$	(Kamoun & Sriskandarajah, 1993)
Survey	General problems with blocking	(Hall & Sriskandarajah, 1996)
CS	$F_2\|block\|E(C_{max})$	(Kalczynski & Kamburowski, 2005)
CS	$F_2\rightarrow D\|block,\ v,c_e,F_i\|C_{max}$	(Soukhal et al. 2005)
CS	$F_m\|block,RCb\|C_{max}$	(Martinez et al., 2006)
CS	$F_2\rightarrow D\|block,\ v,c_e,\ F_i\|C_{max}$	(Yuan et al., 2007)
Objective function study	$F_m\|block\|C_{max}$ and CR	(Cohn et al., 2010)
Review	General problems with blocking	(Emmons & Vairaktarakis, 2013)
New flow shop schematic	$F_m\|block\|C_{max}$	(Panwalkar & Koulamas, 2018)

*B&B – Branch-and-bound; MILP – Mixed Integer Linear Programming; BDP – Bounded Dynamic Programming; HDP – Heuristic Dynamic Programming; GA – Genetic Algorithm; ABC – Artificial Bee Colony; TS – Tabu Search; EDA – Estimation Distribution Algorithm; DE – Differential Evolution; PSO – Particle Swarm Optimization; IG – Iterated Greedy; HS – Harmony Search; SA – Simulated Annealing; SOMA – Self-Organism Migratin Algorithm; AIS – Artificial Immune System; MMA – Memetic Algorithm; VNS – Variable Neighborhood Search; CuckooSA - Cuckoo Search Algorithm; FFO – Fruit Fly Optimization; BBO - Biogeography-Based Optimization; CRO – Chemical Reaction Optimization; WWO – Water Wave Optimization; CGLS – Constraint Guided Local Search; GRASP - Greedy Randomized Adaptive Search Procedure; VBIH – Variable Block Insertion Heuristic; ILS – Iterated Local Search; ECS – Evolutionary Clustering Search; IWO – Invasive Weed Optimization; GSO - Group Search Optimization; MOEA – Multi-objective Evolutionary Algorithm; MBO - Migrating Birds Optimization; GWO – Grey Wolf Optimization; CS – Complexity Study.

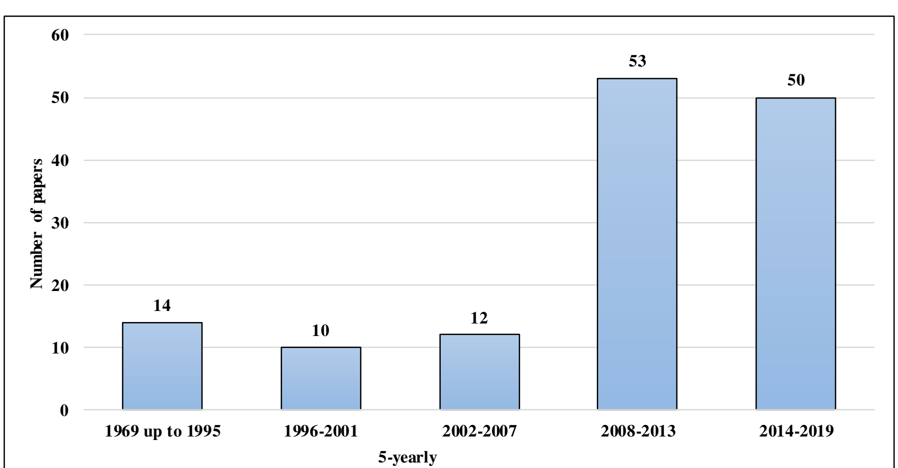

Fig. 6. Number of papers in accordance with 5-yearly.

development of metaheuristics started with proposition of TS for the $F_m|b_{j,j+1},\ st_{sd}|C_{max}$ problem (Norman, 1999). In the ranges 1996–2001, 2002–2007, 2008–2013 and 2014–2019, the percentage of papers that proposed metaheuristics were 60%, 33%, 70% and 82%, respectively. Exact methods are more expressive in the range 1969 up to 1995. In the remaining ranges, papers which proposed only exact methods or constructive heuristics remained constant, with the exception of range 2002–2007.

Finally, approximately 26% of the 139 papers adopted more than only one additional constraint. The quantity of papers that adopt multiple constraints is more expressive from 2008 on, specially, in the range 2014–2019. Setup times was the constraint most adopted by the papers, although the combination of different buffer conditions (no-wait jobs, without blocking, RCb, RCb* and *block*) in the same flow shop has grown significantly. In general, heuristic methods were proposed to solve these kind of problems, with exception of Martinez et al. (2006) and Lee et al. (2010).

4.1. Speed-up methods and upper bounds

This subsection gathered papers that developed speed up methods for local searches and updated upper bounds of instance problems used in common. First, for the $F_m|block|C_{max}$ problem, Wang et al. (2010) proposed a speed-up technology method for the insertion neighborhood search with $(n-1)^2$ sequences generated. The procedure can reduce the computational time from $O(mn^3)$ to $O(mn^2)$. Based on Li et al. (2009), Fernandez-viagas et al. (2016) developed a speed-up procedure for the $F_m|block|\sum_i C_i$ problem for insertion neighborhood search. Tasgetiren et al. (2016) also proposed a speed-up procedure for $F_m|block|\sum_i C_i$ for insertion and exchange pair-wise neighborhood local searches. Both procedures can reduce the computation time from 40% to 50%.

Finally, papers that updated upper bounds of common adopted instance problems are listed as follows. The classical dataset instance problems of Taillard (1993) is one of the most used for both $F_m|block|C_{max}$ and $F_m|block|\sum_i C_i$ problems and consists of 120 problems, with the number of jobs assuming $n=\{20, 50, 100, 200, 500\}$ and the number of machines $m=\{5, 10, 20\}$. Each combination of $n \times m$ is composed by 10 problems. The best solutions found for the $F_m|block|C_{max}$ problem can be seen in Grabowski and Pempera (2007), Jarboui et al. (2009), Wang et al. (2010), Ribas et al. (2011), Davendra et al. (2012), Wang and Tang (2012), Wang et al. (2012), Davendra and Bialic-Davendra (2013), Lin and Ying (2013), Pan et al. (2013), Ribas and Companys (2013), Ding et al. (2015), Han et al. (2015), Eddaly et al. (2016), Han et al. (2016), Tasgetiren et al. (2017) and Shao et al. (2017). For the $F_m|block|\sum_i C_i$ problem, the best solutions found for the Taillard's data instance problems can be seen in Wang et al. (2010), Deng et al. (2012), Khorasanian and Moslehi (2012), Ribas and Companys (2015), Ribas et al. (2015) and Tasgetiren et al. (2016). The benchmark instance problems used in Ronconi and Henriques (2009) for the $F_m|block|\sum_i T_i$ problem is also adopted by Ribas et al. (2011), Toumi et al. (2013b), Nagano et al. (2017) and Shao et al. (2017).

