Operational Research in Engineering Sciences: Theory and Applications
Vol. 5, Issue 1, 2022, pp. 169-184
ISSN: 2620-1607
eISSN: 2620-1747
DOI: https://doi.org/10.31181/oresta250322166r

MATHEMATICAL MODELLING OF NON-PERMUTATION FLOW SHOP PROCESSES WITH LOT STREAMING IN THE SMART MANUFACTURING ERA

Daniel Alejandro Rossit [1,2]*, Fernando Tohmé [2,3], ,Rodrigo Introcaso [1], Jeanette Rodríguez [1]

[1] Engineering Department, Universidad Nacional del Sur, Bahía Blanca, Argentina
[2] INMABB, CONICET, Bahía Blanca, Argentina
[3] Economics Department, Universidad Nacional del Sur, Bahía Blanca, Argentina

Received: 26 October 2021
Accepted: 21 March 2022
First online: 25 March 2022

Research paper

Abstract: *Industry 4.0 is leveraging the production capabilities of the industry. The deep digitalization that Industry 4.0 promotes enables to extend control skills to an exhaustive detail in the shop floors. Then, new planning strategies can be designed and implemented. We present mathematical models to represent non-permutation flow shop processes, incorporating Industry 4.0 features and customer-focused attention. Basically, we study the impact of lot streaming on the ensuing optimization problems, since the work-in-process inventory control is considerably enhanced by Industry 4.0 technologies. Thus, is possible to take advantage of subdividing the production lots into smaller sublots, as lot streaming proposes. To test this hypothesis we use a novel approach to non-permutation flow shop problems which requires a lot streaming strategy, incorporating total tardiness as objective function. Our analysis indicates that lot streaming improves results increasingly with the number of machines. We also find that the improvement is less steep with more sublots, increasing the computational cost of solutions. This indicates that it is highly relevant to fine tune the maximum number of sublots to avoid extra costs.*

Key words: *Scheduling, Mathematical Modelling, Non-Permutation Flow Shop, Lot Streaming, Industry 4.0, Total Tardiness.*

1. Introduction

Manufacturing systems have changed substantially in the last decade by the increasing digitalization of productive processes (Xu et al. 2018). This increases the accessibility, through the so-called Cyber-Physical Systems (CPS), to information that

* Corresponding author.
daniel.rossit@uns.edu.ar (D. Rossit), ftohme@criba.edu.ar (F. Tohmé), rodrigointrocaso@gmail.com (R. Introcaso), jearodriguez.98@gmail.com (J. Rodriguez)

before remained confined inside the production machinery (Lee et al. 2015). With more access to information, often acquired in real time, it becomes possible to address more precisely decision problems that formerly could be only solved approximately (Dolgui et al. 2019). Thus, production planning processes can now be solved in a more efficient and integral way. Scheduling is one of the stages that will be more affected by the new technologies, since it is the last phase before starting the physical production (Ivanov et al. 2016; Bicakci & Kara 2019). Decision-making in scheduling involves solving NP hard problems, being thus at least as hard as any problem in which checking a solution requires polynomial time (Garey et al. 1976, Stanković et al. 2020). In this article we will focus on scheduling for non-permutation flow shop problems. Flow shop processes represent systems in which all the production orders are processed in the same sequence. That is, given a class J of n jobs (with j=1,2,…,n) and a set M of m machines (such tha i =1, 2, …, m), the operations on each job j follows the same sequence 1, 2, ..., m on machines. That is, the first operation on j will be carried out on machine 1, the second on machine 2, and so on until the last operation is carried out by m (Pinedo 2012). This is the production configuration applied by more than one quarter of the industries of the world (Pan et al. 2011).

Flow shop problems have been widely studied in the literature, but largely focusing on permutation sequences (Liao et al. 2006; Rossit et al. 2018). In those cases, a single ordering of the jobs is imposed over all the machines, i.e. on each machine i all the n jobs will be processed in the same order. For instance, given 4 jobs such that the processing sequence on the first machine is 2, 1, 3, 4, in the next machines the sequence will be the same (2,1, 3, 4). This condition does not respond to a production process rationale, since in general the machines can process the jobs in different sequences. The main reason for solving the problem restricted to permutation sequences is that the number of possible solutions is $n!$, while if this restriction is lifted, the number of possible cases raises to $n!^m$ (Potts et al. 1991). The general case, without the permutation constraint is that of non-permutation scheduling flow shop problems (NPFS). Note that the solutions to permutation scheduling problems constitute particular instances of NPFS solutions. The recent improvements in capacities for decision-making in production environments, makes the latter more treatable.

