

# MILP model for the planning of a Computerized Numerical Control Lathes Machining Plant

**Federico Kañevsky**

Universidad Tecnológica Nacional – Facultad Regional Santa Fe  
Santa Fe, Argentina,  
*ficokanievsky@hotmail.com*

and

**María Agustina Franco**

Universidad Tecnológica Nacional – Facultad Regional Santa Fe  
INTEC-UNL-CONICET  
Santa Fe, Argentina,  
*mfranco@frsf.utn.edu.ar*

and

**Maria Rosa Galli**

INGAR-UTN-CONICET  
Santa Fe, Argentina,  
*mrgalli@santafe-conicet.gov.ar*

## **Abstract**

This work introduces the formulation and application of a MILP model to solve the problem of planning the weekly production of a machining plant using numerical control lathes to manufacture spare parts for agricultural machines. The machining plant works under a Flexible Job Shop system and it has reduced workforce with different skills to operate the various high-complexity lathes and to carry out setup operations in each machine. The developed model is based on a basic formulation for the classic problem and we introduce some flexible adjustment for the various situations that may arise from different scheduling problems. The model is applied to various scenarios; and we include a discussion of the improvements brought about by the analysis.

**Keywords:** Flexible job shop; MILP; setup; workforce

## **1 Introduction**

Metallurgical machining plants or companies devoted to producing items in equipment that uses computerized numerical control (CNC) and are characterized by the need of skilled workforce to operate and start up the various pieces of the highly complex equipment involved in their production processes. Machining plants must daily undertake their production scheduling to satisfy clients' demands on settled dates with their scarce resources. This problem becomes even more complex as products are more assorted and manufacturing processes grow in variety. Such is the case of a machining plant that manufactures agricultural machines in the city of Santa Fe, Argentina, which is the object of study in the present work. This company is characterized by a wide range of customized products in relatively small batches. It counts on computerized numerical control lathes that allow accomplishing the various turning tasks involved in manufacturing each spare part. The manufacturing process of each item requires a different set of operations and is accomplished in a particular sequence. Each machine setup and operation (material load and download) require trained workforce, and this is a scarce resource in the company. Not all workers are qualified to carry out all required operations.

It is currently noticed that during the working week, several machines remain without operation for quite long time periods, thus turning the production process quite inefficient. Delivery timelines for finished products are probably affected by constraints that derive from the scarce skilled workforce.

With the aim of analyzing this problem, we compared current timelines for executing production orders with the optimal times provided by planning mathematical models which consider some specific constraints found in the company under analysis.

A classic Flexible Job Shop problem (FJSP) is a sequencing problem in which a set of jobs must be processed in a set of machines where each job consists of a set of operations to be carried out in a preestablished order in any of those machines, including various alternative machines for each operation [1].

Many authors have presented mathematical models to represent either the classic problem or the problem with additional considerations [2]. In general, those formulations do not consider significant set-up times for machines. These times are included in a batch processing time. Some authors have considered sequence-dependent set-up times [3]. Others have considered constraints on two types of resources [4]: overlapping operations [5] or batch division [6]; in general, each characteristic is considered in isolation.

In the company under study, set-up times are very significant and thus they must be carefully considered in the model, even though it is not necessary to consider setup as sequence-dependent. Workers can simultaneously operate two lathes when the processing time for each item is longer than the time required for transportation between lathes. The existence of transfer batches should be also considered, which implies the startup of an operation without completing the whole batch in the preceding operation. Published models do not consider all these conditions simultaneously.

This work presents a reasonably sized mixed integer linear programming model (MILP) that can be adjusted to the introduction of different kinds of constraints, so that the simultaneous consideration of all characteristics of the operation planning problem in the machining plant is feasible without considerably increasing the formulation complexity.

The rest of the paper is structured as follows. Section 2 presents a literature review on the problem. Section 3 presents the notation being used in the model formulation for the classic FJSP problem developed in Section 4. The previous model adjustment to overlapping operations is shown in Section 5. Section 6 discusses the introduction of workforce constraints into the classic problem; and Section 7 addresses sequence-independent set-up times. Section 8 analyzes the characteristics of the various formulations; and Section 9 depicts the application of the most complete formulation to the resolution of the machining plant involving CNC lathes. Section 10 presents a discussion of the results obtained from the real case analysis. Finally, conclusions are drawn in Section 11.

## 2 Literature Review

The FJSP is one of the most difficult-to-solve problems in the production planning area since both job routing and operations scheduling are to be considered. It is well known that this problem is NP-hard. Due to this complexity, much of the existing literature deals with a sort of heuristics and metaheuristics [7,8,9,10,11,12,13].

Even though mathematical programming models cannot compete with metaheuristic tools designed to solve large problems they help to better understand the structure of the problem and are easy to implement, replicate, and modify. As Fanjul *et al.* state [14], MILP models can benefit from continuous improvements in MILP solvers. Mathematical programming formulation is the first key step prior to developing an effective heuristic [15].

The mathematical modeling techniques used to solve production scheduling problems can be classified according to three paradigms: position-based, time-based, and sequence-based, originally proposed by Wagner [16], Bowman [17] and Manne [18], respectively. These variants of modelling paradigms have been separately extended for regular job shop scheduling problems. The main difference between these paradigms lies in the definition of the binary variables used to determine the sequencing problem.

- position-based paradigm:  $Y_{ijhm}$  identifies whether operation  $i$  of job  $j$  is scheduled on the  $h$ -th processing position on machine  $m$ . [1,12,16]
- time-based paradigm:  $Y_{ijmt}$  identifies whether operation  $i$  of job  $j$  is processed by machine  $m$  during time-unit  $t$ . [17, 19, 20]
- sequence-based paradigm:  $Y_{ijhkm}$  identifies whether operation  $i$  of job  $j$  precedes operation  $h$  of job  $k$  (not necessarily immediately) on machine  $m$ . [3, 12, 18]

Demir and Kürsat [15] compared different mathematical models developed under the three paradigms in terms of objective  $C_{mx}$  (minimum completion time for all operations), CPU time, number of variables, and constraints by using the test problems randomly generated by Fattahi *et al.* [1]. They recommended using precedence variable-based models among them, especially those proposed by Özgüven *et al.* for FJSP [21].

From these formulations, some extensions have been developed, which allows considering situations that occur in the industrial practice.

Few authors present MILP models for FJSP considering the possible overlapping of operations [5]. Most papers propose metaheuristic algorithms to solve this kind of problems [5, 22, 23, 24].

Another extension of the FJSP considers that machines and workers represent potential capacity constraints on the shop, which is referred to as a Dual-Resource Constrained Flexible Job Shop Problem (DRCFJSP). It consists of three

subproblems: (i) Assigning operations to machine resources (ii) Assigning operations to worker resources (iii) Sequencing the operations on machines with workers in mind to optimize performance measurement [25]. In [26], the problem is both to determine the problem is formally described using a non-linear integer model and developing a genetic algorithm to solve it. In [27] and [28], a MILP is for the job shop problem with skilled operators without considering the existence of parallel machines. In [29], a MILP multiple criteria is developed considering three issues: company profit, client satisfaction, and the labor environment but for the aggregate production planning problem. The most studied extension has been FJSP with sequence-dependent set-up times [30]. The model presented in [31] served as a basis for other MILP approaches [32, 33]. In [33], an adaptation of the model presented in [32] is proposed for weighting earliness-tardiness minimization and then [14] proposed improvements based on adaptations of subtour elimination constraints. MILP models have not been developed to solve situations where more than two characteristics are added to the classical problem as in the real case presented in this paper.

### 3 Definition of classic FJSP problem and notation

The classic FJSP problem poses the following assumptions:

- 1) the order of operations for each job is fixed and known;
- 2) priorities are not allowed;
- 3) all machines are available at time 0;
- 4) each machine can process only one operation at a time;
- 5) transportation, setup, and release times are negligible; and
- 6) no operation can be interrupted during execution.

The mathematical formulation for the classic FJSP problem proposed in this paper might be associated to the sequence-based paradigm, although the variables used to determine the sequencing of operations do not identify the machines on which they are performed. For this reason, we say that it belongs to the global sequence-based paradigm. This feature allows the model to be easily adjusted to of the addition of extensions of the classical problem. It is also characterized by the use of variables with two indices, unlike other formulations using precedence variables with three or four indices [2].

This feature allows for a formulation with fewer integer variables. This is achieved by enumerating the operations that must be carried out in ascending order; so that, for each job, the numbers assigned to operations specify the sequence in which they are to be accomplished. This characteristic does not remain for operations that correspond to different jobs.

The notations that describe the sets, parameters, and decision variables used in the model are the following:

Sets:

$J$ : set of jobs.

$O$ : set of all operations that make up all jobs of set  $J$ .

$O_{(j)}$ : Subset of operations that make up job  $j \in J$  (are enumerated in ascending order so that the sequence in the subset is specified).

$M$ : Set of machines.

Parameters:

$TProc_{r,m}$ : Processing time for Operation  $r \in O$  in machine  $m \in M$ .

$RM_{r,m}=1$  if operation  $r$  can be carried out in machine  $m$ ; otherwise,  $=0$ .

Continuous decision variables:

$SO_{r,m}$ : Startup time of operation  $r$  in machine  $m$ .

$FO_{r,m}$ : Finishing time of operation  $r$  in machine  $m$ .

$Cmx$ : Makespan (total order completion time)

Binary Decision variables:

$Y_{r,s}=1$  if operation  $r$  precedes operation  $s \neq r$  (not necessarily in an immediate way and not necessarily in the same machine); otherwise,  $=0$ .

$Z_{r,m}=1$  if operation  $r$  is performed in machine  $m$ ; otherwise,  $=0$ .

### 4 Mathematical Formulation I

Under the assumptions and notations presented in the previous section, the problem is both to determine an assignment and a sequence of the operations on all machines that minimizes the total completion time (makespan).