5. Conclusions and directions for future researches

This paper reviewed and classified 139 papers that deal with the m-machine flow shop scheduling problem with blocking occurrences. In general, completion time-based measures are the most studied performance measures, being target of 73% of the papers. Regarding the approach, proposition of solution methods is the focus of 92% of papers. From the 139 papers covered, 63% of the papers aimed the proposition of metaheuristics, 16% exact methods, 14% constructive heuristics and 11% presented analysis of the problem and reviews of literature. Since Hall and Sriskandarajah (1996) literature review, the attention of the researchers has been predominant in the conception of solution methods for the m-machine flow shop with *block* condition. As blocking flow shop scheduling problem is a real condition in practice applications, the proposition of solution methods (exact or heuristic) is an important area of study and continues to be promising in the future. Based on the above results, some suggestions and directions for future researches according to the gaps identified by the quantitative analysis of the results in Section 4 are described as follows:

- Proposition of solution methods for the m-machine flow shop scheduling problem with RCb and RCb* constraints to minimize completion time-based and due date-based measures.
- Adoption of multi-objective functions for the m-machine flow shop with RCb and RCb* constraints.
- Addition of more than one constraint to the problem as setup times, deterioration jobs, different buffer conditions (RCb, RCb*, no-wait, without blocking and *block*) for total flow time-based and due date-based minimization.

Credit authorship contribution statement

Hugo Hissashi Miyata: Conceptualization, Data curation, Formal analysis, Writing - review & editing. **Marcelo Seido Nagano:** Conceptualization, Formal analysis, Data curation, Writing - review & editing.

Conflict of Interest

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - Brazil under grant number 306075/2017-2 and 430137/2018-4 and by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Brazil, finance code 001.

References

Abadi, I. N. K. (2007). A new algorithm for minimizing makespan, C(max), in blocking flow-shop problem through slowing down the operations. *Journal of the Operational Research Society, 58*(1), 134–140. https://doi.org/10.1057/palgrave.jors.2602099.

Abadi, I. N. K., Hall, N. G., & Sriskandarajah, C. (2000). Minimizing cycle time in a blocking flowshop. *Operations Research, 48*(1), 177–180. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-0033685522&partnerID=tZOtx3y1.

Al-Salem, M., & Kharbeche, M. (2016). Throughput optimization for the robotic cell problem with controllable processing times. *RAIRO - Operations Research, 51*(3), 805–818. https://doi.org/10.1051/ro/2016064.

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research, 246*(2), 345–378. https://doi.org/10.1016/j.ejor.2015.04.004.

Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research, 187*(3), 985–1032. https://doi.org/10.1016/j.ejor.2006.06.060.

Armentano, V. A., & Ronconi, D. P. (2000). Minimização do tempo total de atraso no problema de flowshop com buffer zero através de busca tabu. *Gestão & Produção, 7*(3), 352–363. https://doi.org/10.1590/S0104-530x2000000300011.

Bao, Y., Jing, H., & Zheng, L. (2011). A hybrid based GA and HS algorithms in blocking flow shop scheduling problems. In *2011 2nd international conference on mechanic automation and control engineering, MACE 2011 - proceedings* (pp. 953–956). Hohhot: IEEE. https://doi.org/10.1109/MACE.2011.5987089.

Bao, Y., Zheng, L., & Jiang, H. (2012). An improved HS algorithms for the blocking flow shop scheduling problems. In *2012 international conference on computer science and information processing (CSIP)* (pp. 1289–1291). Shaanxi, China: IEEE. https://doi.org/10.1109/CSIP.2012.6309097.

Bautista, J., Cano, A., Companys, R., & Ribas, I. (2012). Solving the Fm|block|Cmax problem using bounded dynamic programming. *Engineering Applications of Artificial Intelligence, 25*(6), 1235–1245. https://doi.org/10.1016/j.engappai.2011.09.001.

Caraffa, V., Ianes, S., Bagchi, T. P., Sriskandarajah, C. P., Bagchi, T., & Sriskandarajah, C. (2001). Minimizing makespan in a blocking flowshop using genetic algorithms. *International Journal of Production Economics, 70*(2), 101–115. https://doi.org/10.1016/S0925-5273(99)00104-8.

Carlier, J., Haouari, M., Kharbeche, M., & Moukrim, A. (2010). An optimization-based heuristic for the robotic cell problem. *European Journal of Operational Research, 202*(3), 636–645. https://doi.org/10.1016/j.ejor.2009.06.035.

Chen, H., Zhou, S., Li, X., & Xu, R. (2014). A hybrid differential evolution algorithm for a two-stage flow shop on batch processing machines with arbitrary release times and blocking. *International Journal of Production Research, 52*(19), 5714–5734. https://doi.org/10.1080/00207543.2014.910625.

Chowdhury, A., Ghosh, A., & Sinha, S. (2013). A novel genetic algorithm to solve travelling salesman problem and blocking flow shop scheduling problem. *International Journal Bio-Inspired Computation, 5*(5), 303–314.

Cohn, A., Kurnaz, S., Guan, Y., & Jiang, Y. (2010). Trading off between makespan and customer responsiveness in flow shop sequencing. *International Journal of Production Research, 48*(22), 6777–6797. https://doi.org/10.1080/00207540903280531.

Companys, R. (1966). Métodos heurísticos en la resolución del problema del taller mecánico. *Estudios Empresariales, 5*, 7–18.

Companys, R., & Mateo, M. (2007). Different behaviour of a double branch-and-bound algorithm on Fm | prmu | Cmax and Fm | block | Cmax problems. *Computers and Operations Research, 34*(4), 938–953. https://doi.org/10.1016/j.cor.2005.05.018.

Companys, R., Ribas, I., & Mateo, M. (2010a). Improvement tools for NEH based heuristics on permutation and blocking flow shop scheduling problems. In B. Vallespir, & T. Alix (Eds.). In *Advances in production management systems: New challenges, new approaches: 338* (pp. 33–40).

Companys, R., Ribas, I., & Mateo, M. (2010b). Note on the behaviour of an improvement heuristic on permutation and blocking flow-shop scheduling. *International Journal of Manufacturing Technology and Management, 20*(1/2/3/4), 331. https://doi.org/10.1504/ijmtm.2010.032905.

Davendra, D., & Bialic-Davendra, M. (2013). Scheduling flow shops with blocking using a discrete self-organising migrating algorithm. *International Journal of Production Research, 51*(8), 2200–2218. https://doi.org/10.1080/00207543.2012.711968.

Davendra, D., Bialic-Davendra, M., Senkerik, R., & Pluhacek, M. (2013). Scheduling the flow shop with blocking problem with the chaos-induced discrete self organising migrating algorithm. In W. Rekdalsbakken, R. T. Bye, & H. Zhang (Eds.), *Proceedings 27th european conference on modelling and simulation ECMS 2013*.

