

# Solving two production scheduling problems with sequence-dependent set-up times

João António Noivo<sup>a</sup> and Helena Ramalhinho-Lourenço<sup>b</sup>

*a* Departamento de Electrónica Industrial, Universidade do Minho, Campus de Azurém, 4800 Guimarães, Portugal. Joao.Noivo@dei.uminho.pt

*b* Department of Economics and Business, Universitat Pompeu Fabra, R. Trias Fargas, 25-27, 08005 Barcelona, Spain. ramalhin@upf.es

**Journal of Economic Literature Classification:** C61, M11, L60

---

## Abstract

In today's competitive markets, the importance of good scheduling strategies in manufacturing companies lead to the need of developing efficient methods to solve complex scheduling problems.

In this paper, we studied two production scheduling problems with sequence-dependent setups times. The setup times are one of the most common complications in scheduling problems, and are usually associated with cleaning operations and changing tools and shapes in machines.

The first problem considered is a single-machine scheduling with release dates, sequence-dependent setup times and delivery times. The performance measure is the maximum lateness.

The second problem is a job-shop scheduling problem with sequence-dependent setup times where the objective is to minimize the makespan.

We present several priority dispatching rules for both problems, followed by a study of their performance. Finally, conclusions and directions of future research are presented.

## 1. Introduction

Recent trends in scheduling attempt to fill the gap between scheduling theory and scheduling practice, with the aim to give answer to respond to market demand for more efficient method to solve complex scheduling problems. Although classical scheduling theory are one of the most studied field in Operations Research, some practical environments are often ignored in the classical models, since they improve the complexity of mathematical models. For a discussion in the gap between scheduling theory and scheduling practice see MacCarthy and Liu (1993).

The setup times appear frequently in real scheduling problems and are one of the most frequent additional complications in scheduling. Moreover, these type of constraints are particularly relevant in production scheduling.

The setup time is defined as the time intervals between the end of job processing and beginning of next job. In this time interval no jobs can be processed in machine. The cleaning operations and changing tools and shapes are some examples of these setup times, and are frequent in manufacturing companies as commercial printing, plastics manufacturing, metal and chemical processing, paper industry, etc. The most complicated case is sequence-dependent setup times, where the setup time depends on the job previously scheduled. A typical example is the manufacturing of different colors of paint, Conway et al. (1967). In this case a cleaning operation time is needed, and is related with sequence of the colors processed. Another example is the extrusion machine for plastics films. The time spent in cleaning operations depends of film type and color. The trend in manufacturing of the production of small batches or unit products to satisfy demand and avoid inventory has made more relevant the scheduling problems with sequence-dependent setup times between all jobs, and not only between batches.

The aim of this paper is to study the performance of dispatching priority rules for the single-machine and job-shop scheduling problems with sequence dependent-dependent setup times and to indicate how to develop a good heuristic strategy to solve these problems in a practical and dynamic environment.

In chapter 2, we present related scheduling research with setup times. In chapter 3 and 4, we present the single-machine and the job-shop scheduling problem with sequence-dependent setup times, respectively. The priority dispatching rules are present in chapter 5, for the single-machine scheduling and in chapter 6 for the job-shop scheduling. Chapter 7 and 8 present the

computational experiment and the results obtained, and finally we present the conclusions and the directions of future research.

## **2. Related research**

Different types of setup times have been considered in the literature, and can be classified in four different types:

1. Sequence-independent setup times. In this situation, they can be added to the processing times.
2. Sequence-independent batch setup times. There are no setup times into sets of identical jobs (batches). They only appear when changing the batch in production, and are independent of the jobs previously scheduled.
3. Sequence-dependent batch setup times. In this case the setup times also appear only when processing a switch from a job in a batch to a job in another batch, but the time depends on the batch of the previously scheduled job.
4. Sequence-dependent setup times. This is the most general and complex case, where there exist a setup time between any pair of jobs that depends on the sequence.

Hoogeveen et al. (1997) and Lawler et al. (1993) present a review of the main contributions to the area deterministic scheduling problems, with emphasis on the classical models. Most of the references are on theoretical work, and with respect to setup times, the only references are on sequence-independent batch setup times for the single-machine scheduling problem. There is no reference to sequence-dependent setup times for the single-machine and job-shop scheduling problems.

For the single-machine scheduling problems with sequence-independent batch setup times several works have been published where different performance measures are considered. See Bruno and Downey (1978), Monma and Potts (1989), Zdrzalka (1992), Williams and Wirth (1996), Mason and Anderson (1991) and Gupta (1988).

Kim and Bobrowski (1994) present a computer simulation model for a limited machine job-shop scheduling problem with sequence-dependent setup times. They study the influence of setup times and due dates information in priority rules performance for job-shop problem with setup times.