The makespan is the most used objective to optimize, since it is equivalent to minimizing downtime or maximizing the use of machines. Therefore, this is the objective to take into account for the model of this paper. The mathematical formulation we have developed for the classic FJSP problem to be used as a basis for the formulation of more restricted problems is the following:

*Minimize Cmx*  
*subject to*

$$Y_{r,s} + Y_{s,r} = 1 \quad (\forall r, s \in O, s < r). \quad (1)$$

$$\sum_{\substack{m \in M \\ RM_{r,m} = 1}} Z_{r,m} = 1 \quad (\forall r \in O). \quad (2)$$

$$FO_{r,m} \geq SO_{r,m} + TProc_{r,m} \times Z_{r,m} \quad (\forall r \in O, \forall m \in M, RM_{r,m} = 1) \quad (3)$$

$$FO_{r,m} - SO_{s,m} \leq U(1 - Y_{r,s}) \quad (\forall r \in O, \forall m \in M, RM_{r,m} = 1, \forall s \in O, r < s \wedge RM_{r,m} = 1). \quad (4)$$

$$SO_{s,n} \geq FO_{r,m} + U(Z_{r,m} + Z_{s,n} - 2) \quad (\forall j \in J, \forall m, n \in M, \forall s, r \in O(j), RM_{r,m} = 1 \wedge RM_{s,n} = 1 \wedge r < s). \quad (5)$$

$$FO_{s,n} \geq FO_{r,m} + U(Z_{r,m} + Z_{s,n} - 2) \quad (\forall j \in J, \forall m, n \in M, \forall r \in O(j), RM_{r,m} = 1, \forall s \in O(j), RM_{s,n} = 1, r < s). \quad (6)$$

$$FO_{r,m} \leq Cmx \quad (\forall r \in O, \forall m \in M, MR_{(r,m)} = 1). \quad (7)$$

$$\sum_{\substack{r \in O \\ RM_{r,m} = 1}} (TProc_{r,m} \times Z_{r,m}) \leq Cmx \quad (\forall m \in M). \quad (8)$$

$$SO_{r,m} \geq 0 \quad (\forall r \in O, \forall m \in M, RM_{(r,m)} = 1)$$

$$FO_{r,m} \geq 0 \quad (\forall r \in O, \forall m \in M, RM_{(r,m)} = 1)$$

$$Y_{r,s} \geq 0 \quad (\forall r, s \in O, r > s) \quad (9)$$

$$Y_{r,s} \in \{0,1\} \quad (\forall r, s \in O, r < s)$$

$$Z_{r,m} \in \{0,1\} \quad (\forall r \in O, \forall m \in M, RM_{(r,m)} = 1).$$

where:

U is a sufficiently large number.

Constraints (1) represent the feasible order between two operations that may or may not belong to the same job. Constraints (2) indicate that each operation is carried out in a single machine. Constraints (3) impose that if an operation is performed in a particular machine, final time for that operation must not be shorter than the processing time for the operation in that machine. Constraints (4) impose that if two operations are carried out in the same machine, the one that is performed first must finish before the subsequent operation starts.

Constraints (5) and (6) consider the sequence of operations of the same job. Constraints (5) impose that no operation can start until preceding ones have finished, even if they are carried out in different machines. That is, if  $r$  and  $s$  are two operations on the same job  $j$  ( $r, s \in O(j)$ ), and  $r$  must be performed before  $s$  ( $r < s$ ) in any machine  $m$  and  $n$ , respectively ( $Z_{r,m} = Z_{s,n} = 1$ ), the start time of operation  $r$  must be less at the beginning of operation  $s$ . Constraints (6) consider that final time for an operation can be shorter than the final time for the previous operation. Under the same conditions above, the completion time of operation  $r$  must be less than that of operation  $s$ . Constraints (7) determine the makespan (time interval between the beginning of the first job processing and the end of the last job processing), the objective measure to be minimized.

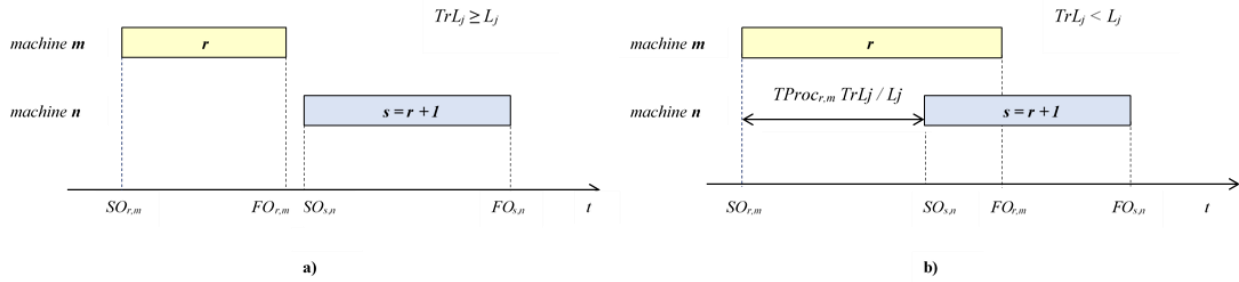
Constraints (8) define that, for each machine, the final processing time of its last operation cannot be higher than the makespan. Avalos-Rosales *et al.* [34] showed that these constraints significantly improve convergence times of branch-and-bound algorithm.

The structure of constraint (1) allows one of the two variables zero-one  $Y_{r,s}$  and  $Y_{s,r}$  to be relaxed and, therefore, be defined as positive instead of binary. The definitions of the variables involved in the model are expressed in the set of constraints (9).

### 5 Formulation II: FJSP with overlapping operations

This work considers that a job corresponds to a production order for a batch of a particular item that must be manufactured in the period to be scheduled.

In the previous formulation, the whole batch is assumed to be processed in a machine before the execution of the subsequent operation. This condition usually becomes highly restrictive, especially in situations in which a large quantity of items is required or processing times are high. In those cases, a certain degree of overlapping is sometimes allowed between successive operations of a same job in different machines. This evidences real situations that can be usually observed in manufacturing companies. The transfer batch is the quantity of items in process that are transported from one machine to another between successive operations. If the transfer batch is smaller than the total batch to be processed, an overlapping of operations occurs. Figure 1 shows a situation in which there is overlapping between operations  $r$  and  $s$  ( $s = r + I$ ) that belong to the same job  $j$ .



**Figure 1:** Overlapping operations: a) Transfer batch larger than production batch. b) Transfer batch smaller than production batch

The model presented in the previous section requires a little adjustment so as to allow considering these situations. Constraints (5) must be replaced by constraints (10) and (11).

Let  $TrL_j$  be the size of Transfer batch and  $L_j$  be the size of Job batch  $j \in J$

$$SO_{s,n} \geq FO_{r,m} + U(Z_{r,m} + Z_{s,n} - 2) \quad (\forall j \in J, \forall m, n \in M, m \neq n \forall r, s \in O(j), \quad (10)$$

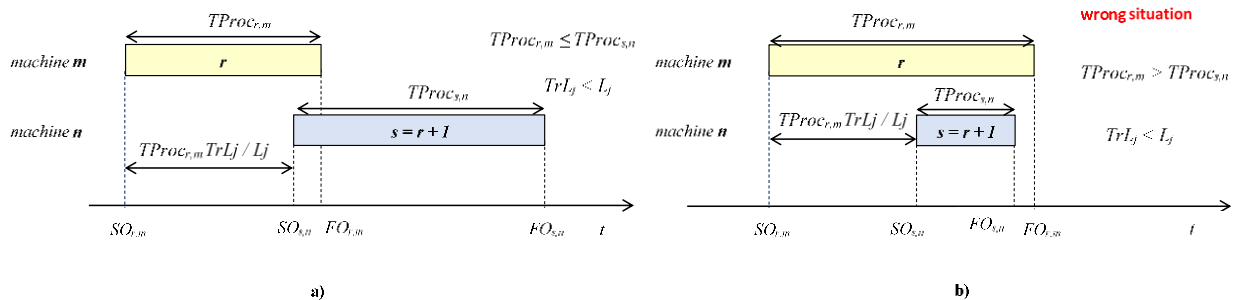
$$RM_{r,m} = 1, RM_{s,n} = 1, r < s, TrL_j \geq L_j).$$

$$SO_{s,n} \geq SO_{r,m} + \frac{TrL_j \cdot TProc_{r,m}}{L_j} + U(Z_{r,m} + Z_{s,n} - 2) \quad (\forall j \in J, \forall m, n \in M, m \neq n \forall r, s \in O(j), \quad (11)$$

$$RM_{r,m} = 1, RM_{s,n} = 1, r < s, TrL_j < L_j).$$

According to constraints (10), if the transfer batch is bigger than the total batch, the whole batch is processed before entering the subsequent machine. Constraints (11) consider that if two operations correspond to the same job but are performed in different machines, the following operation cannot start before the previous operation is finished in the transfer batch.

In addition, the subsequent operation can never finish before the transfer batch is completely processed in the previous operation. This constraint becomes quite significant when the processing speed of the subsequent operation is higher than that of the preceding operation (Fig.2). Constraint (12) reflects this situation.



**Figure 2:** End of overlapping operations: a) correct situation, b) wrong situation

$$FO_{s,n} \geq FO_{r,m} + \frac{TrL_j \ T \ Pr \ oc_{s,m}}{L_j} + U (Z_{r,m} + Z_{s,n} - 2) \quad (\forall j \in J, \forall m, n \in M, m \neq n \ \forall r, s \in O(j)), \quad (12)$$

$$RM_{r,m} = 1, RM_{s,n} = 1, r < s, \ TrL_j < L_j).$$

### 6 Formulation III: FJSP considering workforce

Previous models considered equipment availability as the only constraining resource. In practice, workforce is another resource that is usually scarce. In the available literature, the problems that consider constraints on both resources are known as DRCFJSP (dual-resource constrained flexible job shop problem) [25, 29, 35].

In this section, we will consider both the constrained number of workers and the different skills each worker has, which enable them to operate just some machines.