Davendra, D., Senkerik, R., Zelinka, I., & Pluhacek, M. (2014). Scatter Search Algorithm with chaos based stochasticity. In *Proceedings of the 2014 IEEE congress on evolutionary computation, CEC 2014* (pp. 860–866). https://doi.org/10.1109/CEC.2014.6900273.

Davendra, D., Zelinka, I., Bialic-Davendra, M., Senkerik, R., & Jasek, R. (2012). Clustered enhanced differential evolution for the blocking flow shop scheduling problem. *Central European Journal of Operations Research, 20*(4), 679–717. https://doi.org/10.1007/s10100-011-0198-3.

Deb, K., Member, A., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Deng, G., Xu, Z., & Gu, X. (2012). A discrete artificial bee colony algorithm for minimizing the total flow time in the blocking flow shop scheduling. *Chinese Journal of Chemical Engineering, 20*(6), 1067–1073.

Ding, J.-Y., Song, S., Gupta, J. N. D., Wang, C., Zhang, R., & Wu, C. (2015). New block properties for flowshop scheduling with blocking and their application in an iterated greedy algorithm. *International Journal of Production Research, 7543*(September), 1–14. https://doi.org/10.1080/00207543.2015.1076941.

Duan, J. H., Pan, Q. K., Gao, K. Z., Li, J., & Bao, Z. (2010). Scheduling the blocking flow shop problem using a harmony search algorithm. In *Proceedings - 2010 6th international conference on natural computation, ICNC 2010: 8* (pp. 4263–4266). IEEE. https://doi.org/10.1109/ICNC.2010.5583649.

Dutta, S. K., & Cunningham, A. A. (1975). Sequencing two-machine flow-shops with finite intermediate storage. *Management Science, 21*(9), 989–996.

Eddaly, M., Jarboui, B., & Siarry, P. (2016). Combinatorial particle swarm optimization for solving blocking flowshop scheduling problem. *Journal of Computational Design and Engineering, 3*(4), 295–311. https://doi.org/10.1016/j.jcde.2016.05.001.

Emmons, H., & Vairaktarakis, G. (2013). Blocking or limited buffers in flow shops. In *Flow shop scheduling: theoretical results, algorithms, and applications* (pp. 221–245). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4614-5152-5_7.

Fernandez-viagas, V., Leisten, R., & Framinan, J. M. (2016). A computational evaluation of constructive and improvement heuristics for the blocking flow shop to minimise total flowtime. *Expert Systems With Applications, 61*, 290–301. https://doi.org/10.1016/j.eswa.2016.05.040.

Foley, R. D., & Suresh, S. (1984). Minimizing the expected flowtime in stochastic flow shops. *IIE Transactions (Institute of Industrial Engineers), 16*(4), 391–395. https://doi.org/10.1080/07408178408975260.

Fu, Y., Wang, H., Tian, G., Li, Z., & Hu, H. (2018). Two-agent stochastic flow shop deteriorating scheduling via a hybrid multi-objective evolutionary algorithm. *Journal of Intelligent Manufacturing*. 2014 https://doi.org/10.1007/s10845-017-1385-4.

Fu, Y., Wang, Z. Z., Zhang, J., & Wang, Z. Z. (2017). A blocking flow shop deteriorating scheduling problem via a hybrid chemical reaction optimization. *Advances in Mechanical Engineering, 9*(6), 1–10. https://doi.org/10.1177/1687814017701371.

Gendreau, M., Hertz, A., & Laporte, G. (1992). New insertion and postoptimization procedures for the travelling salesman problem. *Operations Research, 40*(6), 1086–1094.

Gilmore, P. C., & Gomory, R. E. (1964). Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. *Operations Research*. https://doi.org/10.1287/opre.12.5.655.

Gong, D., Han, Y., & Sun, J. (2018). Knowledge-base d systems a novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems. *Knowledge-Based Systems, 148*, 115–130. https://doi.org/10.1016/j.knosys.2018.02.029.

Gong, H., & Tang, L. (2008). Two-machine flowshop scheduling problem with transportation and blocking features. In *2008 chinese control and decision conference (CCDC 2008)* (pp. 1471–1474). Yantai, China: IEEE.

Gong, H., Tang, L., & Duin, C. W. (2010). A two-stage flow shop scheduling problem on a batching machine and a discrete machine with blocking and shared setup times. *Computers & Operations Research, 37*(5), 960–969. https://doi.org/10.1016/j.cor.2009.08.001.

Grabowski, J., & Pempera, J. (2000). Sequencing of jobs in some production system. *European Journal of Operational Research, 125*(3), 535–550. https://doi.org/10.1016/S0377-2217(99)00224-6.

Grabowski, J., & Pempera, J. (2007). The permutation flow shop problem with blocking. A tabu search approach. *Omega-International Journal of Management Science, 35*(3), 302–311. https://doi.org/10.1016/j.omega.2005.07.004.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*. https://doi.org/10.1016/S0167-5060(08)70356-X.

Guanlong, D., Shuning, Z., & Mei, Z. (2016). A discrete group search optimizer for blocking flow shop multi-objective scheduling. *Advances in Mechanical Engineering, 8*(8), 1–9. https://doi.org/10.1177/1687814016664262.

Gupta, J. N. D., & Stafford, E. F., Jr (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research, 169*, 699–711. https://doi.org/10.1016/j.ejor.2005.02.001.

Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research, 44*(3), 510–525. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-0030134958&partnerID=tZOtx3y1.

Han, Y.-Y., Duan, J.-H., Yang, Y.-J., Min, Z., & Bao, Y. (2012). Minimizing the total flowtime flowshop with blocking using a discrete artificial bee colony. *Advanced Intelligent Computing Theories and Applications: With Aspects of Artificial Intelligence, 6839*, 91–97.

Han, Y.-Y., Duan, J.-H., & Zhang, M. (2011). Apply the discrete artificial bee colony algorithm to the blocking flow shop problem with makespan criterion. In *2011 chinese control and decision conference: 1-6* (pp. 2131–2135).

Han, Y.-Y., Gong, D., & Sun, X. (2015). A discrete artificial bee colony algorithm incorporating differential evolution for the flow-shop scheduling problem with blocking. *Engineering Optimization, 47*(7), 927–946. https://doi.org/10.1080/0305215x.2014.928817.

Han, Y.-Y., Pan, Q.-K., Li, J.-Q., & Sang, H. (2012). An improved artificial bee colony algorithm for the blocking flowshop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 60*(9–12), 1149–1159. https://doi.org/10.1007/s00170-011-3680-0.

Han, Y., Gong, D., Jin, Y., & Pan, Q. (2016). Evolutionary multi-objective blocking lot-streaming flow shop scheduling with interval processing time. *Applied Soft Computing Journal, 42*, 229–245. https://doi.org/10.1016/j.asoc.2016.01.033.

Han, Y., Gong, D., Jin, Y., & Pan, Q. (2019). Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns. *IEEE Transactions on Cybernetics, 49*(1), 184–197. https://doi.org/10.1109/TCYB.2017.2771213.