Nevertheless, to avoid the combinatorial explosion of seeking NPFS solutions, some strategies to reduce the search space are still needed. Our approach is to incorporate a technique that contributes to facilitate production activities, namely lot streaming (Trietsch & Baker 1993). In this treatment, the number of items to be produced by each job is partitioned such that each part is processed independently. Adding the lot streaming condition to flow shop problems has led to improved performances in the production processes (Sarin & Jaiprakash 2007; Cetinkaya & Duman 2021). Lot streaming does not require neither extra layouts nor new technologies (D'Amico et al. 2021), but demand more attention at the shop floor, since orders are now divided in several suborders. This division increases the demands on the information and control systems that have to keep track of more entities (Pan et al. 2011; Ferraro et al. 2019). It becomes thus interesting to analyze how this strategy may impact in the context of the new production environments where the information and control systems have been considerably enhanced. The implementation of lot streaming in non-permutation problems has not been widely analyzed, particularly when the focus is the quality of customer service. We analyze this problem in systems in which the compliance with the delivery date agreed on with the customer is the measure of the performance of the system.

The goal of this paper is to present new ways of addressing the problems of scheduling in the Industry 4.0 by focusing on the new challenges that the new paradigm poses for production planning processes. More specifically, this paper presents a novel MILP model for NPFS problems, where Total Tardiness is the objective function optimized by allowing lot streaming. This paper contributes to the literature on NPFS by presenting a concrete contribution, namely the introduction of new mathematical formulations and the ensuing results.

The rest of the paper is organized as follows. Section 2 introduces Industry 4.0 and decision making processes in that paradigm, and presents a brief NPFS literature review. Then, in Section 3, we develop new mathematical formulations, detailing their underlying assumptions. Section 4 presents and discusses the experimental design and the main results of our investigation.

2. Industry 4.0 concepts and Literature Review

In this section we review the relevant notions of Industry 4.0 needed for our analysis as well as the literature on lot streaming in non-permutation flow shop processes. Both issues become relevant in the last decade thanks to the technological advances that gave rise to the current fourth industrial revolution.

2.1. Industry 4.0 concepts

The main drivers of this revolution have been the Internet of Things (IoT) and Cyber-Physical Systems, which allow the connection among all the components in the shop floor, leading to the full digitalization of production. In this way, all the information generated in the production process becomes available to the different business functions of the firms (Xu et al. 2018; Dolgui et al. 2019). Figure 1 illustrates how different levels of decision-making, associated to the classical control of production structure ISA-95, are integrated by CPS.

The five levels of ISA-95 start at level 0, where the physical process of production is carried out (raw materials are transformed into end products). Next, level 1 is in charge of controlling the production tools, recording data as processing speed, temperatures of the tools and pieces, vibrations, etc. Level 2 incorporates control systems like PLC and SCADA, which can correct deviations in the production flow. At level 3 are the Manufacturing Execution Systems, in charge of production planning and quality control. At this level is where scheduling problems are solved and the compliance with the plan is monitored. Finally, the level 4, of Business Logistics Systems, takes care of the strategic decisions of the firm. CPS relate these systems by sharing their information among them, allowing its analysis in real time improving the global efficiency of decision-making (Lee et al. 2015; Grassi et al. 2020). This richness of information and the availability of powerful computing equipment at level 3 allow handling hard problems like NPFS.

Figure 1. ISA-95 levels associated to CPS.

2.2. Flow Shop Literature review and research gap

The literature on flow shop problems has a long history, starting with Johnson's first paper on the subject in 1954 (Johnson, 1954). While the largest part of that literature is centered on PFS, the branch devoted to NPFS is rich enough. A foundational result on these problems was published by Conway et al. (1967), which shows that when makespan is the objective function, permutation solutions are enough to yield the optimal schedule for up to 3 machines. In a much simpler way, this result had been stated already in (Johnson, 1954). This means that NPFS genuine solutions make sense for makespan maximization with more than 3 machines. Potts et al. (1991) studied instances in which NPFS solutions improve the makespan over PFS ones in

$\frac{1}{2}\sqrt{m}$. Rebaine (2005) analyzed the ratio of the makespans of NPFS and PFS solutions in the presence of delays in the operations, showing that even with 2 machines PFS solutions cannot ensure the optimal result. Rossit et al. (2018b) studied the critical paths of NPFS and PFS solutions for 2 jobs and m machines, while in (Rossit et al. 2021a) analyzed the processing times that allow PFS solutions to be better than NPFS ones, in the same case of 2 jobs and m machines. Besides these theoretical contributions there are many empirical studies that show that under different settings NPFS solutions improve on those of PFS (Tandon et al. 1991; Strusevich & Zwaneveld 1994; Koulamas 1998; Jain & Meeran 2002; Nagarajan & Sviridenko 2009; Rudek 2011; Rossi & Lanzetta 2014; Benavides & Ritt 2016; Benavides & Ritt 2018).