For the development of this new formulation, the following additional notation should be considered:

Sets:

$W$ : set of workers.

Parameters:

$NW$ : number of available workers.

$WP_{w,m}=1$  if worker  $w$  is trained to operate machine  $m$ ; otherwise,  $=0$ .

$tu$ : time threshold for operating two machines.

Binary decision variables:

$X_{w,r}=1$  if worker  $w$  carries out operation  $r$ ; otherwise,  $=0$ .

$XW_w=1$  if worker  $w$  carries out some activity; otherwise,  $=0$ .

The MILP formulation presented in the previous section can be adjusted to these new scenarios by introducing just a few constraints.

Each operation must be allocated just to one worker:

$$\sum_{\substack{w \in W \\ WP_{w,m}=1}} X_{w,r} = 1 \quad (\forall r \in O), \quad (13)$$

A worker can carry out operation  $r$  only if this operation can be executed in a machine  $m$  that can be operated by that worker.

$$X_{(w,r)} \leq \sum_{\substack{m \in M \\ MR_{r,m}=1 \\ WP_{w,m}=1}} Z_{r,m} \quad (\forall r \in O, \forall w \in W). \quad (14)$$

A worker is required if he carries out one or more operations.

$$\sum_{r \in O} X_{(w,r)} \leq U \times XW_{(w)} \quad (\forall w \in W). \quad (15)$$

The total quantity of required workers cannot surpass the number of available workers.

$$\sum_{w \in W} XW_{(w)} \leq NW \quad (16)$$

In general, a worker cannot operate two machines simultaneously. In some cases, as when operating computerized lathes, they can do it if the unit operation time in each machine surpasses a given threshold ( $tu$ ); for example, due to the transportation from one machine to another one plus loading and downloading times for an item. If this is not the case,  $tu = \infty$ .

$$SO_{s,n} \geq FO_{r,m} + U (X_{w,r} + X_{w,s} + Y_{r,s} + Z_{r,m} + Z_{s,n} - 5) \quad (\forall x \in W, \forall r, s \in O, r \neq s, \forall m, n \in M, m \neq n, RM_{r,m} = 1 \\ RM_{s,n} = 1, (T \ Pr \ oc_{r,m}/L_r) \leq tu, (T \ Pr \ oc_{s,n}/L_s) \leq tu) \quad (17)$$

### 7 Formulation IV: FJSP considering sequence-independent setup times

Machine set-up process is not an added value factor, and thus, we must explicitly consider set-up times and costs while making scheduling decisions to enhance productivity, dispose wastes, improve the use of resources, and meet

deadlines [30]. When set-up times are independent of the sequence of operations, they are usually considered within the batch processing time and the mathematical model does not differ from those previously described. But when there are constraints related to the staff trained to carry out these operations, it is necessary that the models explicitly consider this machine set-up process. In this section, we will present the modifications that must be introduced into Formulations I, II, and III to consider set-up times.

For the development of this new formulation, the following additional notation should be considered:

Parameters:

$Setup_{r,m}$ : setup time of machine  $m$  for operation  $r$ .

$WS_{w,m}=1$  if worker  $w$  is trained to carry out the setup of machine  $m$ ; otherwise,  $=0$

Continuous decision variables:

$SS_{r,m}$ : Starting setup time of operation  $r$  in machine  $m$ .

Binary decision variables:

$XS_{w,r}=1$  if worker  $w$  performs setup of operation  $r$ ; otherwise,  $=0$ .

If operation  $r$  is carried out in machine  $m$ , the start of the operation must be subsequent to the end of the setup process.

$$SO_{r,m} \geq SS_{r,m} + Setup_{r,m} \times Z_{r,m} \quad (\forall r \in O, \forall m \in M, RM_{r,m} = 1), \quad (18)$$

If two operations are carried out in the same machine, the first one must be finished before the start of the subsequent operation setup. Therefore, constraints (4) must be replaced by:

$$FO_{r,m} - SS_{s,m} \leq U(1 - Y_{r,s}) \quad (\forall r \in O, \forall m \in M, RM_{r,m} = 1, \forall s \in O, r < s \wedge RM_{s,m} = 1), \quad (19)$$

With the aim of considering constraints related to workforce trained to perform the various activities, formulation III must include the following constraints:

Each setup process must be allocated to a sole worker.

$$\sum_{\substack{w \in W \\ WS_{w,m}=1}} XS_{w,r} = 1 \quad (\forall r \in O), \quad (20)$$

A worker can carry out the setup of operation  $r$  only if the latter is performed in a machine  $m$  that the worker can set up.

$$XS_{w,r} \leq \sum_{\substack{m \in M \\ MR_{r,m}=1 \\ WS_{w,m}=1}} Z_{r,m} \quad (\forall r \in O, \forall w \in W), \quad (21)$$

A worker is required if he/she performs one or more activities either in machine operation or setup. To consider this more general case, constraints (15) must be replaced by:

$$\sum_{r \in O} (X_{x,r} + XS_{w,r}) \leq U \times XW_{(w)} \quad (\forall w \in W), \quad (22)$$

If the same worker carries out the setup of operations  $r$  and  $s$  ( $XS_{w,r}=XS_{w,s}=1$ ),  $r$  preceding  $s$  ( $Y_{r,s}=1$ ), and if  $r$  is carried out in machine  $m$  ( $Z_{r,m}=1$ ) and  $s$  in machine  $n$  ( $Z_{s,n}=1$ ), the start of setup for operation  $s$  must occur after the setup of operation  $r$  is finished.

$$SS_{s,n} \geq SS_{r,m} + Setup_{r,m} + U(XS_{w,r} + XS_{w,s} + Y_{r,s} + Z_{r,m} + Z_{s,n} - 5) \quad (\forall r \in O, \forall m \in M, RM_{r,m} = 1, \forall s \in O, r \neq s, \forall n \in M, RM_{s,n} = 1, \forall w \in W, WS_{w,m} = 1 \wedge WS_{w,n} = 1), \quad (23)$$

If the same worker performs an operation and the setup of a preceding operation, the former operation cannot start until the corresponding setup is finished:

$$SS_{s,n} \geq FO_{r,m} + U(X_{w,r} + XS_{w,s} + Y_{r,s} + Z_{r,m} + Z_{s,n} - 5) \quad (\forall r, s \in O, r \neq s, \forall m, n \in M, RM_{r,m} = 1, RM_{s,n} = 1, \forall w \in W, WP_{w,m} = 1 \wedge WS_{w,n} = 1), \quad (24)$$

If the same worker performs an operation and the setup of another preceding operation, the former operation cannot start until the corresponding setup is finished:

$$SO_{s,n} \geq SS_{r,m} + Setup_{r,m} + U(XS_{w,r} + X_{w,s} + Y_{r,s} + Z_{r,m} + Z_{s,n} - 5) \quad (\forall r, s \in O, r \neq s, \quad (25)$$

$$\forall m \in M, RM_{r,m} = 1, RM_{s,n} = 1,$$

$$\forall w \in W, WS_{w,m} = 1 \wedge WP_{w,n} = 1)$$

## 8 Formulations assessment

Table 1 summarizes how to obtain the models for the different developed problems by formulating the classic FJSP problem. We used the test problems randomly generated by Fattahi *et al.* [1] to assess the developed models.

**Table 1:** FJSP Models

PROBLEM	MODEL
I- Classic FJSP	<p><i>Minimize Cmx</i>  <i>subject to:</i>                      (1), (2), (3), (4), (5), (6), (7), (8),                      Non-negativity constraints and binary variables</p>
II-FJSP with overlapping	<p><i>Minimize Cmx</i>  <i>subject to:</i>                      (1), (2), (3), (4), (6), (7), (8), (10), (11), (12),                      Non-negativity constraints and binary variables</p>
III- FJSP with Overlapping and restricted workforce	<p><i>Minimize Cmx</i>  <i>subject to:</i>                      (1), (2), (3), (4), (6), (7), (8), (10), (11), (12), (13), (14), (15), (16), (17),                      Non-negativity constraints and binary variables</p>
IV- FJSP with Overlapping, Sequence-independent setup and restricted workforce	<p><i>Minimize Cmx</i>  <i>subject to:</i>                      (1), (2), (3), (6), (7), (8), (10), (11), (12), (13), (14), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25),                      Non-negativity constraints and binary variables</p>

Table 2 depicts the numbers for continuous variables (CVN), binary variables (BVN), and constraints (CN) in each problem by applying Formulation I, which was presented in this work for the classic FJSP problem. It also includes CPU times and the achieved objective value. In the same table, the size of those problems is compared to those of the MILP-1 formulation presented by Özgüven *et al.* [21], which is considered as the most computationally efficient MILP model to solve the classic problem (less CPU time requirement to find the optimal solution) [2][36]. This sequence-based model is presented in Appendix A.1.

All models were solved using solver LINGO 17.0 [37] in a PC with 2.5GHz and 8GB RAM. In all cases, reported CPU times are the average value of 10 runs.

It can be observed that for most test problems, the number of binary variables is lower in Formulation I than in the MILP-1 model, although the number of continuous variables and constraints is greater. Since the number of binary variables is the most influential factor for MILP models [36], it can be said that the proposed model turns out to be a slightly simpler model if compared to the MILP-1 formulation. As regards CPU times, they are very low in both cases, so it can be said that both models have a similar performance with respect to this criterion.

The extension of the classic problem to take into account overlapping operations was only considered in [5]. This position-based model is presented in Appendix B. We have extended the sequence-based model so that we can compare it with the other formulations. This extension is presented in Appendix A.2. Table 3 compares the sizes and CPU times of these two models with those of the model proposed as Formulation II. For the assessment, we used the same test problems as those for the previous case, adjusting the transfer batch to one unit.