Han, Y., Gong, D., Li, J., & Zhang, Y. (2016). Solving the blocking flow shop scheduling problem with makespan using a modified fruit fly optimisation algorithm. *International Journal of Production Research, 54*(May), 6782–6797. https://doi.org/10.1080/00207543.2016.1177671.

Han, Y., Li, J., Gong, D., & Sang, H. (2019). Multi-objective migrating birds optimization algorithm for stochastic lot-streaming flow shop scheduling with blocking. *IEEE Access, 7*, 5946–5962. https://doi.org/10.1109/ACCESS.2018.2889373.

Han, Y. Y., Liang, J. J., Pan, Q. K., Li, J. Q., Sang, H. Y., & Cao, N. N. (2013). Effective hybrid discrete artificial bee colony algorithms for the total flowtime minimization in the blocking flowshop problem. *International Journal of Advanced Manufacturing Technology, 67*(1–4), 397–414. https://doi.org/10.1007/s00170-012-4493-5.

Hu, R., Wang, L., Qian, B., & Huang, F. Z. (2008). Differential evolution method for stochastic flow shop scheduling with limited buffers. In *2008 IEEE congress on evolutionary computation, CEC 2008: 7* (pp. 1295–1301). https://doi.org/10.1109/CEC.2008.4630963.

Ishibuchi, H., Yoshida, T., & Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation, 7*(2), 204–223. https://doi.org/10.1109/TEVC.2003.810752.

Januario, T., de, O., Arroyo, J. E. C., & Moreira, M. C. O. (2009). Genetic algorithm for tardiness minimization in flowshop with blocking. In N. Krasnogor, MB. Melián-Batista, J. A. M. Pérez, J. M. Moreno-Vega, & D. A. Pelta (Eds.), *Nature inspired cooperative strategies for optimization (NICSO 2008)* (pp. 153–164). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-03211-0_13.

Jarboui, B., Eddaly, M., Siarry, P., & Rebaï, A. (2009). An estimation of distribution algorithm for minimizing the makespan in blocking flowshop scheduling problems. In U. K. Chakraborty (Ed.), *Computational intelligence in flow shop and job* *shop scheduling* (pp. 151–167). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02836-6_5.

Jia, C. (1998). Minimizing variation in stochastic flow shop. *Operations Research Letters, 23*(3–5), 109–111. https://doi.org/10.1016/s0167-6377(98)00039-x.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly, 1*(1), 61–68.

Kalczynski, P. J. A. N., & Kamburowski, J. (2005). Two-machine stochastic flow shops with blocking and the travelling salesman problem. *Journal of Scheduling, 8*(1993), 529–536.

Kamoun, H., & Sriskandarajah, C. (1993). The complexity of scheduling jobs in repetitive manufacturing systems. *European Journal of Operational Research, 70*(3), 350–364. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-0027906646&partnerID=tZOtx3y1.

Karabulut, K. (2016). A hybrid iterated greedy algorithm for total tardiness minimization in permutation flowshops. *Computers and Industrial Engineering, 98*, 300–307. https://doi.org/10.1016/j.cie.2016.06.012.

Kellegöz, T., Toklu, B., & Wilson, J. (2010). Elite guided steady-state genetic algorithm for minimizing total tardiness in flowshops. *Computers and Industrial Engineering, 58*(2), 300–306. https://doi.org/10.1016/j.cie.2009.11.001.

Khorasanian, D., & Moslehi, G. (2012). An iterated Greedy algorithm for solving the blocking flow shop scheduling problem with total flow time criteria. *International Journal of Industrial Engineering & Production Research, 23*(4), 301–308.

Khorasanian, Danial;, & Moslehi, G (2017). Two-machine flow shop scheduling problem with blocking, multi-task flexibility of the first machine, and preemption. *Computers and Operations Research, 79*(September 2016), 94–108. https://doi.org/10.1016/j.cor.2016.09.023.

Khorramizadeh, M., & Riahi, V. (2015). A bee colony optimization approach for mixed blocking constraints flow shop scheduling problems. *Mathematical Problems in Engineering*, 1–10. *2015* https://doi.org/10.1155/2015/612604.

Koulamas, C., & Panwalkar, S. S. (2018). New index priority rules for no-wait flow shops. *Computers and Industrial Engineering, 115*(November 2017), 647–652. https://doi.org/10.1016/j.cie.2017.12.015.

Lageweg, B. J., Lenstra, J. K., & Kan, A. H. G. R. (1978). A general bounding scheme for the permutation flow-shop problem. *Operations Research, 26*(1), 53–67.

Lebbar, G., Barkany, A. E., Ikram, E. A., & Moumen, D. A. (2018). Solving the multi objective flow shop scheduling problems using an improved NSGA-II. *International Journal of Operations and Quantitative Management, 24*(3), 211–230.

Lebbar, G., El Abbassi, A., Jabri, A., El Barkani, A., & Darcherif, M. (2018). Multi - criteria blocking flow shop scheduling problems : Formulation and performance analysis. *Advances in Production Engineering & Management, 13*(2), 136–146.

Lee, W. C., Shiau, Y. R., Chen, S. K., & Wu, C. C. (2010). A two-machine flowshop scheduling problem with deteriorating jobs and blocking. *International Journal of Production Economics, 124*(1), 188–197. https://doi.org/10.1016/j.ijpe.2009.11.001.

Leinsten, R. (1990). Flowshop sequencing problems with limited buffer storage. *International Journal of Production Research, 28*(11), 2085–2100.

Levner, E., Kogan, K., & Maimon, O. (1995). Flowshop scheduling of robotic cells with job-dependent transportation and set-up effects. *Journal of the Operational Research Society, 46*, 1447–1455.

Levner, E. V. (1969). Optimal planning of parts machining on a number of machines. *Automatic Remote Control, 12*, 1972–1978.

Li, S. H., & Tang, L. X. (2005). A tabu search algorithm based on new block properties and speed-up method for permutation flow-shop with finite intermediate storage. *Journal of Intelligent Manufacturing, 16*(4–5), 463–477. https://doi.org/10.1007/s10845-005-1658-1.

Li, X., Wang, Q., & Wu, C. (2009). Efficient composite heuristics for total flowtime minimization in permutation flow shops. *Omega, 37*(1), 155–164.

Li, Y., Wang, S., Ding, L., & Xie, X. (2013). A dynamic programming based heuristic in Max-algebra for solving a blocking flow-shop problem. In *Proceedings of 2013 2nd international conference on measurement, information and control* (pp. 375–379). IEEE. https://doi.org/10.1109/MIC.2013.6757986.

Liang, J. J., Pan, Q.-K., Chen, T., & Wang, L. (2011). Solving the blocking flow shop scheduling problem by a dynamic multi-swarm particle swarm optimizer. *International Journal of Advanced Manufacturing Technology, 55*(5–8), 755–762. https://doi.org/10.1007/s00170-010-3111-7.