As shown in Table 1, in most of these works the objective function is makespan. Only a few ones consider alternative goals, as for instance those related to delivery dates (for a more exhaustive list, see Rossit et al. 2018a). Liao et al. (2006) present a key result analyzing several single-objective functions and comparing the PFS and NPFS solutions: they show that NPFS solutions improve upon PFS ones, even for delivery date-related objective functions. Ying et al. (2010) ran a similar analysis and found that in the cases of delivery date functions, NPFS solutions improve over PFS ones even more than in the case of completion time-related functions. This is consistent with the findings of Liao & Huang (2010), who show that for total tardiness, NPFS solutions are indeed better than PFS solutions.

Table 1. Main works related to Non-permutation flow shop scheduling. For further details see Rossit et al. 2018a.

Reference	NPFS	Lot streaming	Objective Function	Solution Approach
Potts et al. (1991)	✓	✓	Makespan	Exact
Tandon et al. (1991)	✓	x	Makespan	Heuristic
Strusevich & Zwaneveld (1994)	✓	x	Makespan	Exact
Koulamas (1998)	✓	x	Makespan	Heuristic
Jain & Meeran (2002)	✓	x	Makespan	Meta-Heuristic
Rebaine (2005)	✓	✓	Makespan	Exact
Liao et al. (2006)	✓	x	Total Tardiness (among others)	Meta-Heuristic
Nagarajan & Sviridenko (2009)	✓	x	Makespan	Exact
Liao & Huang (2010)	✓	x	Total Tardiness (among others)	Meta-Heuristic
Rudek (2011)	✓	x	Makespan	Exact
Ziaee (2013)	✓	x	Total weighted tardiness	Heuristic
Rossi & Lanzetta (2014)	✓	x	Makespan	Meta-Heuristic
Rossit et al. (2016)	✓	✓	Makespan	Exact
Benavides & Ritt (2016)	✓	x	Makespan	Heuristic
Rossit et al. (2018b)	✓	x	Makespan	Exact
Benavides & Ritt (2018)	✓	x	Makespan	Heuristic
Rossit et al. (2021a)	✓	x	Makespan	Exact
Rossit et al. (2021b)	✓	x	Total Tardiness	Meta-Heuristic
CURRENT STUDY	✓	✓	Total Tardiness	Exact

Ziaee (2013) addressed NPFS with setup times depending on the schedule, under the goal of minimizing the Total Weighted Tardiness, by applying a two-stage method. The first stage yields a permutation solution while in the second stage a non-permutation local search improves it. Rossit et al. (2021b) studied NPFS problems in Industry 4.0 environments with missing operations, optimizing total tardiness, showing that NPFS solutions improved over PFS ones in average, in 98% of the cases. This indicates that NPFS solutions are relevant in digital manufacturing environments. Interestingly enough, there are no contributions analyzing NPFS problems with lot streaming and delivery date-related objective functions. As far as we know Rossit et al. (2016) is the only one that applies lot streaming strategies to find non-permutation schedules, but with makespan as objective function. We intend, thus, to extend that line of analysis, studying the same problem but under objective functions appropriate for production systems focused on the customer, as for instance seeking the minimization of total tardiness. These features are highlighted in the last row of Table 1, indicating that the current one is the only study incorporating NPFS and lot streaming as well as Total Tardiness as objective function.

3. Mathematical models

In this section we discuss the mathematical formulation of our problem. Since it involves Industry 4.0 and client-oriented production system (Wang et al. 2017; El Hamdi et al. 2019; Perez et al. 2022) some of the classical assumptions in the analysis of scheduling problems must be replaced. For instance, production orders are no longer make-to-stock but make-to-order, and thus, will not be released in bulk but according to demand. Then, the release date becomes a relevant feature of jobs. Other assumptions about this scheduling problem are:

- Preemption is not allowed
- Each machine can process only one job (or sublot) at a time
- Each job (or sublot) can be processed by only one machine at a time
- Processing times are standard and deterministic

We follow here the notion of Graham et al. (1979), in which $F\left|r_j\right|\sum_j T_j$

corresponds to NPFS without lot streaming, while $F\left|r_j, lot\ streaming\right|\sum_j T_j$ denotes the problem with lot streaming.

3.1. NPFS without lot streaming

Sets

J: Jobs, indexed by {*j*}

M: Machines, indexed by {*i*}

Parameters

$p_{i,j}$ processing time of unit of job *j* at the machine *i*

r_j release date of job *j*

d_j due date of job *j*

U_j Lot size of items produced by job *j*

$st_{i,j}$ setup time for processing job *j* at machine *i*.

Ω a positive large number

Variables

$C_{i,j}$ Completion time of job *j* at machine *i*.

T_j tardiness of job *j*.