**Table 2:** Assessment of Classic FJSP models

Test Problem	$C_{mx}^*$	FORMULATION I				SEQUENCE-BASED MODEL MILP-1			
		CVN	BVN	CN	CPU Time (sec.)	CVN	BVN	CN	CPU Time (sec.)
SFJS1	66	21	12	67	0.18	19	16	42	0,18
SFJS2	107	17	8	59	0.17	15	10	30	0,19
SFJS3	221	33	20	95	0.18	24	26	67	0,51
SFJS4	355	33	20	99	0.17	24	26	67	0,3
SFJS5	119	37	24	129	0.24	28	36	87	0,55
SFJS6	320	58	39	179	0.69	34	39	99	0,5
SFJS7	397	64	45	198	0.83	40	36	93	0,22
SFJS8	253	64	45	215	1.08	40	45	111	0,76
SFJS9	210	64	45	244	1.03	40	55	131	0,7
SFJS10	516	95	70	232	0.74	45	48	124	0,55

As already observed in [15], position-based models are less efficient, especially due to the large number of required binary variables. The size of this model strongly depends on the number of stages considered for each machine. We assume the maximum number of operations possible to be performed on each machine as the number of allowed stages. Taking the transfer batch into consideration does not change the problem size in Formulation II and sequence-based model, but it allows improving the makespan in bigger problems.

**Table 3:** Assessment of FJSP models with overlapping

Test Problem	$C_{mx}^*$	FORMULATION II				EXTENDED SEQUENCE-BASED MODEL				POSITION-BASED MODEL			
		CVN	BVN	CN	CPU Time (sec.)	CVN	BVN	CN	CPU Time (sec.)	CVN	BVN	CN	CPU Time (sec.)
SFJS1	66	21	12	67	0.18	19	16	42	0.18	19	40	129	0.41
SFJS2	107	17	8	59	0.17	15	10	30	0.19	15	22	79	0.32
SFJS3	221	33	20	95	0.18	24	26	67	0.51	21	58	191	2.52
SFJS4	355	33	20	99	0.17	24	26	67	0.3	21	60	197	3.14
SFJS5	119	37	24	129	0.24	28	36	87	0.55	25	84	265	62.68
SFJS6	256	58	39	179	0.69	34	39	99	0.5	31	89	295	15.89
SFJS7	233.5	64	45	198	0.83	40	36	93	0.22	37	84	274	2.5
SFJS8	193	64	45	215	1.08	40	45	111	0.76	37	102	328	105.49
SFJS9	171.7	64	45	244	1.03	40	55	131	0.7	37	132	418	2904.9
SFJS10	419.5	95	70	232	0.74	45	48	124	0.55	56	156	520	4834.21

To evaluate Formulation III, including skilled labor constraints, a maximum of 4 workers were considered: workers assigned even numbers can operate all machines and those assigned odd numbers can operate only machines that were assigned odd numbers. When workforce is constrained, the problem size grows both in integer variables and constraints, which leads to higher CPU times and –as it is expectable– higher makespan values in some situations (Table 4).

Neither model, sequence-based nor position-based, can be easily modified to account for labor constraints. Neither of them handles variables that identify the relative sequence of operations on the different machines. This is a requirement to express the constraints that imply that a worker cannot operate two machines simultaneously. For this reason, for both Formulation III and Formulation IV, comparative studies cannot be made.

**Table 4:** Assessment of FJSP Model with overlapping and workforce constraint

Test Problem	FORMULATION III				
	$C_{mx}^*$	CVN	BVN	CN	CPU Time (sec.)
SFJS1	66	21	32	140	0.39
SFJS2	107	17	28	94	0.4
SFJS3	221	33	48	214	0.44
SFJS4	355	33	48	214	0.46
SFJS5	119	37	52	284	0.51
SFJS6	256	58	79	581	1.18
SFJS7	264.5	64	85	852	43.61
SFJS8	193	64	85	765	4.63
SFJS9	171.7	64	85	774	7.35
SFJS10	457.5	95	122	1513	1230.1

To evaluate formulation IV, every operation was assumed to require a 20-unit setup time –in all cases– on any machine and workers assigned even numbers were assumed to be trained to carry out the setup of all machines. Table 5 shows problem sizes and CPU times for Formulation IV.

**Table 5:** Assessment FJSP model with overlapping, workforce constraint and sequence-independent setup

Test Problem	FORMULATION IV				
	$C_{mx}^*$	CVN	BVN	CN	CPU Time (sec.)
SFJS1	106	37	40	456	0.56
SFJS2	147	31	36	276	0.44
SFJS3	281	55	60	746	1.77
SFJS4	415	55	60	746	0.52
SFJS5	179	61	64	1046	0.77
SFJS6	317	91	97	1829	6.45
SFJS7	315	100	103	2643	1460.01
SFJS8	252.5	100	103	2556	1427.21
SFJS9	227	100	103	2565	35.15
SFJS10	549.7	142	144	4355	1531.20

To analyze the characteristics of the optimal solutions according to each formulation, those corresponding to the SFJS10 test case are shown and those data are specified in Table 6.

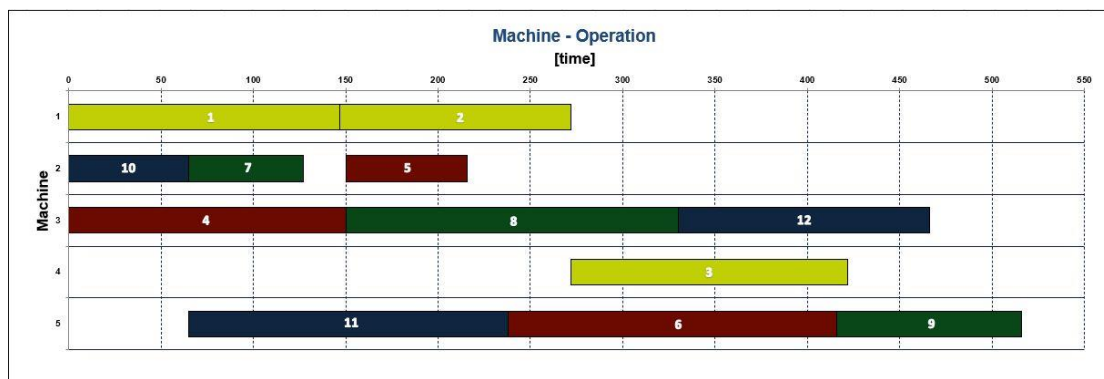
Figures 3 to 6 depict the different solutions found for the particular case of the SFJS10 test problem.

The different solutions found for the particular case of the SFJS10 test problem are shown graphically in Figure 3 to 6. Figure 3 shows an optimal solution found for the classical problem. Each job is represented by a different color and the operation number is displayed within bars.

Operations 5 and 7 –corresponding to jobs 2 and 3, respectively– are represented in the figure with their maximum slack (22 and 23 time units, respectively). Operations 5 and 7 –corresponding to jobs 2 and 3, respectively– have a slack of 22 and 23 time units, and are graphed at their earliest start.

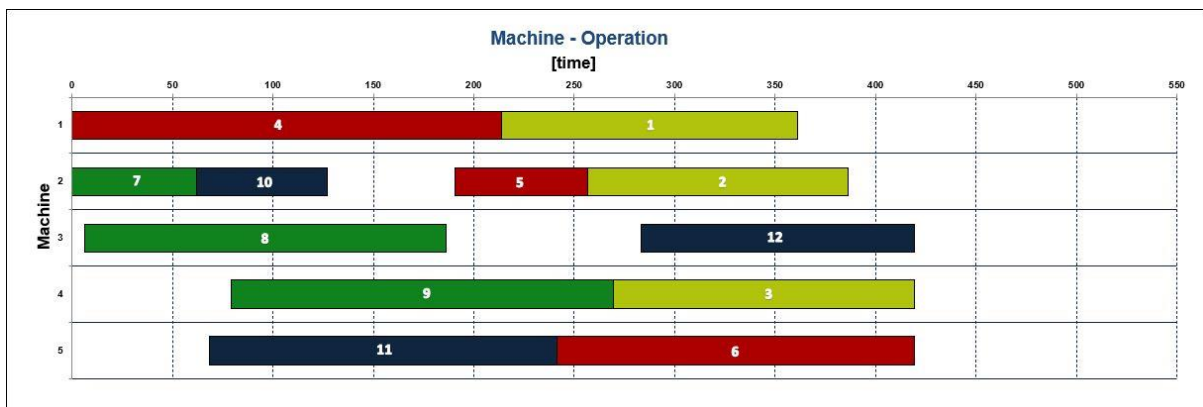
**Table 6:** SFJS10 test problem data

Job	Operation	Machine					L
		1	2	3	4	5	
1	1	147	123	145	0	0	10
	2	125	130	0	140	0	
	3	0	0	0	150	160	
2	4	214	0	150	0	0	10
	5	0	66	87	0	0	
	6	0	0	0	0	178	
3	7	87	62	0	0	0	10
	8	0	0	180	0	0	
	9	0	0	0	190	100	
4	10	87	65	0	0	0	10
	11	0	0	0	0	173	



**Figure 3:** SFJS10 optimal solution Formulation I

Figure 4 shows how the overlapping of operations allows considerably decreasing the makespan. All operations of job 1 overlap: operation 2 can start after operation 1 has been performed on 2.9 units, and operation 3, after operation 2 has been performed on one unit, if it is considered that operations 1 and 2 start at their earliest time, since both present slack. In job 2, operation 5 begins after operation 4 has been performed on 8.9 units, and operation 6 after operation 5 has been performed on 7.7 units. In job 3, operation 8 begins after operation 7 is performed on one unit, and operation 9 begins after operation 8 is performed on 4 units. In job 4 there is only one overlapping operation; operation 11 starts after operation 10 is performed on one unit. Although operation 10 has a slack of 63.5 time units, the entire batch operation cannot start after the earliest time is given in the solution, because operation 11 has no slack and therefore could not delay its start. In this way, the entire order is processed in 419.5 units of time.