Lin, S. W., & Ying, K. C. (2013). Minimizing makespan in a blocking flowshop using a revised artificial immune system algorithm. *Omega (United Kingdom), 41*(2), 383–389. https://doi.org/10.1016/j.omega.2012.03.006.

Lin, S., & Ying, K. (2013). Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm. *Computers & Operations Research, 40*(6), 1625–1647. https://doi.org/10.1016/j.cor.2011.08.009.

Liu, B., Wang, L., & Jin, Y. H. (2008). An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research, 35*(9), 2791–2806. https://doi.org/10.1016/j.cor.2006.12.013.

Liu, S. Q., & Kozan, E. (2009). Scheduling a flow shop with combined buffer conditions. *International Journal of Production Economics, 117*(2), 371–380. https://doi.org/10.1016/j.ijpe.2008.11.007.

Liu, S., Wang, P., & Zhang, J. (2018). An improved biogeography-based optimization algorithm for blocking flow shop scheduling problem. *Chinese Journal of Electronics, 27*(2), 351–358. https://doi.org/10.1049/cje.2018.01.007.

Liu, X., Li, K., & Ren, H. (2015). A hybrid algorithm for the permutation flowshop scheduling problem without intermediate buffers. *Discrete Dinyamics in Nature and Society 2015*.

Logendran, R., & Sriskandarajah, C. (1993). Two-machine group scheduling problem with blocking and anticipatory setups. *European Journal of Operational Research, 69*(3), 467–481. https://doi.org/10.1016/0377-2217(93)90029-M.

Maccarthy, B. L., & Liu, J. (1993). Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling. *International Journal of Production Research, 31*(1), 59–79. https://doi.org/10.1080/00207549308956713.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*, 1567–1579. https://doi.org/10.1016/j.amc.2006.11.033.

Maleki-Darounkolaei, A., Modiri, M., Tavakkoli-Moghaddam, R., & Seyyedi, I. (2012). A three-stage assembly flow shop scheduling problem with blocking and sequence-dependent set up times. *Journal of Industrial Engineering International, 8*(26), 603–622. https://doi.org/10.1186/2251-712X-8-26.

Manjeshwar, P. K., Damodaran, P., & Srihari, K. (2009). Minimizing makespan in a flow shop with two batch-processing machines using simulated annealing. *Robotics and Computer-Integrated Manufacturing, 25*, 667–679. https://doi.org/10.1016/j.rcim.2008.05.003.

Manjeshwar, P. K., Damodaran, P., & Srihari, K. (2011). Genetic algorithms for minimizing makespan in a flow shop with two capacitated batch processing machines. *International Journal of Advanced Manufacturing Technology, 55*, 1171–1182. https://doi.org/10.1007/s00170-010-3150-0.

Martinez, S. (2005). *Ordonnancement de systèmes de production avec contraintes de blocage*. France: IRCCyN, Nntes.

Martinez, S., Dauzère-Pérès, S., Guéret, C., Mati, Y., & Sauer, N. (2006). Complexity of flowshop scheduling problems with a new blocking constraint. *European Journal of Operational Research, 169*(3), 855–864. https://doi.org/10.1016/j.ejor.2004.08.046.

Martinez, S., Guéret, C., & Sauer, N. (2004). A simulated annealing method for the flow-shop with a new blocking constraint. In *20th European conference on operational research (EURO'04)* (p. 27).

McCormick, S. T., Pinedo, M. L., Shenker, S., & Wolf, B. (1989). Sequencing in an assembly line with blocking to minimize cycle time. *Operations Research, 37*(6), 925–935. https://doi.org/10.1287/opre.37.6.925.

McCormick, S. T., Pinedo, M. L., & Wolf, B. (1987). The complexity of scheduling a flexible assembly line with finite buffers. *IE/OR Technical Report*. New York.

Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*(1), 1097–1100.

Moslehi, G., & Khorasanian, D. (2013). Optimizing blocking flow shop scheduling problem with total completion time criterion. *Computers & Operations Research, 40*(7), 1874–1883. https://doi.org/10.1016/j.cor.2013.02.003.

Moslehi, G., & Khorasanian, D. (2014). A hybrid variable neighborhood search algorithm for solving the limited-buffer permutation flow shop scheduling problem with the makespan criterion. *Computers and Operations Research, 52*, 260–268. https://doi.org/10.1016/j.cor.2013.09.014.

Mutlu, M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem : Literature review, classification and current trends. *Omega, 45*, 119–135. https://doi.org/10.1016/j.omega.2013.07.004.

Nagano, M. S., & Moccellin, J. V. (2002). A high quality solution constructive heuristic for flow shop sequencing. *Journal of the Operational Research Society, 53*, 1374–1379. https://doi.org/10.1057/palgrave.jors.2601466.

Nagano, M. S., Komesu, A. S., & Miyata, H. H. (2017). An evolutionary clustering search for the total tardiness blocking flow shop problem. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-017-1358-7.

Nawaz, M., Enscore, E. E., Jr, & Ham, I. (1983). A Heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega-International Journal of Management Science, 11*(1), 91–95. https://doi.org/10.1016/0305-04838390088-9.

Norman, B. A. (1999). Scheduling flowshops with finite buffers and sequence-dependent setup times. *Computers and Industrial Engineering, 36*(1), 163–177. https://doi.org/10.1016/S0360-8352(99)00007-8.

Nouha, N., & Talel, L. (2016). A particle swarm optimization metaheuristic for the blocking flow shop scheduling problem: Total tardiness minimization. In M. Rovatsos, G. Vouros, & V. Julian (Eds.), *Multi-agent systems and agreement technologies* (pp. 145–153). Cham: Springer International Publishing.

Nouri, N., & Ladhari, T. (2015). Minimizing regular objectives for blocking permutation flow shop scheduling : Heuristic approaches. In *GECCO '15 proceedings of the 2015 annual conference on genetic and evolutionary computation* (pp. 441–448). Madrid, Spain: ACM New York. https://doi.org/10.1145/2739480.2754638.

Nouri, N., & Ladhari, T. (2017). Evolutionary multiobjective optimization for the multi-machine flow shop scheduling problem under blocking. *Annals of Operations Research*, 1–18. https://doi.org/10.1007/s10479-017-2465-8.

Nowicki, E. (1999). The permutation flow shop with buffers : A tabu search approach. *European Journal of Industrial Engineering, 116*, 205–219.

Ogbu, F. A., & SMith, D. K. (1990). The application of the simulated annealing algorithm to the solution the n/m/ Cmax flowshop problem. *Computers & Operations Research, 17*(3), 243–253.

Omran, M. G. H. (2008). Global-best harmony search. *Applied Mathematics and Computation, 198*, 643–656. https://doi.org/10.1016/j.amc.2007.09.004.