Ovacik and Uzsoy (1994) present a family of rolling horizon heuristics to minimize the maximum lateness on a single machine in the presence of sequence-dependent setup times. They also present a survey on the work done on this scheduling problem.

Laguna (1997) presents an heuristic procedure to a realistic production and inventory control problem with sequence-dependent setup times. The heuristic is based on a simple short-term tabu search coordinated with a linear programming and traveling salesperson solvers to guide the search.

Ríos-Mercado and Bard (1997) present a branch-and-bound enumeration method scheme for the makespan minimization of the flow-shop scheduling problem with sequence-dependent setup times.

### 3. Single-machine problem with sequence-dependent setup times

The single-machine scheduling problem with sequence-dependent setup times can be described as follows: a set of jobs  $J_1, \dots, J_n$  have to be scheduled on one machine; each job  $J_j$ ,  $j \in N = \{1, \dots, n\}$  has a release date  $r_j$ , a processing time  $p_j$ , and a delivery time  $q_j$ . Each job cannot be processed before its release time. Whereas at most one job can be processed at a time, all jobs can be simultaneously delivered; if  $C_j$  denotes the time at which job  $J_j$  completes processing, then it is delivered at time  $L'_j = C_j + q_j$ . Finally, between the completion time of  $J_j$ ,  $C_j$ , and the starting time of the next job in the sequence  $J_k$ ,  $S_k$ , there must exist a time interval, the setup time,  $s_{jk} \geq 0$ , i.e.  $C_j + s_{jk} \leq S_k$ . During this setup time, no other job can be processed in the machine. The objective is to minimize the maximum delivery date, i.e.  $L'_{\max} = \max_{j \in N} L'_j$ .

The single-machine scheduling problem with sequence-dependent setup times can be denoted by  $I/r_j, s_{ij}/L'_{\max}$ , and can be considered as a generalization of the usual single-machine scheduling problem, i.e. the setup times are all equal to zero. The problem is NP-Complete, even if no setup times are considered, Lenstra, Rinnooy Kan and Brucker (1977).

The Horn's algorithm, Horn (1974) solves to optimality the standard single-machine problem if preemption is allowed. If no preemption is allowed the Carlier branch-and-bound

algorithm, Carlier (1982), solves the standard single machine problem. Schrage (1971) presented an efficient heuristic for the same problem, that will be presented in section 5.1.

#### 4. Job-shop problem with sequence-dependent setup times

The classical job-shop scheduling problem,  $J//C_{max}$ , is defined as follows: we are given a set of  $m$  machines  $\{M_1, \dots, M_m\}$  and a set of  $n$  jobs  $\{J_1, \dots, J_n\}$ . Each job  $J_j$ ,  $j=1, \dots, n$ , consists of a sequence of  $m_j$  operations  $O_{1j}, \dots, O_{m_j j}$ , where  $O_{ij}$  is an operation of job  $J_j$  to be processed on machine  $\mu_{ij}$  for a given uninterrupted processing time  $p_{ij}$ , where  $\mu_{ij} \neq \mu_{i+1j}$ , for  $i=1, \dots, m_j$ ,  $j=1, \dots, n$ .

The operations of each job must be processed in the given sequence. Each machine  $M_i$ ,  $i=1, \dots, m$ , can process at most one operation at a time, and at most one operation of each job  $J_j$ ,  $j=1, \dots, n$ , can be processed at a time. Let  $C_{ij}$  be the completion time of operation  $O_{ij}$ . The objective is to get a schedule that minimizes the maximum completion time  $C_{max} = \max_{i,j} C_{ij}$ . A schedule is an allocation of a single time interval for each operation.

The job-shop scheduling problem is *NP*-hard in the strong sense, Garey and Johnson (1976). Furthermore, most of the special cases of this problem are *NP*-hard or strongly *NP*-hard.

In this work we will consider an extension of the classical job-shop scheduling, in the presence of sequence-dependent setup times. We will denote this problem by  $J/s_{ij}/C_{max}$ . For each pair of operations,  $O_{ij}$  and  $O_{kl}$  processed in the same machine, define a setup time  $s_{ij,kl}$  such that if  $O_{ij}$  is processed just before  $O_{kl}$  then there must exist a time interval of value  $s_{ij,kl}$  where no other operation can be processed. Note that to define the sequence-dependent setup times, we need  $m \times n \times n$  matrices. The performance measure is again the makespan.

#### 5. Heuristics Methods for $1/r_j, s_{ij}/L'_{max}$

In this chapter, we will present the priority dispatching heuristics for the single-machine scheduling problem with sequence-dependent setup times. But, first, we will describe the Schrage heuristic for the problem without setup times.