**Figure 4:** SFJS10 Optimal solution Formulation II

Figure 5 shows an optimal solution when considering workforce constraints. The numbers within each bar represent the corresponding operation and worker (o; w). In this case, it is not possible for four workers to perform five operations simultaneously, and thus the makespan is increased to 457.5. The threshold time was 20 time units in all cases, which prevents a worker from performing two operations simultaneously. In this solution, only operation 5 has

slack but in this case, it cannot be used to overtake the batch production because it would finish before the preceding operation. In this case, the slack allows considering longer processing times.

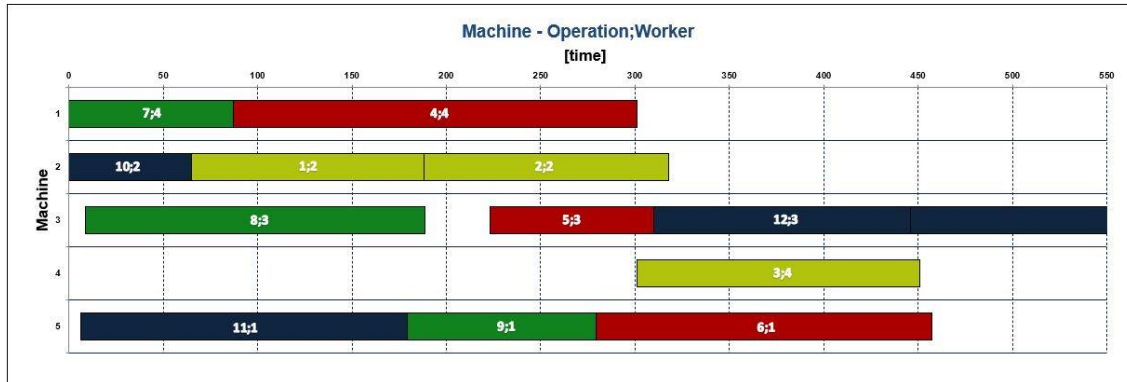


Figure 5: SFJS10 Optimal solution Formulation III

Figure 6 shows an optimal solution when considering that each operation requires a 20-unit set-up time and can be performed by only two of the four available workers (workers 2 and 4). Set-up time for an operation is shown in a similar but lighter color to that of jobs. This last constraint leads to solutions with idle times between the completion of the machine set-up tasks and the start of the operation itself. This is clearly seen in operation 1, which cannot start until worker 4 has finished the setup of operation 5. The time required to process the entire order has increased to 549.7.

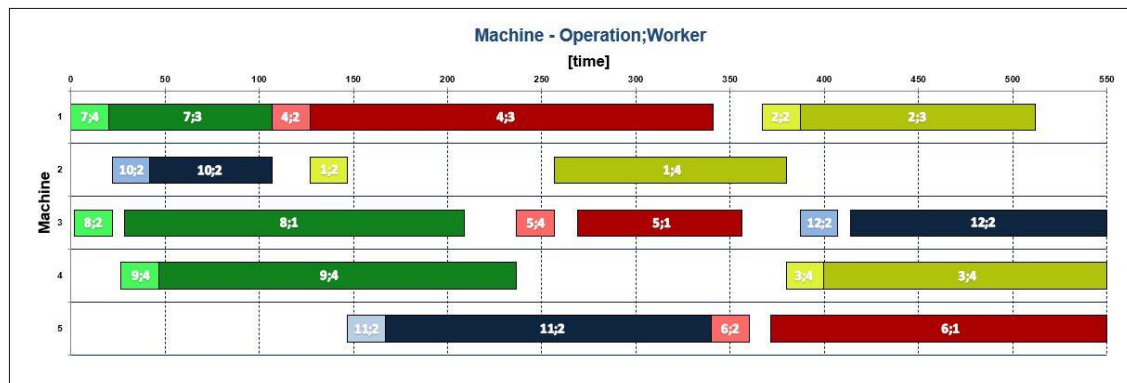


Figure 6: SFJS10 Optimal solution Formulation IV

## 9 Case Study: Computerized Numerical Control Lathes Machining Plant

The engineering company that is studied in this section is devoted to the manufacture of spare parts for tractors and harvesters. The standard manufacturing line comprises around 1000 components: straight and helical gears, bearings, shaft heads, conical gears, and differential gear boxes, among others. The most critical sector in the company is the Lathes sector, which has 6 machines (vertical drill, Pittler parallel lathe, Hyundai CNC lathe, Doosan CNC lathe, Victor Taichung CNC lathe, and Macro Micro Machining Center) and counts on 3 workers. These 3 workers are not all trained to calibrate the lathes, and a fourth worker is allocated to the sector just in exceptional cases.

The production manager has noticed that, even though between 25% and 76% of the main lathes (CNC) that belong to the previously mentioned sector remain idle (see Table7), the amount of material to be processed in the Lathes sector is constant. For that reason, workers' training is being considered so that they can perform tasks in this area. This leads to an improved use of machines and the avoidance of bottleneck situations in this sector of the production process.

**Table 7:** Current efficiency of machines

Nr	Machine	Production [%]	Setup [%]	Dead Time [%]
1	Vertical drill	3.42	1.18	95.4
2	Pittler Parallel Lathe	2.36	3.53	94.11
3	Hyundai CNC Lathe	39.08	36.47	24.45
4	Doosan CNC Lathe	26.92	21.18	51.90
5	Victor Taichung CNC Lathe	40.38	28.24	31.39
6	Macro Micro Machining Center	23.18	0.59	76.24

We present a situation in a typical week (42.5 available hours) as an example. In that week, a ten-item production (jobs) is required, which implies 20 operations, as shown in Table 8.

**Table 8:** Production order in a typical week

Job	Batch size	Required operations
1	100	1 – 2
2	100	3 – 4
3	12	5 – 6
4	87	7 – 8 – 9
5	25	10 -11-12
6	10	13 – 14
7	6	15
8	12	16 – 17
9	96	18 – 19
10	150	20

Table 9 depicts which machines can be used to carry out the various operations ( $RM_m$ ). Every worker can operate the various machines, but only some of them are trained to carry out the setup, as shown in Table 10. Table 11 presents a list of setup and processing times for each operation in each machine.

**Table 9:** Machine that can perform each operation

Operation	Machine						Job
	1	2	3	4	5	6	
1	0	0	1	1	1	0	1
2	0	1	0	0	0	0	
3	0	0	1	1	1	0	2
4	0	0	0	0	0	1	
5	0	0	1	1	1	0	3
6	0	0	1	1	1	0	
7	1	0	0	0	0	0	4
8	0	0	1	1	0	0	
9	0	0	1	1	0	0	
10	0	0	0	0	1	0	5
11	0	0	0	0	1	0	
12	0	0	0	0	1	0	
13	0	0	1	1	0	0	6
14	0	0	1	1	0	0	
15	0	0	0	0	1	0	7
16	0	0	1	1	0	0	8
17	0	0	1	1	0	0	
18	0	0	1	1	0	0	9
19	0	0	1	1	0	0	
20	0	0	0	0	0	1	10

Along this production period, the distribution of operations over time is accounted for by the Gantt diagram shown in Figure 7. Figure 8 depicts the way in which the jobs were distributed for the three workers that carried out the production plan. It is worth noticing that, in several occasions, worker 3 performs setup and processing tasks simultaneously, which leads to a higher time consumption for the execution of certain operations.

We applied the above developed MILP model (Formulation IV) to analyze the current situation and the possible actions to be taken to solve the problem of the company. We considered three possible scenarios: 4, 3, and 2 available workers.

The model for scenario 1 (4 workers) has 370 binary variables and 13278 constraints. It was solved by using solver LINGO 17.0 [37] in a PC with 2.5GHz and 8GB RAM and it required 2114.26 seconds of CPU time. The optimal makespan found was 29.16 hours, which is much lower than the real time used to execute the production order (42.5 hours). The transfer batch being used was made up of 100 items, an average quantity that corresponds to the capacity of the carts used to transport intermediate products. The average time for the transfer of a worker from one machine to another and the load and download of a piece is 0.02 hours (time threshold).

The optimal solution for scenario 2 (3 workers) provided the same makespan value as that for the previous scenario, which proves that a higher number of workers does not mean improved production times. Figures 9 and 10 show the distribution of operations over time and the optimal allocation of workers, respectively. It is evidenced that in the proposed optimal planning no overlapping of tasks occurs when one worker is in charge of a machine setup.

Another considered option was to reduce the number of workers allocated to the sector to 2 (scenario 3). The optimal makespan was found to be 33.84 hours. Figures 11 and 12 show the distribution of operations over time and the optimal allocation of workers for this scenario, respectively. This solution requires one worker to simultaneously carry out more than two operations, which is not a desirable situation in practice.