$x_{j',j,i}$ binary, 1 if job *j'* is processed before job *j* at machine *i*, 0 otherwise.

$$\min z = \sum_{j=1}^{n} T_j \tag{1}$$

$$C_{ij} \geq C_{(i-1)j} + p_{ij} \cdot U_j + st_{ij} + tr_{ij}, \quad \forall j, i > 1 \tag{2}$$

$$C_{ij} \geq C_{ij'} + p_{ij} \cdot U_j + st_{ij} - \left(1 - x_{j'ji}\right) \cdot \Omega, \quad \forall i, j \neq j' \tag{3}$$

$$x_{j'ji} + x_{jj'i} = 1, \; j \neq j' \tag{4}$$

$$C_{ij} \geq r_j + p_{ij} \cdot U_j + st_{ij}, \quad i = 1, \forall j \tag{5}$$

$$T_j = \max\left\{0, d_j - C_{(i=m)j}\right\}, \quad \forall j \tag{6}$$

$$T_j, C_{ij} > 0; x_{ij} \in \{0,1\} \tag{7}$$

Expressions (1)-(7) characterize the problem. (1) indicates the objective function, the minimization of total tardiness (which is computed according to equation (6)). Inequality (2) represents the precedence restriction: a job cannot be processed by machine *i* until the processing has finished in machine *i* – 1. Inequality (3) indicates that a job *j* can be processed by machine *i* after job *j'* has released *i*, if and only if *j'* precedes *j* in the sequence. Equation (4) is the logic constraint according to which if job *j'* precedes job *j* on machine *i*, the opposite cannot be the case. Inequality (5) represents a capacity constraint on the first machine, according to which no job cannot start its processing before a request has been received in the form of a due release date, and the completion time depends on all the activities involved in its processing. Equation (6) determines the tardiness of each job with respect to its due date, considering only positive values of tardiness. Finally, (7) are the feasibility conditions on the variables.

3.2. NPFS with lot streaming

We have to introduce the expressions that correspond to the incorporation of lot streaming strategies. We keep expressions (1), (4) (6) and (7) of the previous subsection, while (2), (3) and (5) have to be adapted to consider sublots. Some additional constraints are also needed.

Sets

F: sublots, indexed by {f}

Parameters

$tr_{i,j}$ transfer time of a sublot of job j from machine i to machine i + 1.

$stm_{i,j}$ setup time for processing a sublot of job j at machine i.

Variables

$C_{i,f,j}$ Completion time of sublot f of job j at machine i.

$s_{f,j}$ sublot size of sublot f of job j.

$$\sum_{f=1}^{F} s_{fj} = U_j, \quad \forall j \tag{8}$$

$$s_{fj} \leq y_{fj} \cdot \Omega, \quad \forall f, j \tag{9}$$

$$C_{i(f=1)j} \geq C_{(i-1)(f=1)j} + p_{ij} \cdot s_{(f=1)j} + st_{ij} + tr_{ij}, \quad \forall j, i > 1 \tag{10}$$

$$C_{i(f=1)j} \geq C_{i(f=F)j'} + p_{ij} \cdot s_{(f=1)j} + st_{ij} - \left(1 - x_{j'ji}\right) \cdot \Omega, \quad \forall i, j \neq j' \tag{11}$$

$$C_{i(f=1)j} \geq C_{i(f=F)j'} + p_{ij} \cdot s_{fj} + stm_{ij} \cdot y_{fj}, \quad \forall i, j, f = 2, \ldots, F-1 \tag{12}$$

$$C_{ifj} \geq C_{i(f-1)j} + p_{ij} \cdot s_{fj} + y_{fj} \cdot \left(stm_{ij} + tr_{ij}\right), \quad \forall f, j, i = 2, \ldots M-1 \tag{13}$$

$$C_{i(f=1)j} \geq r_j + p_{ij} \cdot s_{(f=1)j} + st_{ij}, \quad i = 1, \forall j \tag{14}$$

Expression (8) indicates that all the units of job *j* must be included in a sublot *f* of *j*. Since sublots are not fixed (i.e. the size of the sublots is determined by the optimization process), inequality (9) detects the non-empty sublots which require setups and displacement times. Equation (10) is a precedence inequality analogous to (2): the first sublot of a product cannot be processed by machine *i* until it has been finished at machine *i* – 1. (11) captures the same constraint as (3), namely that job *j* can be processed after *j'* has released machine *i*, if and only if *j'* precedes *j* in the sequence. This is done considering the first and last sublots of *j* and *j'*, $f = 1$ and $f = F$, respectively. Inequality (12) orders the sublots of the same job to be processed sequentially at a given machine. In turn, equation (13) indicates that a sublot cannot be processed simultaneously by two different machines. Constraint (14) replaces (5) ensuring that the first sublot of a job will not be processed until its release date has been received.

4. Experiments and results

We present here the experiment design and the results obtained by using exact methods (CPLEX). These experiments are in order to compare the models with and without lot streaming, analyzing the impact of using lot streaming strategies.