## 5.1. Schrage Heuristic

Schrage (1971) presented a polynomial heuristic to solve the single-machine scheduling problem with release dates and delivery times. The heuristic does not guarantee the optimal solution, but obtains very good results in reasonable times. The strategy adopted by the Schrage heuristic is to schedule the available job not yet scheduled with the longest delivery time at any time.

Consider the job sequence  $S = (S(1), \dots, S(n))$  created by Schrage heuristic. Then, there exist a set of jobs that are scheduled in sequence an such that the associated performance measure ( $L'_{max}$ ) is as follows:

$$L'_{max} = r_{S(i)} + \sum_{h=i}^j p_{S(h)} + q_{S(j)} \quad \text{for } 1 \leq i \leq j \leq n.$$

The sequence  $S(i), \dots, S(j)$  is designated by critical sequence, and the last job in sequence is the critical job. The output schedule of the above heuristic below to the class of non-delay schedules, which is characterized by the no existence of available jobs to be scheduled if a machine is idle (Conway, Maxwell et al. 1967). Therefore we can guarantee that:  $r_{S(i)} \leq r_{S(h)}$  for  $h = i+1, \dots, j$ . It is well known that if  $q_{S(j)} \leq q_{S(h)}$  for  $h=i, \dots, j-1$ , then the schedule is the optimal for the problem  $1/r_j/L'_{max}$ .

In a certain way, the strategy adopted by the Schrage heuristic is similar to Earliest Due Date which obtains the optimal solution for the scheduling problem  $1//L_{max}$ , where due dates are considered instead of delivery times. The problems are symmetric and equivalent, Lageweg et al. (1978).

## 5.2. Adapted Schrage Heuristic

The problem  $1/r_j, s_{ij}/L'_{max}$  is more complex due the presence of the sequence-dependent setup times. If we the apply the Schrage Heuristic, updating the release dates to guarantee the feasibility with respect to the setup time constraints, the new value of  $L'_{max}$  is as follows:

$$L'_{max} = r_{S(i)} + \sum_{h=i}^j p_{S(h)} + \sum_{h=i+1}^j s_{S(h-1)S(h)} + q_{S(j)} \quad \text{for } 1 \leq i \leq j \leq n$$

The extra expression in the formula is associated with the setup times, however these values are not taking in account when selecting the next job to be scheduled. Therefore the schedule obtained by the heuristic Schrage can lead to high values of the setup times and

consequently to high values of  $L'_{max}$ . On the other hand, if we only take in account the setup times, can lead to solutions where the delivery time of the critical job is too high.

The objective is to develop an heuristic that will consider both elements, the delivery times and the setup time, when selecting the next job to be scheduled. Therefore, the major change done in the adapted Schrage heuristic is related with taking in consideration the setup times in the selection of the next job to be schedule from the subset of available jobs not yet scheduled. Please note that the release times must be updated to verify the setup times constraints.

The main idea of the new heuristic is, first, select a subset of the available jobs that have small setup time with respect of the last job scheduled, and afterwards, among these ones, select the job with longest delivery time. So, a job is selected if the difference between its setup time and the minimal setup time of all available and not scheduled jobs is less or equal than a percentage of the difference between the larger and smaller setup time of all available jobs. This percentage is measured by a parameter  $P$ .

This procedure looks for a reduction of the new expression in the performance measure.

A value of  $P$  allows us to control the equilibrium between selecting the jobs only considering the setup times, or selecting the jobs with respect of the delivery times or taking in account both times. The value of the control parameter  $P$  can vary between 0 and 100 %. If  $P$  is equal to 100%, then we are applying the original Schrage Heuristic, and not considering the setup times at all. On the other hand if  $P$  is equal to 0%, we are selecting for scheduling the job with smallest setup time with respect of the last job scheduled and ignoring the delivery times. Any intermediate value will take in account both times, such that, if the value is close to zero means than the setup times have more importance and in a opposite way, if the value is close to 100% means that the delivery time play a more important role for selecting the next job to be scheduled.

## **6. Priority Rules for the Problem $J/s_{ij}/C_{max}$**

In this chapter, we present the classical dispatching priority rules and the two new rules for the job-shop scheduling problem with sequence-dependent setup times.

## 6.1. Classical priority rules

The most well-known and simple heuristic to solve the job-shop scheduling problem are the priority dispatching rules, also known by list scheduling. These heuristics are greedy heuristic, where at each iteration an operation is selected between the available ones and scheduled as soon as possible in the respective machine. The choice of the next operation to be scheduled is done following a priority function or rule.

Several priority rules have been proposed for the job-shop scheduling, and also, several works have been done to compare the performance of these rules. See for example: Jeremiah, Lalchandani et al. (1964) , Lawrence (1984), Morton and Pentico (1993) and Barman (1997) .