**Table 10:** Workers trained for setup operations

Worker	Machine					
	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	1	1	1	1	0
3	1	0	1	1	1	0
4	0	0	0	1	1	0

**Table 11:** Batch processing and setup times [hr.]

Operation	Machine												Job
	1		2		3		4		5		6		
	Setup	Proc.	Setup	Proc.	Setup	Proc.	Setup	Proc.	Setup	Proc.	Setup	Proc.	
1	0	0	0	0	2.00	4.35	2	4.35	2	4.35	0	0	1
2	0	0	1.5	1	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	3.8	0	3.80	0	3.80	0	0	2
4	0	0	0	0	0	0	0	0	0	0	0	4	
5	0	0	0	0	1.25	0.50	1.25	0.50	1.25	0.50	0	0	3
6	0	0	0	0	1.25	0.50	1.25	0.50	1.25	0.50	0	0	
7	0.5	1.45	0	0	0	0	0	0	0	0	0	0	4
8	0	0	0	0	2.5	1.96	2.5	1.96	0	0	0	0	
9	0	0	0	0	2.5	1.96	2.5	1.96	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	5.82	0	0	5
11	0	0	0	0	0	0	0	0	4.5	5.82	0	0	
12	0	0	0	0	0	0	0	0	0.5	3.02	0	0	
13	0	0	0	0	6	2.50	6	2.50	0	0	0	0	6
14	0	0	0	0	2.5	3.00	2.5	3.00	0	0	0	0	
15	0	0	0	0	0	0	0	0	7	2.50	0	0	7
16	0	0	0	0	2.5	2.52	2.5	2.52	0	0	0	0	8
17	0	0	0	0	2.5	2.52	2.5	2.52	0	0	0	0	
18	0	0	0	0	0	2.88	0	2.88	0	0	0	0	9
19	0	0	0	0	1.5	1.56	1.5	1.56	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0.25	5.85	10