4.1. Experimental design

We aimed to detect whether including lot streaming strategies improve results in Industry 4.0 environments. In order to do that, we tested problems of different sizes (in jobs and machines) and different numbers of sublots. The number of jobs chosen was 4, 6, 8 and 10, as well as 3, 5 and 10 machines. We covered all the possible combinations yielding 12 different problems. In turn, for the problems with lot streaming we considered different numbers of sublots. To incorporate a larger number of sublots implies to extend the range of *f*, increasing the number of instances of expressions (8) – (14), with the consequence of enlarging the computation cost of analyzing the problems. For *f* we chose 2, 3, 4 and 5, meaning that we had to solve 48 problems.

For the parameters defined in subsections 3.1 and 3.2 we selected the following values:

$p_{i,j}$ uniform distribution [1;5] (it corresponds to processing each unit of U_j).

r_j uniform distribution [1;50]

U_j uniform distribution [1;22]

$st_{i,j}$ uniform distribution [10;25]

$tr_{i,j}$ uniform distribution [1;4]

$stm_{i,j}$ uniform distribution [1;10]

For d_j we used the following rule: $d_j = r_j + \sum_{i=1}^{m} p_{ij}$.

Five data sets are generated for all the combinations of machines, jobs and sublots. Each data set corresponds to a well-defined problem where each parameter takes a value drawn from one of the probabilistic distributions presented above. Then, each problem is solved deterministically by CPLEX12.10, with a time limit of 3.600 seconds. The experiments are performed on an Intel Core i5-7200U PC with 8GB of RAM.

4.2. Results

Our analysis starts by considering the results on the impact of using lot streaming to solve an NPFS problem with total tardiness as objective function. Table 2 shows the value of the objective function with plain NPFS solutions and the improvement resulting from using lot streaming strategies. The improvements are expressed as percentages.

Table 2. Improvement in the value of the objective function, with respect to the different number of sublots allowed. Results correspond to the average of all the runs.

N	m	NPFS	NPFS-lot streaming			
			$2f$	$3f$	$4f$	$5f$
4	3	521	60.7%	61.4%	61.4%	61.4%
	5	768	27.1%	35.4%	37.0%	37.2%
	10	1241	90.7%	95.2%	98.5%	100.0%
6	3	1149	39.1%	42.1%	42.2%	42.2%
	5	1504	18.9%	23.5%	24.3%	24.5%
	10	2219	73.3%	83.5%	85.4%	86.0%
8	3	2058	30.0%	30.2%	30.2%	30.2%
	5	2523	12.3%	15.9%	16.0%	16.0%
	10	3513	60.9%	69.9%	72.0%	-
10	3	3136	24.0%	-	-	-
	5	3721	-	-	-	-
	10	4796	-	-	-	-

Table 2 shows clearly that lot streaming has a considerable impact in improving the objective function. In many cases those improvements are over 50%, and for some case, like the case of 4 jobs and 10 machines, the result is 100% better when 5 sublots are allowed for each job. This means that no product was delivered at a late date, complying with the agreed on delivery dates while without lot streaming total tardiness was 1241. Also in the cases where the improvement is not that large, it is over 10%, meaning that the whole system performance can be enhanced without requesting new machines or doubling resources, just exploiting production planning strategies.

These enhancements are related to the number of sublots: the more the lot is split in sublots, the larger the resulting improvement. This can be observed by comparing at Table 2, at the same row, moving to the right. Nevertheless, this improvement is not monotonic, since it reaches a maximum. The largest variations from a number of sublots to the next one obtain at the transition from no lot streaming to allowing 2 sublots. The improvement from further increases in the number of sublots is less pronounced. On the downside, notice that incorporating lot streaming strongly increases the computational cost of finding exact solutions. This can be seen in Table 2 by the use of "-" in the cases in which no satisfactory solution is found after an hour of running the solver. We mean by "satisfactory" here a solution that yields a better result with the incorporation of more sublots. So, for instance, if with 2 sublots total tardiness is 1136, when we increase the division to up to 3 sublots, the result will be less than 1136 (since the case of up to 3 sublots includes the case of 2).

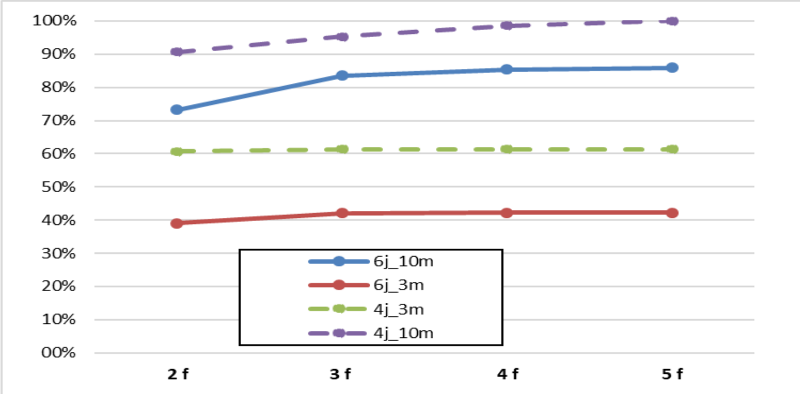

Figure 2. Lot streaming objective function improvement with respect to different numbers of sublots