The most common rules used are the following ones:

**SPT** (*Shortest Processing Time*) : select the available job with shortest processing time.

**FCFS** (*First Come First Served*) : select the available job with larger time waiting in the system.

**MWKR** (*Most Work Remaining*) : select the available job with most work remaining.

**LWKR** (*Least Work Remaining*) : select the available job with least work remaining.

**MOPNR** (*Most Operations Remaining*) : select the available job with larger number of operations remaining to be scheduled.

**RANDOM** : randomly select the next job.

## 6.2. New priority rules

None of the above rules takes in account the sequence-dependent setup times. Therefore, we propose two new priority rules. Note that, the presence of sequence-dependent setup times have a high impact in the makespan of the final schedule, so these values must be considered in some way when deciding the next operation to be scheduled.

As in the previous chapter, the new rules are mix rules which consider the classical priority rules and the setup times when selecting the next operation to be scheduled, and can be defined as follows:

**SPST** (*Shortest (Processing+ Setup Time)*) : Select the operation with smaller value of the sum of the setup time with the processing time. The setup time is related with the



previous operation scheduled in the corresponding machine.

**MWKRST** (*Most (Work Remaining - Setup Time)*): Select the operation with the greatest difference between the work remaining and the setup time related with the last operation scheduled in the correspondent machine.

Designate these rules by SPT2 e MWKR2 respectively. The motivation to use these rules is related with the observation that SPT and MWKR usually perform well for the classical job-shop scheduling.

Finally, we define a rule that only considers the sequence-dependent setup times:

**SST** (*Shortest Setup Time*): Select the operation with shortest setup time with respect of the last operation scheduled in the corresponding machine.

In the next chapter, we will present the computational experiment realized to evaluate the performance of the heuristics proposed for the single-machine and job-shop scheduling problems with sequence-dependent setup times.

## **7. Computation experiment for the problem $1/r_j, setups/L_{max}$**

### **7.1. Computational Experiment**

The test problems proposed Grabowski, Nowicki et al. (1986) for the problem  $1/r_j/L_{max}$  were extended to the problem  $1/r_j, s_{ij}/L'_{max}$ . For each instance of the problem  $1/r_j/L_{max}$  we associated a  $n \cdot n$  matrix related with the sequence-dependent setup times. These problems have the propriety that the values of the release times and delivery times range between a large set and consequently allows us to study the influence of these values in the performance of the heuristics.

The set of test problems consists in 400 instance, divided in 5 groups of 80 instances characterized by the number of jobs ( $n = 20, 40, 80, 150$  e  $200$ ).

The data was generated randomly following the uniform distribution with an interval between the unit and a maximal value  $r_{max}$ ,  $p_{max}$  and  $q_{max}$  for the delivery times, the processing times and the delivery times respectively.

Each group of 80 instance is again divided in 16 subgroups of 5 instances each for different values of  $r_{max}$ ,  $p_{max}$  and  $q_{max}$ . The 16 groups were obtained by 4 combinations of two parameters  $R, Q \in \{0.5, 2, 0.5 \times n, 2 \times n\}$  that are used to calculate the previous values as follows:  $r_{max} = 50 \times R$ ,  $p_{max} = 50$ ,  $q_{max} = 50 \times Q$ .

For the setup times, we assume that the maximum setup time is at most 20% of the maximum processing time. The motivation to assume this is frequent relation between setup times and processing times, Kim and Bobrowski (1988). The setup times also are obtained randomly following a uniform distribution with interval between the unit and  $0.2 \times p_{max}$ .

All heuristics where implement in C and the tests where done in a PC Pentium a 120Mhz.

## 7.2. Analyses of the results

One of main objectives of the computational experiment is to perform a comparative study on the results for different values of P. We considered 6 different heuristic where the value of P is: heu0 (P = 0%), heu20 (P = 20%), heu40 (P = 40%), heu60 (P = 60%), heu80 (P = 80%) e heu100 (P = 100%). We also pretend to evaluate the performance of the heuristic with respect of the dimension of the problem ( $n$ = number of jobs) and the different values of the data, as the release times, the processing times and the delivery times.

The results are presented in a aggregate format, to give a better understand of the overall performance of the heuristics.

The Table 1 presents the number of times the best solution was obtained for each heuristic and the average running time (in milliseconds)

num	tempo	heu0	heu20	heu40	heu60	heu80	heu100
400	78	216	134	144	146	162	165

**Table 1** -  $1/r_j, s_{ij}/L'_{max}$ : Global results.

From the results obtained, the first conclusion is that the 5 heuristics *heur20*, *heur40*, *heur50*, *heur80* and *heur100* had a similar behavior, and were outperformed by the heuristic *heur0*.

For a detailed analysis of the results, we present next the results aggregated in different ways to emphasize the influence of the data in the heuristics performance.