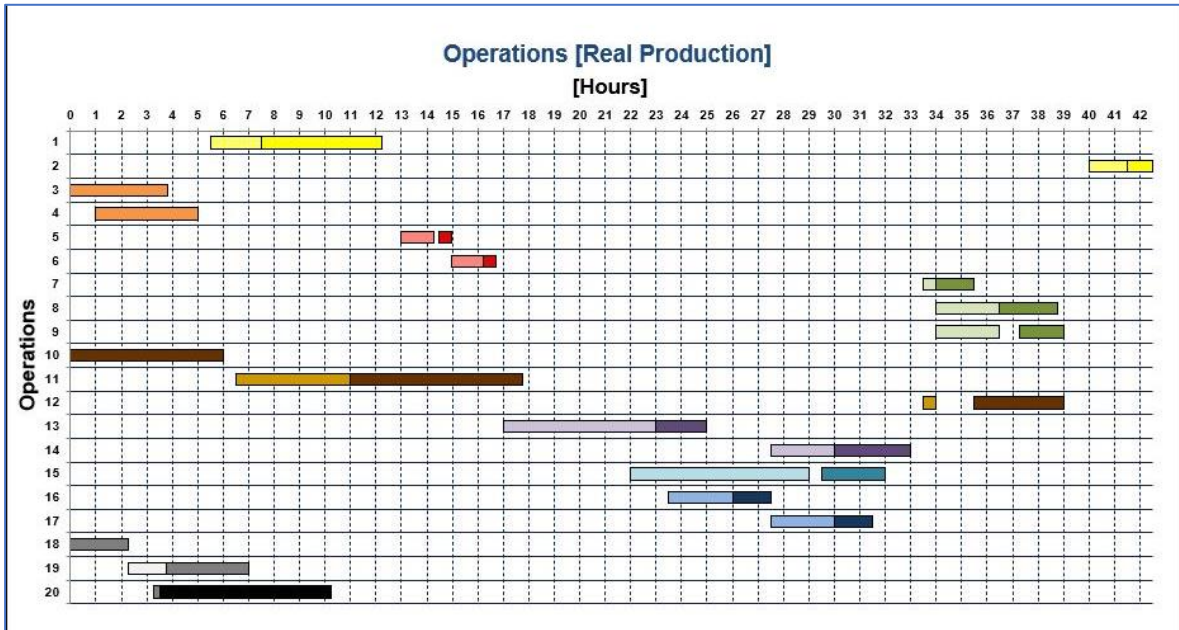


Figure 7: Distribution of operations in real production week

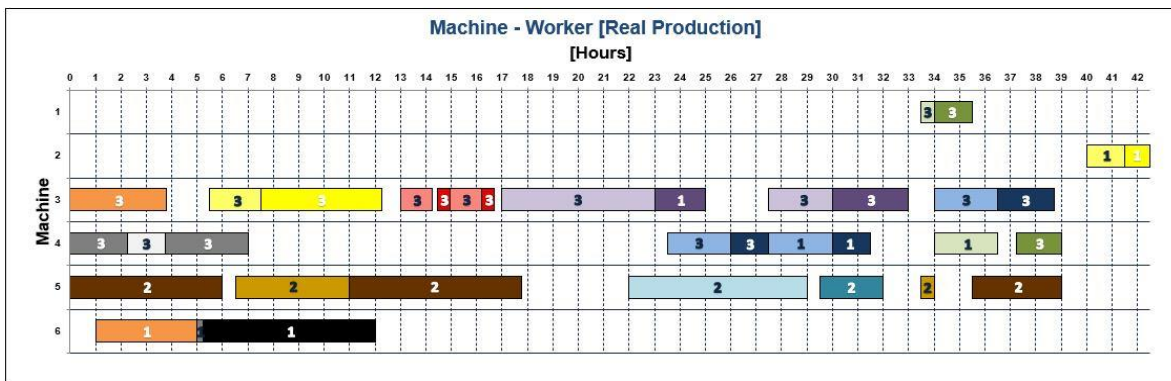


Figure 8: Workers allocation to machines in a real production week

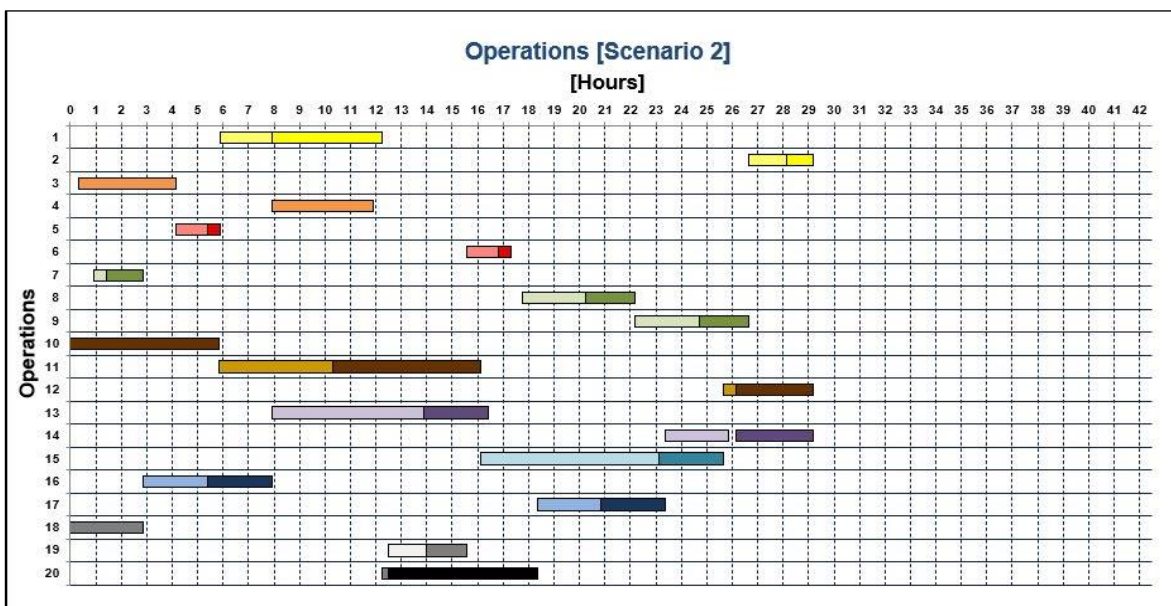


Figure 9: Optimal distribution of operations with 3 workers

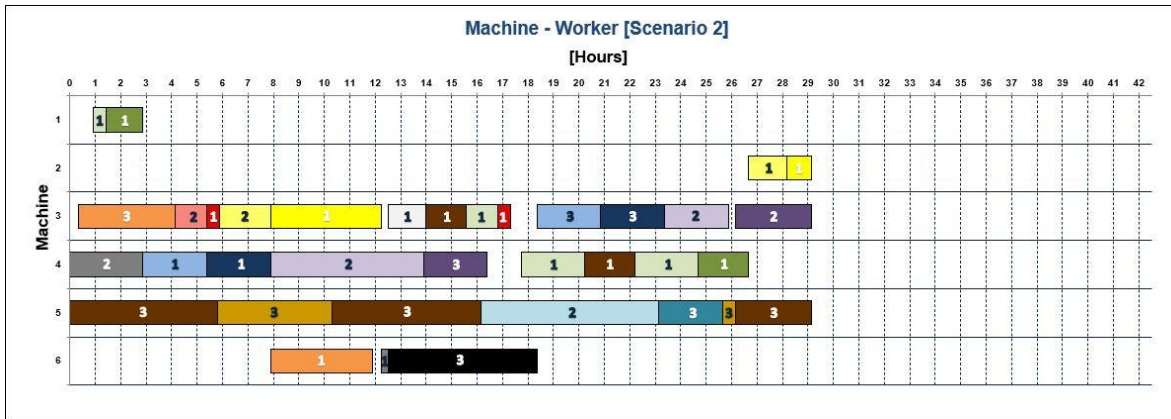


Figure 10: Optimal allocation of workers. Scenario 2

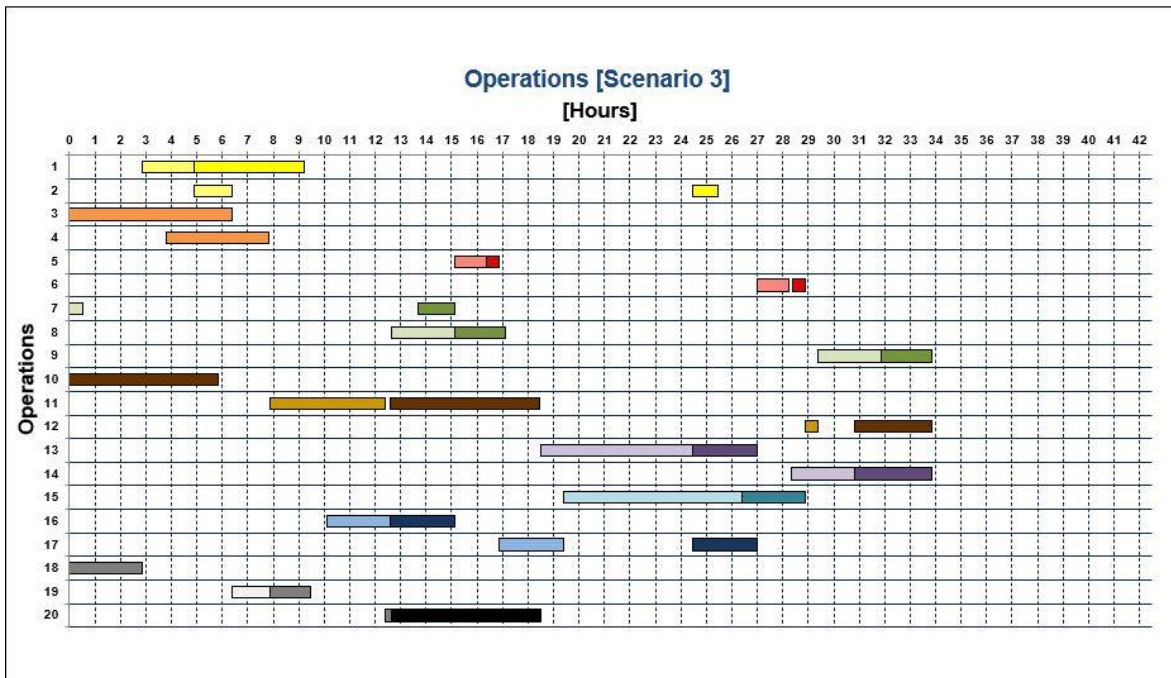


Figure 11: Optimal operation distribution with 2 workers

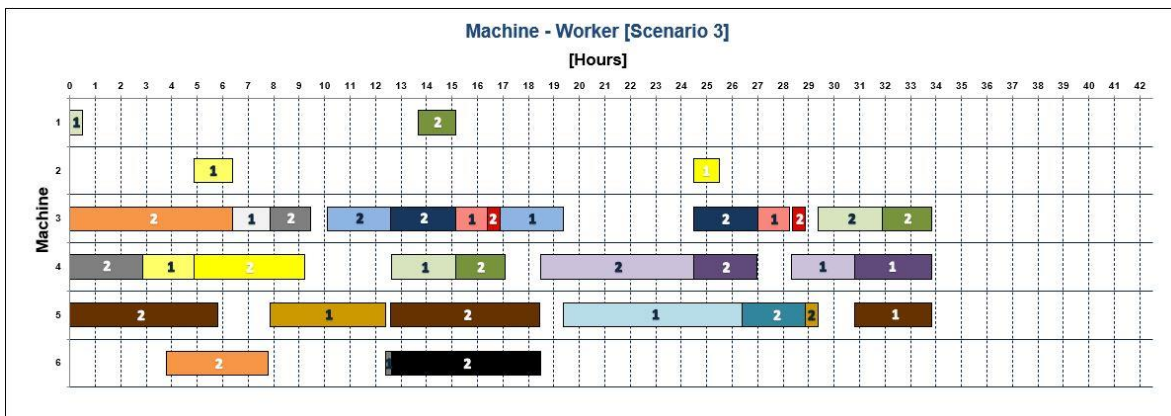


Figure 12: Optimal allocation of workers. Scenario 3



## 10 Result Discussion

The main requirement of the company was to reduce idle times of the main equipment in the lathe sector and it was met in all studied scenarios (see Table 12).

**Table 12:** Comparison of machine use

Nr	Machine	Current Production [%]	Scenario 1 [%]	Scenario 2 [%]	Scenario 3 [%]
3	Hyundai Lathe	76.00	91.21	93.42	81.12
4	Doosan CNC Lathe	45.29	89.03	86.82	81.82
5	Victor Tag. CNC Lathe	72.35	100	100	86.17
6	Macro Micro Machining Center	25.88	34.64	34.64	29.85

Furthermore, in all cases, a more uniform distribution was achieved for the use of machines with similar characteristics, requiring an improved weekly scheduling. The presented model proved to be a valuable and helpful tool for making this kind of decisions. Solutions reveal that the availability of more workers than those currently allocated to the sector is not convenient. Further decreasing the number of workers allows reducing idle times but it does not seem to be an efficient solution since it requires the overlapping of many jobs, which would be detrimental to work quality.

Reducing the makespan implies the decrease in execution times for production orders, and thus, the corresponding reduction in the quantity of products being processed that are accumulated before reaching the CNC Lathes Sector.

## 11 Conclusions

This work has presented a MILP model to solve the classic FJSP problem, whose size and computational efficiency is comparable to the best published models. It serves as a basis for the formulation of problems that consider constraints that are typical of real problems. The main characteristic of this model is that it belongs to a different production scheduling modelling paradigm, which uses a new sequencing binary variable that makes the order of execution of the operations independent of machine assignment.

Test problems were adjusted to analyze the different variations of problems, so that the results presented in this paper will serve as a reference for future benchmarking processes.

The most complete model here presented allowed solving the planning of a lathe machining plant that requires simultaneously considering constraints related to the quantity and skills of the workers to be allocated, overlapping operations, constraining set-up times, and transfer batches. Even though solution times seem to be high, they do not constrain the weekly planning of the company. By applying the present model, the company has been able to reduce idle times in lathes. The model was also used for strategic decision-making that involves decisions on workforce skills. The use of metaheuristics to solve the model will constitute the subject of further research.

## Acknowledgements

The authors acknowledge Universidad Tecnológica Nacional of Argentina and the National Scientific and Technical Research Council of Argentina (CONICET, for its Spanish initials) for their financial support.

## Appendix A

### A.1. Sequence-based model

In this appendix we show the model presented by Özgüven et al. [21] for classical FJSP. This model, denoted MILP-1, uses the following variables:

$X_{i,j,k}$ : 1 if operation  $j$  of job  $i$  precedes operation  $k$  of job  $i$  in machine  $k$ ; otherwise, =0.

$Y_{i,j,g,h,k}$ : 1 if operation  $j$  of job  $i$  succeeds operation  $h$  of job  $g$  on machine  $k$ ; otherwise, =0.

$SO_{i,j,m}$ ,  $FO_{i,j,m}$ : Startup time and finishing time of operation  $j$  of job  $i$  in machine  $m$ , respectively.

$C_i$ : Completion time of job  $i$

The rest of the nomenclature used coincides with that defined in this paper.

From these variables this model is:

Minimize  $Cmx$

subject to:

$$\sum_{\substack{k \in M \\ RM_{j,k}=1}} X_{i,j,k} = 1 \quad (\forall i \in J, \forall j \in O_{(i)}) \quad (26)$$

$$SO_{i,j,k} + FO_{i,j,k} \leq U X_{i,j,k} \quad (\forall i \in J, \forall j \in O_{(i)}, \forall k \in M, RM_{j,k} = 1) \quad (27)$$

$$FO_{i,j,k} \geq SO_{i,j,k} + TProc_{i,j,k} - U(1 - X_{i,j,k}) \quad (\forall i \in J, \forall j \in O_{(i)}, \forall k \in M, RM_{j,k} = 1) \quad (28)$$

$$SO_{i,j,k} \geq FO_{g,h,k} - U Y_{i,j,g,h,k} \quad (\forall i \in J, \forall j \in O_{(i)}, \forall k \in M, RM_{j,k} = 1, \forall g \in J, \forall h \in O_{(g)}, RM_{h,k} = 1) \quad (29)$$

$$SO_{i,j,k} \geq FO_{i,j,k} - U(1 - Y_{i,j,g,h,k}) \quad (\forall i \in J, \forall j \in O_{(i)}, \forall k \in M, RM_{j,k} = 1, \forall g \in J, \forall h \in O_{(g)}, RM_{h,k} = 1) \quad (30)$$

$$\sum_{\substack{k \in M, \\ RM_{j,k}=1}} SO_{i,j,k} \geq \sum_{\substack{k \in M, \\ RM_{j-1,k}=1}} FO_{i,j-1,k} \quad (\forall i \in J, \forall j \in O_{(i)}) \quad (31)$$

$$C_i = \sum_{\substack{k \in M, \\ RM_{j,k}=1}} FO_{i,j,k} \quad (\forall i \in J, \forall j = \text{last element } O_{(i)}) \quad (32)$$

$$C_i \leq Cmx \quad (\forall i \in J) \quad (33)$$

$$SO_{i,j,k} \geq 0, FO_{i,j,k} \geq 0, C_i \geq 0, X_{i,j,k} \in \{0,1\}, Y_{i,j,g,h,k} \in \{0,1\} \quad (34)$$

## A.2. Sequence-based model with overlapping

We extended the sequence-based model to allow for overlapping operations, in which case constraint (31) should be changed to:

$$\sum_{\substack{k \in M, \\ RM_{j,k}=1}} SO_{i,j,k} \geq \sum_{\substack{k \in M, \\ RM_{j-1,k}=1}} FO_{i,j-1,k} \quad (\forall i \in J, \forall j \in O_{(i)}, L_i \leq TrL_i) \quad (32)$$

And the following constraints should be added:

$$SO_{i,j,k} \geq SO_{i,j-1,l} + \frac{TrL_i TProc_{i,j-1,l}}{L_i} + U(X_{i,j-1,l} + X_{i,j,k} - 2) \quad (\forall i \in J, j \in O_{(i)}, j \neq first_i, L_i \geq TrL_i, \forall k \in M, RM_{j,k} = 1, \forall l \in M, RM_{j-1,l} = 1, l \neq k) \quad (33)$$

$$SO_{i,j,k} \geq FO_{i,j-1,k} + U(X_{i,j-1,k} + X_{i,j,k} - 2) \quad (\forall i \in J, j \in O_{(i)}, j \neq first_i, L_i \geq TrL_i, \forall k \in M, RM_{j,k} = 1, RM_{j-1,k} = 1) \quad (34)$$

$$\sum_{\substack{k \in M \\ RM_{j,k}=1}} FO_{i,j,k} \geq \sum_{\substack{k \in M \\ RM_{j,k}=1}} \left( FO_{i,j,k} + \frac{TrL_i TProc_{i,j,k}}{L_i} X_{i,j-1,k} \right) \quad (\forall i \in J, j \in O_{(i)}, j \neq first_i, L_i \geq TrL_i) \quad (35)$$

$$SO_{i,j,k} \geq FO_{i,r,k} + U(X_{i,j,k} + X_{i,r,k} - 2) \quad (\forall i \in J, j \in O_{(i)}, r \in O_{(i)}, r < j, \forall k \in M, RM_{j,k} = 1, RM_{r,k} = 1) \quad (36)$$

## Appendix B

### Position-based model with overlapping

The position-based model we have implemented to perform comparative assessments is based on the formulation of Fattahi *et al.* [1]. This model uses the following variables:

$X_{i,j,m,k} = 1$  if operation  $j$  of job  $i$  is scheduled on the  $k$ -th processing position on machine  $m$  ( $k \in PP_{(m)}$ ); otherwise,  $=0$ .  
 $Y_{i,j,m} = 1$  if operation  $j$  of job  $i$  is performed in machine  $m$ ; otherwise,  $=0$ .