On the other hand, the impact of lot streaming varies with the size of the problem. Figure 2 depicts the results for problems with 4 jobs and 3 and 10 machines, (dotted lines) and 6 jobs with 3 and 10 machines (solid lines). We can see that keeping the number of jobs fixed, lot streaming yields better results with more machines. In turn, if we fix the number of machines, a larger number of jobs worsen the objective function. Finally, all the curves have the same shape, with decreasing marginal increases as a function of the number of allowed sublots. That is, there seems to be a saturation number of sublots, after which the objective function no longer improves. We can analyze this more clearly seeing Table 3.

Table 3. Number of sublots used in final solutions. (The values are presented in average).

N	*m*	*f_allowed*	*f_used*
4	3	2	1.8
		3	2.05
		4	2.05
		5	2.05
	5	2	1.95
		3	2.65
		4	3.1
		5	3.25
	10	2	2
		3	2.9
		4	3.25
		5	3.25
6	3	2	1.93
		3	2.03
		4	2.1
		5	2.1
	5	2	1.97
		3	2.6

		4	2.73
		5	2.93
	10	2	2
		3	2.97
		4	3.53
		5	3.71
8	3	2	1.6
		3	1.65
		4	1.75
		5	1.75
	5	2	1.93
		3	2.58
		4	2.75
		5	2.75
	10	2	2
		3	2.78
		4	3.53
		5	-
10	3	2	1.5

Table 3 shows that, even if a number of sublots are allowed, the optimal value of the objective function can be reached using fewer sublots. As shown by Table 2, we can see that allowing more sublots may improve the results in certain cases, but with an increasing computational cost. It is, thus, highly relevant to determine the useful maximal number of sublots that may allow to benefit from adding lot streaming to the search of solutions to NPFS.

Table 4. Average CPU time for solving each problem to optimality.

N	*m*	*NPFS*	*NPFS-lt*			
			2 f	*3 f*	*4 f*	*5 f*
4	**3**	<sec	<sec	<sec	<sec	<sec
	5	<sec	<sec	1,0	1,0	2,0
	10	<sec	<sec	2,0	2,0	2,0
6	**3**	<sec	<sec	<sec	<sec	1,0
	5	<sec	<sec	1,0	2,0	2,0
	10	<sec	1,0	2,0	3,0	3,0
8	**3**	<sec	<sec	<sec	1,0	2,0
	5	1,0	1,0	2,0	10,0	191,0
	10	1,0	1,0	3,0	11,0	429,0
10	**3**	2,0	1,0	4,0	4,0	>3600
	5	2,0	>3600	>3600	>3600	>3600
	10	3,0	>3600	>3600	>3600	>3600

For a deeper analysis we examine the computational behavior of the problem according to the features of the problem (number of machines, jobs and allowed sublots). The results are shown in Table 4. It can be seen that for any problem size, lot streaming has a direct impact on the computational effort, increasing the time demanded to solve the problem. This effect is proportional to the maximum number of sublots allowed. The larger the number of allowed sublots, the larger the CPU time required by the solver to yield the optimal solution.

4.3. Discussion of results and future developments

Let us consider the cases in which allowing more sublots per job is associated to a reduction in the value of the objective function (Table 2), for instance, in the case of 4 jobs and 10 machines. In this case if we consider the information provided by Table 3, the average number of sublots does not change (it remains fixed at 3.25) when the maximum allowed number of sublots increases from 4 to 5 sublots. On the other hand, the objective function corresponding to these problems (Table 2) yields a lower value in the case of 5 allowed sublots than in the case of 4 sublots. This means that when the maximum number of sublots remains fixed at 4, some jobs are divided into 4 sublots (the average is over 3), but when 5 sublots are allowed the average is the same. This can be explained by the fact that when 5 sublots are allowed, some jobs that were split into 4 sublots in the case of a maximum of 4 sublots can now be divided into 5 sublots while some other jobs are split into 3 sublots. The composition of sublots must change, because the value of the objective function changes. This prevents us from considering that the solution structure will remain the same for both maximum numbers of sublots.

5. Conclusions

In this article we analyze the introduction of lot streaming to find optimal schedules in Industry 4.0 environments focused on the requests of customers. We seek non-permutation solutions appropriate to flow shop problems. We found that incorporating lot streaming strategies improves results, reducing the total tardiness of delivery. We detected that subdividing the number of items in more sublots has cumulative beneficial effects up to a point. Afterwards, adding more sublots does not improve further the results. On the other hand, the computational cost of lot streaming is considerably larger than those of finding solutions without lot streaming.