In Table 2 we present the results aggregated in function of the dimension of the problem ( $n = 20, 40, 80, 150$  e  $200$ ).

num	n	time	heur0	heur20	heur40	heur60	heur80	heur100
80	20	6	36	32	31	32	39	37
80	40	22	39	30	25	33	32	31
80	80	37	49	23	29	31	29	32
80	150	123	45	23	31	25	31	36
80	200	202	47	26	28	25	31	29

**Table 2** -  $1/r_j s_{ij}/L'_{max}$ : Results aggregated by dimension ( $n$ ).

The analysis of the results permits to detect to opposite trends in the performance of the heuristics. As the dimension of the problem increases the efficiency of *heur0* increases, i.e. the number of times the heuristic obtains the best results increases. In the opposite way, the efficiency of the remaining heuristics decreases.

The explanation for this behavior is related with the fact that, as the number of jobs increases, the importance of the sequence-dependent setup times also increases since it will have a large weight in the value of the performance measure. Therefore, we can expect that the *heur0* will perform better for large instances.

As expected the running times increase with the dimension of the problem.

In Table 3 we present the results but separated by the values of  $r_{max}$ ,  $p_{max}$  and  $q_{max}$ .

num	rmax	pmax	qmax	time	heu0	heu20	heu40	heu60	heu80	heu100
25	25	50	25	104	23	3	0	0	0	0
25	100	50	25	112	22	3	0	0	0	0
25	500	50	25	52	20	3	1	2	1	0
25	2.000	50	25	37	22	22	22	22	22	25
25	25	50	100	118	20	3	2	0	0	0
25	100	50	100	122	22	3	0	1	1	0
25	500	50	100	44	10	11	3	3	3	0
25	2.000	50	100	10	25	25	25	25	25	25
25	25	50	500	116	0	4	10	5	5	1
25	100	50	500	125	0	0	13	5	6	2
25	500	50	500	55	2	5	1	6	6	9
25	2.000	50	500	33	22	22	22	22	22	25
25	25	50	2.000	106	2	3	11	12	20	24
25	100	50	2.000	111	0	0	5	15	17	20
25	500	50	2.000	44	2	3	5	4	10	9
25	2.000	50	2.000	27	24	24	24	24	24	25

**Table 3** –  $1/r_j, s_{ij}/L'_{max}$ : Results aggregated by rmax, pmax and qmax.

The 400 instances are grouped in 16 groups of 25 instances each which were generated using the same input data for the release dates ( $r_{max}$ ), the processing times ( $p_{max}$ ) and the delivery times ( $q_{max}$ ).

For the large value of the maximum release date ( $r_{max} = 100 \times n$ ), almost all heuristics obtained good results, since these are easy problems because the jobs are rarely release at the same time. Therefore, at many time, there is at most one job available to be scheduled.

For the remaining instances, the performance of the heuristics are related with the values of the delivery times. The heuristics with small value for the parameter P (heu0, heu20 and heu40) perform better when the maximum delivery time ( $q_{max} = 25$  and  $q_{max} = 100$ ) is similar to the maximum processing time ( $p_{max} = 50$ ), and so similar to the setup times. The explanation for this behavior in this set of instances is since the delivery times are similar, the setup times play an important role in the construction of the good solutions.

On the other hand, the heuristics for large values of P (heu60, heu80 and heu100) have a better behavior when the delivery times ( $q_{max} = 25 \times n$  and  $q_{max} = 100 \times n$ ) are of greater dimension then the setup times, and the processing times ( $p_{max} = 50$ ). These heuristics give a greater importance to the delivery times than to the setup times, when selecting the next job to be scheduled, and since the first ones have higher values the performance of these heuristics can be easily explained.

The running times decrease as the value of the maximum release dates increase ( $r_{max}$ ), since, as mentioned before, these instance result to be easy to solve.

## 8. Computational experiment for the problem $J/s_{ij}/C_{max}$

### 8.1. Computational Experiment

The test problems for the problem  $J/s_{ij}/C_{max}$  can be divide in two groups:

- 82 classical problems (available in the OR library : <http://www.ms.ic.ac.uk/info.html>)
- 80 problems proposed by Taillard (Taillard 1993).

Again, we had to adapt the above problems to take in account the sequence-dependent setup times. We need to generate  $m$  square  $n \times n$  matrix, one for each machine.

The 82 classical problems are the following ones, and have dimensions between  $6 \times 6$  and  $20 \times 20$ :

- Problems 1 - 5 (Adams, Balas et al. 1988)
- Problems 6 - 8 (Fisher and Thompson 1963)
- Problems 9 - 48 (Lawrence 1984)
- Problems 49 - 58 (Applegate and Cook 1991)
- Problems 59 - 78 (Storer, Wu et al. 1992)
- Problems 79 - 82 (Yamada and Nakano 1992)

The 80 problems proposed by Taillard have larger dimension which goes from  $(10 \times 10)$  to  $(100 \times 20)$ .