Ozolins, A. (2019). Improved bounded dynamic programming algorithm for solving the blocking flow shop problem. *Central European Journal of Operations Research, 27*(1), 15–38. https://doi.org/10.1007/s10100-017-0488-5.

Pan, Q.-K., Wang, L., & Gao, L. (2011). A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. *Applied Soft Computing, 11*(8), 5270–5280. https://doi.org/10.1016/j.asoc.2011.05.033.

Pan, Q., Fatih, M., & Liang, Y. (2008). A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering, 55*(4), 61–82. https://doi.org/10.1016/j.cie.2008.03.003.

Pan, Q. K., & Wang, L. (2012). Effective heuristics for the blocking flowshop scheduling problem with makespan minimization. *Omega, 40*(2), 218–229. https://doi.org/10.1016/j.omega.2011.06.002.

Pan, Q. K., Wang, L., Gao, L., & Li, W. D. (2011). An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers. *Information Sciences, 181*(3), 668–685. https://doi.org/10.1016/j.ins.2010.10.009.

Pan, Q., Wang, L., Sang, H., Li, J., & Liu, M. (2013). A high performing memetic algorithm for the flowshop scheduling problem with blocking. *IEEE Transactions on Automation Science and Engineering, 10*(3), 741–756. https://doi.org/10.1109/TASE.2012.2219860.

Panwalkar, S. S., & Koulamas, C. (2018). The evolution of schematic representations of flow shop scheduling problems. *Journal of Scheduling*. https://doi.org/10.1007/s10951-018-0594-9.

Papadimitriou, C. H., & Kanellakis, P. C. (1978). Flowshop scheduling with limited temporary storage. In *Proceedings - annual Allerton conference on communication, control, and computing* (pp. 214–223). Retrieved from. http://www.scopus.com/inward/record.url?eid=2-s2.0-0018210671&partnerID=tZOtx3y1.

Pinedo, M. L. (1982). Minimizing expected makespan in stochastic flow shops. *Operations Research, 30*(1), 148–162. https://doi.org/10.2307/1427030.

Pinedo, M. L. (1995). *Scheduling theory, algorithms, and systems*: 53 (2nd ed). Nova York: Prentice Hall https://doi.org/10.1017/CBO9781107415324.004.

Pitty, S. S., & Karimi, I. A. (2008). Novel MILP models for scheduling permutation flowshops. *Chemical Product and Process Modeling, 3*(1). https://doi.org/10.2202/1934-2659.1176.

Ponnambalam, S. G., Jawahar, N., & Chandrasekaran, S. (2009). Discrete particle swarm optimization algorithm for flowshop scheduling. In A. Lazinica (Ed.), *Particle swarm optimization*. InTech Retrieved from http://www.intechopen.com/books/particle_swarm_optimization/discrete_particle_swarm_optimization_algorith%0Am_for_flowshop_scheduling.

Pour, H. D. (2001). A new heuristic for the n -job, m -machine ow-shop problem. *Production Planning & Control, 12*(January), 648–653. https://doi.org/10.1080/09537280152582995.

Prasad, S. D., Rajendran, C., & Chetty, O. V. K. (2006). A genetic algorithmic approach to multi-objective scheduling in a Kanban-controlled flowshop with intermediate buffer and transport constraints. *International Journal of Advanced Manufacturing Technology, 29*(5), 564–576. https://doi.org/10.1007/BF02729110.

Qian, B., Wang, L., Huang, D. X., & Wang, X. (2009). An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers. *International Journal of Production Research, 47*(1), 1–24. https://doi.org/10.1080/00207540701528750.

Qian, B., Wang, L., Huang, D., Wang, W., & Wang, X. (2009). An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. *Computers & Operations Research, 36*, 209–233. https://doi.org/10.1016/j.cor.2007.08.007.

Rajendran, C., & Ziegler, H. (1997). An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs. *European Journal of Operational Research, 103*(1), 129–138.

Rameshkumar, K., Suresh, R. K., & Mohanasundaram, K. M. (2005). Discrete particle swarm optimization (DPSO) algorithm for permutation flowshop scheduling to minimize makespan. *Advances in natural computation. ICNC 2005*. Berlin: Springer, Berling, Heidelberg. Lecture notes in computer science https://doi.org/10.1007/11539902_70.

Reddi, S. S., & Ramamoorthy, C. V. (1972). On the flow-shop sequencing with no wait in process. *Operations Research Quarterly, 23*(3), 323–328.

Ren, T., Liang, J., Liu, Y. Y., Fu, Y., Cui, X. Z., & Hu, Q. B. (2018). Blocking flowshop scheduling model for optimising cubic criterion with release dates. *IEEE Access, 6*(2011), 72179–72188. https://doi.org/10.1109/ACCESS.2018.2881960.

Riahi, V., Newton, M. A. H., Su, K., & Sattar, A. (2018). Local search for flowshops with setup times and blocking constraints. In *Proceedings international conference on automated planning and scheduling, ICAPS: 2018-June* (pp. 199–207).

Riahi, V., Khorramizadeh, M., Hakim Newton, M. A., & Sattar, A. (2017). Scatter search for mixed blocking flowshop scheduling. *Expert Systems with Applications, 79*, 20–32. https://doi.org/10.1016/j.eswa.2017.02.027.

Ribas, I., & Companys, R. (2013). A competitive variable neighbourhood search algorithm for the blocking flow shop problem. *European Journal of Industrial Engineering, 7*(6), 729–754. https://doi.org/10.1504/ejie.2013.058392.

Ribas, I., & Companys, R. (2015). Efficient heuristic algorithms for the blocking flow shop scheduling problem with total flow time minimization. *Computers and Industrial Engineering, 87*, 30–39. https://doi.org/10.1016/j.cie.2015.04.013.

Ribas, I., Companys, R., & Tort-Martorell, X. (2011). An iterated greedy algorithm for the flowshop scheduling problem with blocking. *Omega, 39*(3), 293–301. https://doi.org/10.1016/j.omega.2010.07.007.

Ribas, I., Companys, R., & Tort-Martorell, X. (2013). An efficient iterated local search algorithm for the total tardiness blocking flow shop problem. *International Journal of Production Research, 51*(17), 5238–5252. https://doi.org/10.1080/00207543.2013.802390.

Ribas, I., Companys, R., & Tort-Martorell, X. (2015). An efficient discrete artificial bee colony algorithm for the blocking flow shop problem with total flowtime minimization. *Expert Systems with Applications, 42*(15–16), 37–41. https://doi.org/10.1016/j.eswa.2015.03.026.

Röck, H., Rock, H., Röck, H., & Roeck, H. (1984). Some new results in flow shop scheduling. *ZOR. Zeitschrift Fur Operations-Research, 28*(1), 1–16. https://doi.org/10.1007/BF01919082.

Ronconi, D. P., & Armentano, V. A. (2001). Lower bounding schemes for flowshops with blocking in-process. *Journal of the Operational Research Society*. https://doi.org/10.1057/palgrave.jors.2601220.