The main conclusion it that while some jobs can be divided into several sublots, others are more resistant to be split. If the jobs can be classified by their features (number of units, accumulated processing times, due dates, etc.), the optimizing process can be fine-tuned to allow more sublots only for the types of jobs that require them while keeping as low as possible the number of sublots of the other types. This will reduce the number of variables, and consequently the computational burden of the optimization process. But classifying jobs requires further research since the analyses presented here do not provide enough information on the best way of doing it.

This opens up the possibility of focusing the computational effort (in terms of variables and number of sublots) on those jobs. But detecting them may require a further and deeper analysis. A promising future line of research involves the possibility of running first a parametric analysis of the different types of instances to identify which jobs require this special attention. It would be interesting to design modelling tools able to take advantage of this hypothesis, orienting the computational resources (in terms of variables and restrictions) to those jobs that may need them rather than to the entire set of jobs.

References

Benavides, A. J., & Ritt, M. (2016). Two simple and effective heuristics for minimizing the makespan in non-permutation flow shops. Computers & Operations Research, 66, 160-169. https://doi.org/10.1016/j.cor.2015.08.001

Benavides, A. J., & Ritt, M. (2018). Fast heuristics for minimizing the makespan in non-permutation flow shops. Computers & Operations Research, 100, 230-243. https://doi.org/10.1016/j.cor.2018.07.017

Bicakci, P. S., & Kara, İ. (2019). A New Formulation for the Single Machine Order Acceptance and Scheduling Problem with Sequence-Dependent Setup Times. International Journal of Supply and Operations Management, 6(2), 159-167. https://dx.doi.org/10.22034/2019.2.5

Cetinkaya, F. C., & Duman, M. (2021). Scheduling with lot streaming in a two-machine re-entrant flow shop. Operational Research in Engineering Sciences: Theory and Applications, 4(3), 142-175. https://doi.org/10.31181/oresta111221142c

Conway, R. W., Maxwell, W. L., & Miller, L. W. (1967). Theory of scheduling. Courier Corporation

D'Amico, F., Rossit, D. A. & Frutos, M. (2021). Lot streaming Permutation Flow shop with energy awareness. International Journal of Industrial Engineering and Management, 12(1), 25-36. http://dx.doi.org/10.24867/IJIEM-2021-1-274

Dolgui, A., Ivanov, D., Sethi, S. P., & Sokolov, B. (2019). Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. International Journal of Production Research, 57(2), 411-432. https://doi.org/10.1080/00207543.2018.1442948

El Hamdi, S., Oudani, M., & Abouabdellah, A. (2019). Towards Identification of the Hierarchical Link between Industry 4.0, Smart Manufacturing and Smart Factory: Concept Cross-Comparison and Synthesis. International Journal of Supply and Operations Management, 6(3), 231-244. https://dx.doi.org/10.22034/2019.3.4

Ferraro, A., Rossit, D., Toncovich, A., & Frutos, M. (2019). Lot streaming flow shop with a heterogeneous machine. Engineering Management Journal, 31(2), 113-126. https://doi.org/10.1080/10429247.2018.1522221

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research, 1(2), 117-129. https://doi.org/10.1287/moor.1.2.117

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In Annals of discrete mathematics 5, 287-326. https://doi.org/10.1016/S0167-5060(08)70356-X

Grassi, A., Guizzi, G., Santillo, L. C., & Vespoli, S. (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. Manufacturing Letters, 24, 43-46. https://doi.org/10.1016/j.mfglet.2020.03.007

Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. International Journal of Production Research, 54(2), 386-402. https://doi.org/10.1080/00207543.2014.999958

Jain, A. S., & Meeran, S. (2002). A multi-level hybrid framework applied to the general flow-shop scheduling problem. Computers & Operations Research,29(13), 1873-1901. https://doi.org/10.1016/S0305-0548(01)00064-8

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. Naval research logistics quarterly, 1(1), 61-68. https://doi.org/10.1002/nav.3800010110

Koulamas, C. (1998). A new constructive heuristic for the flowshop scheduling problem. European Journal of Operational Research, 105(1), 66-71. https://doi.org/10.1016/S0377-2217(97)00027-1

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing letters, 3, 18-23. https://doi.org/10.1016/j.mfglet.2014.12.001

Liao, C. J., Liao, L. M., & Tseng, C. T. (2006). A performance evaluation of permutation vs. non-permutation schedules in a flowshop. International Journal of Production Research, 44(20), 4297-4309. https://doi.org/10.1080/00207540600595892

Liao, L. M., & Huang, C. J. (2010). Tabu search for non-permutation flowshop scheduling problem with minimizing total tardiness. Applied Mathematics and Computation, 217(2), 557-567. https://doi.org/10.1016/j.amc.2010.05.089