The generator of random number used to generate the last set of problems was the congruential linear generator,  $X_{i+1} = (16807 \times X_i) \bmod (2^{31} - 1)$ , Bratley, Fox et al. (1983).

The data that needs to be generated for the problem  $J//C_{max}$  are the processing times of each job in each machine, and the order of processing by the machines. The processing times are generated following an uniform distribution with values between 1 and 99. The uniform distribution is also used to find the order of the machines for each job. For more details, see Taillard (1993).

## 8.2. Analyses of the results

The objective of the computational experiment for the problem  $J/s_{ij}/C_{max}$  is to study the behavior and performance of 5 priority rules described before, and to analyze the influence of the dimension of the problem ( $n \times m$ ) in the results.

Table 4 presents the global results for the Taillard instances, and indicates the number of times each heuristic had obtained the best result. Table 5 presents the same information for the classical instances.

num	SPT	MWKR	SST	SPT2	MWKR2
80	13	8	41	17	8

**Table 4** –  $J/s_{ij}/C_{max}$ : Global results for the Taillard' instances.

num	SPT	MWKR	SST	SPT2	MWKR2
82	7	14	45	12	16

**Table 5** –  $J/s_{ij}/C_{max}$ : Global results for the classical instances.

The priority rule based only in the setup times SST (*Shortest Setup Time*) outperformed the remaining ones, for all instances.

The results aggregated by dimension of the problem are presented in tables 6 and 7.

num	n	m	SPT	MWKR	SST	SPT2	MWKR2
10	15	15	2	2	4	2	2
10	20	15	2	2	4	2	2
10	20	20	6	0	2	2	0
10	30	15	0	2	6	2	1
10	30	20	3	1	4	2	1
10	50	15	0	0	6	4	0
10	50	20	0	1	5	3	2
10	100	20	0	0	10	0	0

**Table 6** -  $J/s_{ij}/C_{max}$ : Aggregate results by dimension ( $n \times m$ ) (Taillard' problems).

num	n	m	SPT	MWKR	SST	SPT2	MWKR2
1	6	6	0	0	1	0	0
5	10	5	0	0	2	3	0
5	15	5	0	2	3	0	1
6	20	5	0	1	4	1	0
18	10	10	3	3	6	5	4
5	15	10	0	0	4	1	0
10	20	10	0	1	9	0	1
5	30	10	0	0	5	0	0
5	15	15	0	4	1	0	4
8	20	15	0	2	4	1	3
4	20	20	3	1	0	0	1
10	50	10	1	0	6	1	2

**Table 7** -  $J/s_{ij}/C_{max}$ : Aggregate results by dimension ( $n \times m$ ) (classical problems).

When the number of jobs is bigger than the number of machines, we can observe that the rule SST (shortest setup time) performs better than the remaining ones. The reason for this behavior is the fact that in presence of a large number of jobs, the sequence-dependent setup times have a large impact in the output solution.

## 9. Analyze of results related with the setup times

Since the new elements introduced in this work are the sequence-dependent setup times, we present in this section a more detailed analyses of the influence of these ones in the behavior of the new heuristics.

### 9.1. Adapted Schrage heuristic for the problem $1/r_j, s_{ij}/L'_{max}$

To study the impact of the sequence-dependent setup times in the results, we decide to generate new setup times larger than the previous ones, where the maximum value of a setup time are  $s_{1max} = 0.2 \times p_{max}$  and  $s_{2max} = 0.3 \times p_{max}$ . Therefore, now we have two set of 400 instances each one with different setup times interval.

In tables 8 and 9 we present the global results to each set ( $s_{1max} = 0.2 \times p_{max}$  and  $s_{2max} = 0.3 \times p_{max}$ ) and for each of the 6 adapted Schrage heuristics. The results are measured in terms of the number of times the heuristic had obtained the best results overall.

num	tempo	heu0	heu20	heu40	heu60	heu80	heu100
400	78	216	134	144	146	162	165

**Table 8** -  $1/r_j, s_{ij}/L'_{max}$ : Global results for instances ( $s_{1max} = 0.2 \times p_{max}$ ).

num	tempo	heu0	heu20	heu40	heu60	heu80	heu100
400	81	219	131	141	142	150	161

**Table 9** -  $1/r_j, s_{ij}/L'_{max}$ : Global results for instances ( $s_{2max} = 0.3 \times p_{max}$ ).

As it can be observed, for the instance with larger setup times, the heur0 outperformed the remaining ones, and also it obtains more times the best results than for the instances with smaller setup times. This behavior was expected, since as the setup times increase, these ones play a more important role in finding a good solution.