Ronconi, Débora P. (2004). A note on constructive heuristics for the flowshop problem with blocking. *International Journal of Production Economics, 87*(1), 39–48. https://doi.org/10.1016/S0925-5273(03)00065-3.

Ronconi, Débora P. (2005). A branch-and-bound algorithm to minimize the makespan in a flowshop with blocking. *Annals of Operations Research, 138*, 53–65.

Ronconi, D. P., & Henriques, L. R. S. (2009). Some heuristic algorithms for total tardiness minimization in a flowshop with blocking. *Omega, 37*(2), 272–281. https://doi.org/10.1016/j.omega.2007.01.003.

Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The non-permutation flow-shop scheduling problem : A literature review. *Omega, 77*, 143–153. https://doi.org/10.1016/j.omega.2017.05.010.

Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research, 177*(3), 2033–2049. https://doi.org/10.1016/j.ejor.2005.12.009.

Ruiz, R., & Stützle, T. (2008). An iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research, 187*(3), 1143–1159. https://doi.org/10.1016/j.ejor.2006.07.029.

Ruiz, R., & Vázquez-rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research, 205*(1), 1–18. https://doi.org/10.1016/j.ejor.2009.09.024.

Sadaqa, M., & Moraga, R. J. (2015). Scheduling blocking flow shops using meta-RaPS. *Procedia Computer Science, 61*, 533–538. https://doi.org/10.1016/j.procs.2015.09.211.

Sauer, C., & Sauvey, N. (2012). A genetic algorithm with genes-association recognition for flowshop scheduling problems. *Journal of Intelligent Manufacturing, 23*, 1167–1177. https://doi.org/10.1007/s10845-010-0422-3.

Sauvey, C., & Sauer, N. (2009). Heuristique pour la résolution des problèmes d'ordonnancement de type flowshop avec blocage RCb. In *10ème conférence de la société française de recherche opérationnelle et d'aide à la décision (ROADEF'09)* (pp. 236–237).

Shao, Z., Pi, D., & Shao, W. (2017). Estimation of distribution algorithm with path relinking for the blocking flow-shop scheduling problem. *Engineering Optimization, 0*(0), 1–23. https://doi.org/10.1080/0305215X.2017.1353090.

Shao, Z., Pi, D., & Shao, W. (2017). Self-adaptive discrete invasive weed optimization for the blocking flow-shop scheduling problem to minimize total tardiness. *Computers & Industrial Engineering, 111*, 331–351. https://doi.org/10.1016/j.cie.2017.07.037.

Shao, Z., Pi, D., & Shao, W. (2018a). A multi-objective discrete invasive weed optimization for multi-objective blocking flow-shop scheduling problem. *Expert Systems with Applications, 113*, 77–99. https://doi.org/10.1016/j.eswa.2018.06.020.

Shao, Z., Pi, D., & Shao, W. (2018b). A novel discrete water wave optimization algorithm for blocking flow-shop scheduling problem with sequence-dependent setup times. *Swarm and Evolutionary Computation, 40*, 53–75. December 2017 https://doi.org/10.1016/j.swevo.2017.12.005.

Shen, J. N., Wang, L., & Wang, S. Y. (2015). A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion. *Knowledge-Based Systems, 74*, 167–175. https://doi.org/10.1016/j.knosys.2014.11.016.

Shen, X., Han, Y., & Fu, J. (2017). Robustness measures and robust scheduling for multi-objective stochastic flexible job shop scheduling problems. *Soft Computing, 21*(21), 6531–6554. https://doi.org/10.1007/s00500-016-2245-4.

Soukhal, A., Oulamara, A., & Martineau, P. (2005). Complexity of flow shop scheduling problems with transportation constraints. *European Journal of Operational Research, 161*(1), 32–41. https://doi.org/10.1016/j.ejor.2003.03.002.

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.

Suhami, I., & Mah, R. S. H. (1981). An implicit enumeration scheme for the flowshop problem with no intermediate storage. *Computers and Chemical Engineering, 5*(2), 83–91. https://doi.org/10.1016/0098-1354(81)87003-2.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research, 64*, 278–285.

Takano, M. I., & Nagano, M. S. (2017). A branch-and-bound method to minimize the makespan in a permutation flow shop with blocking and setup times. *Cogent Engineering, 4*(1), 1–16. https://doi.org/10.1080/23311916.2017.1389638.

Takano, M. I., & Nagano, M. S. (2019). Evaluating the performance of constructive heuristics for the blocking flow shop scheduling problem with setup times. *International Journal of Industrial Engineering Computations, 10*(1), 37–50. https://doi.org/10.5267/j.ijiec.2018.5.002.

Tang, Q., & Tang, L. X. (2011). Heuristic particle swarm optimization for batching and scheduling flowshop with sequence-independent setup time. *Applied Mechanics and Materials, 130–134*, 2504–2507. https://doi.org/10.4028/www.scientific.net/amm.130-134.2504.

Tasgetiren, M. F., Kizilay, D., Pan, Q., & Suganthan, P. N. (2017). Iterated greedy algorithms for the blocking flowshop scheduling problem with makespan criterion. *Computers and Operation Research, 77*, 111–126. https://doi.org/10.1016/j.cor.2016.07.002.

Tasgetiren, M. Fatih, Pan, Q. K., Kizilay, D., & Suer, G. (2015). A populated local search with differential evolution for blocking flowshop scheduling problem. In *2015 IEEE congress on evolutionary computation, CEC 2015 - proceedings* (pp. 2789–2796). https://doi.org/10.1109/CEC.2015.7257235.

Tasgetiren, M. F., Pan, Q. K., Suganthan, P. N., & Chen, A. H. L. (2011). A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Information Sciences, 181*(16), 3459–3475. https://doi.org/10.1016/j.ins.2011.04.018.

Tasgetiren, M. F., Sevkli, M., Liang, Y., & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for permutation flowshop sequencing problem. In *Lecture notes in computer science: 3172* (pp. 382–389). Berlin: Springer.

Tasgetiren, M. F., Pan, Q., Kizilay, D., & Gao, K. (2016). A variable block insertion heuristic for the blocking flowshop scheduling problem with total flowtime criterion. *Algorithms, 9*(4), 1–24. https://doi.org/10.3390/a9040071.

Tiwari, A., Chang, P.-C., Tiwari, M. K., & Kollanoor, N. J. (2015). A pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem. *International Journal of Production Research, 53*(3), 793–834. https://doi.org/10.1080/00207543.2014.933273.

Toumi, S., Jarboui, B., Eddaly, M., & Rebaï, A. (2017). Branch-and-bound algorithm for solving blocking flowshop scheduling problems with makespan criterion. *International Journal of Mathematics in Operational Research, 10*(1), 34–48. https://doi.org/10.1504/IJMOR.2017.080743.