Nagarajan, V., & Sviridenko, M. (2009). Tight bounds for permutation flow shop scheduling. Mathematics of Operations Research, 34(2), 417-427. https://doi.org/10.1287/moor.1080.0368

Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Information sciences, 181(12), 2455-2468. https://doi.org/10.1016/j.ins.2009.12.025

Perez, A. T. E., Rossit, D. A., Tohme, F., & Vásquez, Ó. C. (2022). Mass customized/personalized manufacturing in Industry 4.0 and blockchain: Research challenges, main problems, and the design of an information architecture. Information Fusion, 79, 44-57. https://doi.org/10.1016/j.inffus.2021.09.021

Pinedo, M. (2012). Scheduling (Vol. 29). New York: Springer.

Potts, C. N., Shmoys, D. B., & Williamson, D. P. (1991). Permutation vs. non-permutation flow shop schedules. Operations Research Letters, 10(5), 281-284. https://doi.org/10.1016/0167-6377(91)90014-G

Rebaine, D. (2005). Flow shop vs. permutation shop with time delays.Computers & Industrial Engineering, 48(2), 357-362. https://doi.org/10.1016/j.cie.2005.01.019

Rossi, A., & Lanzetta, M. (2014). Native metaheuristics for non-permutation flowshop scheduling. Journal of Intelligent Manufacturing, 25(6), 1221-1233. https://doi.org/10.1007/s10845-012-0724-8

Rossit, D., Tohmé, F., Frutos, M., Bard, J., & Broz, D. (2016). A non-permutation flowshop scheduling problem with lot streaming: A Mathematical model. International Journal of Industrial Engineering Computations, 7(3), 507-516. http://dx.doi.org/10.5267/j.ijiec.2015.11.004

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

Rossit, D. A., Vásquez, Ó. C., Tohmé, F., Frutos, M., & Safe, M. D. (2018b). The dominance flow shop scheduling problem. Electronic Notes in Discrete Mathematics, 69, 21-28. https://doi.org/10.1016/j.endm.2018.07.004

Rossit, D. A., Vásquez, Ó. C., Tohmé, F., Frutos, M., & Safe, M. D. (2021a). A combinatorial analysis of the permutation and non-permutation flow shop scheduling problems. European Journal of Operational Research, 289(3), 841-854. https://doi.org/10.1016/j.ejor.2019.07.055

Rossit, D. A., Toncovich, A., Rossit, D. G. & Nesmachnow, S. (2021b). Solving a flow shop scheduling problem with missing operations in an Industry 4.0 production environment. Journal of Project Management, 6(1), 33-44. http://dx.doi.org/10.5267/j.jpm.2020.10.001

Rudek, R. (2011). Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. Computers & Industrial Engineering, 61(1), 20-31. https://doi.org/10.1016/j.cie.2011.02.005

Sarin, S. C., & Jaiprakash, P. (2007). Flow shop lot streaming. Springer Science & Business Media.

Stanković, A., Petrović, G., Ćojbašić, Ž., & Marković, D. (2020). An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem. Operational Research in Engineering Sciences: Theory and Applications, 3(3), 13-28. https://doi.org/10.31181/oresta20303013s

Strusevich, V. A., & Zwaneveld, C. M. (1994). On non-permutation solutions to some two machine flow shop scheduling problems. Zeitschrift für Operations Research, 39(3), 305-319. https://doi.org/10.1007/BF01435460

Tandon, M., Cummings, P. T., & LeVan, M. D. (1991). Flowshop sequencing with non-permutation schedules. Computers & chemical engineering, 15(8), 601-607. https://doi.org/10.1016/0098-1354(91)80014-M

Trietsch, D., & Baker, K. R. (1993). Basic techniques for lot streaming. Operations Research, 41(6), 1065-1076. https://doi.org/10.1287/opre.41.6.1065

Wang, Y., Ma, H. S., Yang, J. H., & Wang, K. S. (2017). Industry 4.0: a way from mass customization to mass personalization production. Advances in Manufacturing, 5(4), 311-320. https://doi.org/10.1007/s40436-017-0204-7

Xu, L. D., Xu, E. L., & Li, L. (2018). Industry 4.0: state of the art and future trends. International Journal of Production Research, 56(8), 2941-2962. https://doi.org/10.1080/00207543.2018.1444806

Ying, K. C., Gupta, J. N., Lin, S. W., & Lee, Z. J. (2010). Permutation and non-permutation schedules for the flowline manufacturing cell with sequence dependent family setups. International Journal of Production Research, 48(8), 2169-2184. https://doi.org/10.1080/00207540802534707

Ziaee, M. (2013). General flowshop scheduling problem with the sequence dependent setup times: A heuristic approach. Information Sciences, 251, 126-135. https://doi.org/10.1016/j.ins.2013.06.025