## 9.2. Priority rules for the problem $J/s_{ij}/C_{max}$

As done for the single machine scheduling problems, we generate a second set of instance with larger setup times to analyze the impact of this ones in the results obtained by the priority rules. Therefore, we have two set where all data is equal, except for the sequence-dependent setup times which were generated with different maximum values: ( $s_{1max} = 0.2 \times p_{max}$  e  $s_{2max} = 0.3 \times p_{max}$ ).

num	SPT	MWKR	SST	SPT2	MWKR2
80	13	8	41	17	8

**Table 10** -  $J/s_{ij}/C_{max}$ : Global results (Taillard' problems) ( $s_{1max} = 0.2 \times p_{max}$ ).

num	SPT	MWKR	SST	SPT2	MWKR2
80	8	5	48	18	6

**Table 11** -  $J/s_{ij}/C_{max}$ : Global results (Taillard' problems) ( $s_{2max} = 0.3 \times p_{max}$ ).

num	SPT	MWKR	SST	SPT2	MWKR2
82	7	14	45	12	16

**Table 12** -  $J/s_{ij}/C_{max}$ : Global results (classical problems) ( $s_{1max} = 0.2 \times p_{max}$ ).



num	SPT	MWKR	SST	SPT2	MWKR2
82	5	9	46	18	10

**Table 13** –  $J/s_{ij}/C_{max}$ : Global results (classical problems) ( $s_{2_{max}} = 0.3 \times p_{max}$ ).

The results are presented in tables 10 to 13. The mix rule SPT2, that considers the processing times and the setup times, and the priority rule SST obtain better results as the maximum setup time increases. Since the other rules, SPT and MWKR, do not consider the setup times, as expected, they do not performed well.

The mix rule MWKR2, that combines the remaining processing time and the setup times, had a similar behavior as the classical rules SPT and MWKR, so the inclusion of the setup times in this rule does not lead to better results.

## 10. Conclusion and future research

In this work, we considered two production scheduling problems: a single-machine and a job-shop scheduling problems with sequence-dependent setup times.

We present several priority dispatching rules for both problems, followed by a study of their performance.

The computational results were designed to analyze the sensibility of the efficiency of the heuristic with respect to: problem dimension and data of the problem.

We can conclude that the Adapted Schrage Heuristic for the single machine scheduling problem, only based on the setup times ( $P=0\%$ ) outperform the remaining ones as the dimension of the problem increases.

In general, the Adapted Schrage heuristic for the problem  $1/r_j, s_{ij}/L'_{max}$  have proved to be useful since it obtains better results than the Schrage heuristic.

For the data of the problem, we can conclude that:

- the value of  $P$  is indifferent to the high values of the release dates.
- for similar values of the processing times, setup times and the delivery times, the heuristic that perform better where for ( $P = 0, P = 20, P = 40$ ).
- for high values of the delivery time, with respect to the remaining data, the heuristics with  $P = 60, P = 80, P = 100$  performed better.

Therefore, we can conclude that to determine the best value of P we should take in account the setup times and the release times. The heuristic perform better as the setup times increases, and the remaining heuristic perform worst.

For the job-shop scheduling problem with sequence-dependent setup times  $J/s_{ij}/C_{max}$ , the computational experiment was designed to evaluate the behavior of the dispatching priority rules with classical rules and some new ones, where the setup times were considered when deciding the next job to be scheduled.

We can conclude that the new priority rule only based in setup time (SST) performs better than the remaining ones. As the setup times increase, this rule performs better. The remaining rules have the opposite behavior, except the SPT2 that also performs well as the setup times increase.

In relation with the dimension of the problem, the SST rule obtains good results as the dimension increases.

As future work, we would like to develop more complex heuristics to improve the quality of the results. We pretend to apply metaheuristics, since the present heuristics can be used to obtain initial solutions. Since several metaheuristics, as the genetic algorithms, tabu search and simulated annealing have been applied successful to other classical scheduling problems, we intend to continue our research by developing such methods to the single-machine and job-shop scheduling problems with sequence-dependent setup times.

We also intend to extend our research, by considering another practical aspect, such as the existence of several identical machines in the job-shop, as a multiprocessor job scheduling in a job-shop environment with sequence-dependent setup times.

## References

Adams J, Balas E, and Zawack D (1988), The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science*, **34**(3), 391-401.

Barman S (1997), Simple priority rule combinations: an approach to improve both flow time and tardiness, *International Journal of Production Research*, **35** (10), 2857-2870.

Beasley J E (1990), OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*, **41**(11), 1069-1072.

Bratley P, Fox B L, and Schrage LE (1983), *A guide to Simulation*, New York: Springer-Verlag.