Toumi, S., Jarboui, B., Eddaly, M., & Rebai, A. (2013a). An efficient branch and bound algorithm to solve the permutation flowshop scheduling problem with blocking constraints. In *2013 5th international conference on modeling, simulation and applied optimization, ICMSAO 2013* (pp. 1–6).

Toumi, S., Jarboui, B., Eddaly, M., & Rebai, A. (2013b). Lower bounds on the total tardiness and total weighted tardiness for scheduling flowshop with blocking. In *2013 international conference on advanced logistics and transport, ICALT 2013* (pp. 296–301).

Toumi, S., Jarboui, B., Eddaly, M., & Rebai, A. (2013c). New lower bounds for the blocking flowshop scheduling problem to minimize the total completion time criterion. In *2013 5th international conference on modeling, simulation and applied optimization (ICMSAO)* (pp. 1–5). IEEE.

Toumi, S., Jarboui, B., Eddaly, M., & Rebai, A. (2013d). Solving blocking flowshop scheduling problem with branch and bound algorithm. In *2013 international conference on control, decision and information technologies (CoDIT)* (pp. 746–749). IEEE.

Trabelsi, W., Sauvey, C., & Sauer, N. (2011). In *Complexity and mathematical model for flowshop problem subject to different types of blocking constraint. IFAC Proceedings Volumes (IFAC-PapersOnline): 44*. IFAC https://doi.org/10.3182/20110828-6-IT-1002.01887.

Trabelsi, W., Sauvey, C., & Sauer, N. (2012). Heuristics and metaheuristics for mixed blocking constraints flowshop scheduling problems. *Computers and Operations Research, 39*(11), 2520–2527. https://doi.org/10.1016/j.cor.2011.12.022.

Tseng, L.-Y., & Lin, Y.-T. (2010). A hybrid genetic algorithm for no-wait flowshop scheduling problem. *International Journal of Production Economics, 128*(1), 144–152. https://doi.org/10.1016/j.ijpe.2010.06.006.

Tseng, L., & Lin, Y. (2010). A genetic local search algorithm for minimizing total flowtime in the permutation flowshop scheduling problem. *International Journal of Production Economics, 127*(1), 121–128. https://doi.org/10.1016/j.ijpe.2010.05.003.

Van, L. E., Cummings, P. T., & Le van, D. (1990). Scheduling of serial multiproduct batch processes via simulated annealing. *Computers & Chemical Engineering, 14*(12), 1351–1362.

Ventura, J. A., & Yoon, S.-H. (2013). A new genetic algorithm for lot-streaming flow shop scheduling with limited capacity buffers. *Journal of Intel, 24*, 1185–1196. https://doi.org/10.1007/s10845-012-0650-9.

Vo, N. V., Fouillet, P., & Lenté, C. (2015). Elaboration of general lower bounds for the total completion time in flowshop scheduling problems through MaxPlus approach. In E. Pinson, F. Valente, & B. Vitoriano (Eds.), *Operations research and enterprise systems* (pp. 285–299). Cham: Springer International Publishing.

Wang, C., Song, S., Gupta, J. N. D., & Wu, C. (2012). A three-phase algorithm for flowshop scheduling with blocking to minimize makespan. *Computers and Operations Research, 39*(11), 2880–2887. https://doi.org/10.1016/j.cor.2012.02.020.

Wang, F., Deng, G., Jiang, T., & Zhang, S. (2018). Multi-objective parallel variable neighborhood search for energy consumption scheduling in blocking flow shops. *IEEE Access, 6*, 68686–68700. https://doi.org/10.1109/ACCESS.2018.2879600.

Wang, H., Wang, W., Sun, H., Li, C., Rahnamayan, S., & Liu, Y. (2015). A modified cuckoo search algorithm for flow shop scheduling problem with blocking. In *2015 IEEE congress on evolutionary computation (CEC)* (pp. 456–463). Sendai: IEEE. https://doi.org/10.1109/CEC.2015.7256925.

Wang, L., Pan, Q.-K., Fatih Tasgetiren, M., & Tasgetiren, M. F. (2010). Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms. *Expert Systems with Applications, 37*(12), 7929–7936. https://doi.org/10.1016/j.eswa.2010.04.042.

Wang, L., Pan, Q.-K., Suganthan, P. N. N., Wang, W.-H., & Wang, Y.-M. (2010). A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. *Computers & Operations Research, 37*(3), 509–520. https://doi.org/10.1016/j.cor.2008.12.004.

Wang, L., Pan, Q.-K., & Tasgetiren, M. F. (2011). A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem. *Computers & Industrial Engineering, 61*(1), 76–83. https://doi.org/10.1016/j.cie.2011.02.013.

Wang, L., Zhang, L., & Zheng, D. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations, 33*, 2960–2971. https://doi.org/10.1016/j.cor.2005.02.028.

Wang, X., & Tang, L. (2012). A discrete particle swarm optimization algorithm with self-adaptive diversity control for the permutation flowshop problem with blocking. *Applied Soft Computing Journal, 12*(2), 652–662. https://doi.org/10.1016/j.asoc.2011.09.021.

Weng, M. X. (2000). Scheduling flow shops with limited buffer spaces. In *Proceedings of the 2000 winter simulation conference* (pp. 1359–1363). Orlando: IEEE.

Widmer, M., & Hertz, A. (1989). A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research, 41*(2), 186–193. https://doi.org/10.1016/0377-2217(89)90383-4.

Wie, S., & Pinedo, M. (1986). On minimizing the expected makespan and flow time in stochastic flow shops with blocking. *Mathematics of Operations Research, 11*(2), 336–342.

Yang, Z., & Liu, C. (2018). A hybrid multi-objective gray wolf optimization algorithm for a fuzzy blocking flow shop scheduling problem. *Advances in Mechanical Engineering, 10*(3), 1–13. https://doi.org/10.1177/1687814018765535.

Yuan, J., Soukhal, A., Chen, Y., & Lu, L. (2007). A note on the complexity of flow shop scheduling with transportation constraints. *European Journal of Operational Research, 178*(3), 918–925. https://doi.org/10.1016/j.ejor.2005.08.034.

Yuan, K., & Sauer, N. (2007). Application of EM algorithm to flowshop scheduling problems with a special blocking. *IESM'07* Beijing.

Zhang, C., Xie, Z., Shao, X., & Tian, G. (2015). An effective VNSSA algorithm for the blocking flowshop scheduling problem with makespan minimization. In International conference on advanced mechatronic systems, ICAMechS, 2015-Octob (pp. 86–89). https://doi.org/10.1109/ICAMechS.2015.7287134.

Zhao, F., Tang, J., Wang, J., & Jonrinaldi, J. (2014). An improved particle swarm optimisation with a linearly decreasing disturbance term for flow shop scheduling with limited buffers. *International Journal of Computer Integrated Manufacturing, 27*(5), 488–499. https://doi.org/10.1080/0951192X.2013.814165.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPREA2. In *SPREA2: 103* (pp. 12–19). TIK-Report. https://doi.org/10.3929/ethz-a-010782581.