Bruno J and Downey P (1978), Complexity of Task Sequencing With Deadlines, Setup Times and Changeover Costs. *SIAM Journal of Computing*, **7**, 393-404.

Carlier J (1982), The one-machine sequencing problem. *European Journal of Operational Research*, **11**, 42-47.

Conway RW, Maxwell WL, and Miller LW (1967), *Theory of Scheduling*. Reading, Massachusetts: Addison-Wesley.

Fisher H and Thompson GL (1963), Probabilistic learning combinations of local job-shop scheduling rules. *Industrial Scheduling*. J. F. Muth and G. L. Thompson. (Englewood Cliffs, New Jersey: Prentice Hall: 225-251).

Garey MR and Johnson DS (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.

Grabowski JE, Nowicki E and Zdrzalka S (1986), A block approach for single machine scheduling with release dates and due dates. *European Journal of Operational Research*, **26**, 278-285.

Gupta JND (1988), Single facility scheduling with multiple job classes. *European journal Operations Research*, **8**, 42-45.

Hoogeveen JA, Lenstra JK and van de Velde SL (1997), Sequencing and scheduling: an annotated bibliography, Memorandum COSOR 97-02, Eindhoven University of Technology, Eindhoven, The Netherlands.

Horn WA (1974), Some scheduling algorithms, *Naval Res. Logist. Quart.* **21**, 177-185.

Jeremiah B, Lalchandani A and Schrage L (1964), Heuristics Rules Toward Optimal Scheduling. Department of Industrial Engineering, Cornell University, New York, USA.

Kim SC and Bobrowski PM (1994), Impact of sequence-dependent setup time on job shop scheduling performance. *International Journal of Production Research*, 1503-1520.

Laguna M (1998), A heuristic for production scheduling and inventory control in the presence of sequence-dependent setup times, forthcoming *IIE Transactions*.

Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys D (1993), Sequencing and scheduling: algorithms and complexity. Graves SC, Zipkin PH and Rinnooy Kan AHG (eds.). *Logistics of Production and Inventory; Handbooks in Operations Research and Management Science*, vol. 4, North-Holland, Amsterdam, 445-522.

Lawrence S (1984), Resource constrained project scheduling: an experiment investigation of heuristic scheduling techniques. Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA.

Lageweg BJ, Lenstra JK and Rinnooy Kan AHG (1978), Minimizing maximum lateness on one machine: computational experience and some applications, *Statistica Neerlandica* **30**, 25-41.

Lenstra JK (1977), *Sequencing by Enumerative Methods*. Mathematisch Centrum, Amsterdam, Holland.

Lenstra JK, Rinnooy Kan AHG and Brucker P (1977), Complexity of machine scheduling problems, *Annals of Discrete Mathematics*, **1**, 343-362.

Maccarthy BL and 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.

Mason AJ and Anderson EJ (1991), Minimizing flow time on a single machine with job classes and setup times. *Naval Research Logistics*, **38**, 333-350.

Monma CL and Potts CN (1989), On the Complexity of Scheduling with Batch Setup Times. *Operations Research*, **37**(5), 798-804.

Morton TE and Pentico DW (1993), *Heuristic scheduling systems*, John Wiley & Sons, New York.

Ovacik IM and Uzsoy R (1994), Polling horizon for a single-machine dynamic scheduling problem with sequence-dependent setup times, *International Journal of Production Research*, **32** (6), 1243-1263.

Potts CN (1980), Analysis of a heuristic for one machine with release dates and delivery times. *Operations Research*, **28**, 1436-1441.

Rinnooy Kan AHG (1976), *Machine Scheduling Problems: Classification, Complexity and Computations*. Holland, The Hague, Holland.

Ríos R and Bard J (1997), A branch-and-bound algorithm for flowshop scheduling with setup times, Technical Report ORP97-02, University of Texas, Austin.

Schrage L (1971), Obtaining optimal solutions to resource constrained network scheduling problem, unpublished manuscript.

Storer RH, Wu SD and Vaccari R (1992), New search spaces for sequencing instances with application to job-shop scheduling. *Management Science* **38**, 1495-1509.

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

Williams D and Wirth A (1996), A New Heuristic for a Single Machine Scheduling Problem with Setup Times. *Journal of the Operations Research Society*, **47**, 175-180.

Yamada T and Nakano R (1992), A genetic algorithm applicable to large-scale job-shop instances. *Parallel instance solving from nature*. R. Manner and B. Manderick. (Amsterdam: North-Holland: **2**, 281-290).

Zdrzalka S (1992), Analysis of an Approximation Algorithm for Single-Machine Scheduling with Delivery Times and Sequence Independent Batch Setup Times. Technical University of Wroclaw, Poland.