$Tm_{mk}$ : start working time for machine  $m$  on the  $k$ -th processing position.

$Ps_{i,j}$ : processing time of operation  $j$  of job  $i$  after selecting a machine.

The rest of the used nomenclature coincides with that defined in this paper.

Minimize  $Cmx$

subject to:

$$Cmx \geq SO_{i,j} + Ps_{i,j} \quad (\forall i \in J, \forall j \in O_{(i)}) \quad (37)$$

$$\sum_{\substack{m \in M \\ MR_{j,m}=1}} (Y_{i,j,m} + TProc_{i,j,m}) = Ps_{i,j} \quad (\forall i \in J, \forall j \in O_{(i)}) \quad (38)$$

$$SO_{i,j} + Ps_{i,j} \frac{TrL_i}{L_i} \leq SO_{i,j+1} \quad (\forall i \in J, \forall j \in O_{(i)}, j < last_i) \quad (39)$$

$$SO_{i,j} + Ps_{i,j} \leq SO_{i,j+1} + Ps_{i,j+1} - Ps_{i,j} \frac{TrL_i}{L_i} \quad (\forall i \in J, \forall j \in O_{(i)}, j < last_i) \quad (40)$$

$$Tm_{m,k-1} + TProc_{i,j,m} X_{i,j,m,k-1} \leq Tm_{m,k} \quad (\forall i \in J, \forall j \in O_{(i)}, \forall m \in M, MR_{j,m} = 1, \forall k \in PP_{(m)}, k > 1) \quad (41)$$

$$Tm_{m,k} \leq SO_{i,j} + U(1 - X_{i,j,m,k}) \quad (\forall i \in J, \forall j \in O_{(i)}, \forall m \in M, MR_{j,m} = 1, \forall k \in PP_{(m)}) \quad (42)$$

$$Tm_{m,k} + U(1 - X_{i,j,m,k}) \leq SO_{i,j} \quad (\forall i \in J, \forall j \in O_{(i)}, \forall m \in M, MR_{j,m} = 1, \forall k \in PP_{(m)}) \quad (43)$$

$$Y_{i,j,m} \leq MR_{j,m} \quad (\forall i \in J, \forall j \in O_{(i)}, \forall m \in M) \quad (44)$$

$$\sum_{\substack{i \in J \\ j \in O_{(i)} \\ MR_{j,m}=1}} X_{i,j,m,k} \leq 1 \quad (\forall m \in M, \forall k \in PP_{(m)}) \quad (45)$$

$$\sum_{\substack{m \in M \\ MR_{j,m}=1}} Y_{i,j,m} = 1 \quad (\forall i \in J, \forall j \in O_{(i)}) \quad (46)$$

$$SO_{i,j} \geq 0, FO_{i,j} \geq 0, Ps_{i,j} \geq 0, Tm_{m,k} \geq 0, X_{i,j,m,k} \in \{0,1\}, Y_{i,j,k} \in \{0,1\}$$

## References

- [1] P. Fattahi, M. Saidi Mehrabad and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems" . *J Intell Manuf* **18**, pp. 331–342 (2007). <https://doi.org/10.1007/s10845-007-0026-8>
- [2] Y. Demir and S. Kürşat İşleyen, "Evaluation of mathematical models for flexible job-shop scheduling problems," *Applied Mathematical Modelling*, **37** pp. 977-988 (2013). <https://doi.org/10.1016/j.apm.2012.03.020>
- [3] C. Özgüven, Y. Yavuz and L. Özbakır, "Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times," *Applied Mathematical Modelling*, **36** pp. 846-858 (2012). <https://doi.org/10.1016/j.apm.2011.07.037>
- [4] M. Dhiflaoui, H.E. Nouri, O.B. Driss, "Dual-Resource Constraints in Classical and Flexible Job Shop Problems: A State-of-the-Art Review," *Procedia Computer Science* **126** pp. 1507-1515 (2018). <https://doi.org/10.1016/j.procs.2018.08.123>
- [5] P. Fattahi, F. Jolaib and J. Arkatc, "Flexible job shop scheduling with overlapping in operations," *Applied Mathematical Modelling*, **33**, 7 pp. 3076-3087(2009). <https://doi.org/10.1016/j.apm.2008.10.029>
- [6] J. Novas, "Modelo MILP para la programación de la producción en ambientes job-shop flexibles con división de lotes," *Iberoamerican Journal of Industrial Engineering* **8**, 16 pp.56–72 (2017).
- [7] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search", *Ann. Oper. Res.* **41**, 3, pp .157–183 (1993). <https://doi.org/10.1007/BF02023073>
- [8] J.C. Chen, C. Wu, C. Chen and K. Chen, "Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm", *Expert Systems with Applications*, **39**, 11,pp. 10016–10021 (2012). <https://doi.org/10.1016/j.eswa.2012.01.211>
- [9] S. Dauzère-Pérès and J. Paulli, "An integrated approach for modelling and solving the general multiprocessor job-shop scheduling problem using tabu search" *Annals of Operations Research*, **70**, pp. 281–306 (1997). <https://doi.org/10.1023/A:1018930406487>
- [10] S. Jia and Z. Hu, "Path-relinking tabu search for the multi-objective flexible job shop scheduling problem", *Computers and Operations Research*, **47**, pp. 11–26 (2014). <https://doi.org/10.1016/j.cor.2014.01.010>
- [11] I. Kacem, S. Hammadi and P. Borne, "Approach by localization and multiobjective evolutionary optimization for flexible job shop scheduling problems". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* February 2002 <https://doi.org/10.1109/TSMCC.2002.1009117>

- [12] M. Mastrolilli, and L. Gambardella, "Effective neighbourhood functions for the flexible job shop problem". *Journal of Scheduling*, **3**, pp. 3–20. (2000). [https://doi.org/10.1002/\(SICI\)1099-1425\(200001/02\)3:1<3::AID-JOS32>3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1099-1425(200001/02)3:1<3::AID-JOS32>3.0.CO;2-Y)
- [13] V. Roshanaei, A. Azab, and H. ElMaraghy, "Mathematical modelling and a meta-heuristic for flexible job shop scheduling". *International Journal of Production Research*, **51**, 20, pp. 6247–6274 (2013). <https://doi.org/10.1080/00207543.2013.827806>
- [14] L. Fanjul-Peyro, R. Ruiz, F. Perea. "Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times". *Computers & Operations Research*, **101** pp 173-182 (2019). <https://doi.org/10.1016/j.cor.2018.07.007>
- [15] Y. Demir and S. Kürsat İşleyen, "Evaluation of mathematical models for flexible job-shop scheduling problems," *Applied Mathematical Modelling* **37** pp. 977–988 (2013) <https://doi.org/10.1016/j.apm.2012.03.020>
- [16] H.M. Wagner, "An integer linear programming model for machine scheduling", *Naval Res. Logistics Quart.* **6** pp. 131–140 (1959). <https://doi.org/10.1002/nav.3800060205>
- [17] E.H. Bowman, "The scheduling sequence problem", *Oper. Res.* **7** , pp. 621–624 (1959). <https://doi.org/10.1287/opre.7.5.621>
- [18] A.S. Manne, "On the job-shop scheduling problem" , *Oper. Res.* **8** , pp. 219–223 (1960).
- [19] C. S. Thomalla, "Job shop scheduling with alternative process plans." *International Journal of Production Economics* **74** (1-3), pp. 125-134 (2001). [https://doi.org/10.1016/S0925-5273\(01\)00119-0](https://doi.org/10.1016/S0925-5273(01)00119-0)
- [20] M. C. Gomes, A. P. Barbosa-Povoa and A. Q. Novais , "Optimal scheduling for flexible job shop operation." *International Journal of Production Research* **43**, 11, pp. 2323-2353 (2005). <https://doi.org/10.1080/00207540412331330101>
- [21] C. Özgüven, L. Özbakır, Y. Yavuz, "Mathematical models for job-shop scheduling problems with routing and process plan flexibility", *Appl. Math. Modell.* **34** pp. 1539–1548 (2010). <https://doi.org/10.1016/j.apm.2009.09.002>
- [22] R. Alvarez-Valdes, A. Fuertes, J.M. Tamarit, G. Gimenez, and R.S. Ramos, "A heuristic to schedule flexible job shop in a glass factory", *European Journal of Operational Research*, **165** pp. 525-534 (2005) <https://doi.org/10.1016/j.ejor.2004.04.020>
- [23] T. Meng, Q-K. Pan and H.-Y. Sang, "A hybrid artificial bee colony algorithm for a flexible job shop scheduling problem with overlapping in operations," *International Journal of Production Research*, **56**,16, pp. 5278-5292 (2018) <https://doi.org/10.1080/00207543.2018.1467575>
- [24] Y. Demir and S. Kürsat İşleyen "An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations", *International Journal of Production Research*, **52**,13, pp. 3905-3921, (2014) <https://doi.org/10.1080/00207543.2014.889328>
- [25] M. Dhiflaoui, H. E. Nouri, O. B. Driss, "Dual-Resource Constraints in Classical and Flexible Job Shop Problems: A State-of-the-Art Review". *Procedia Computer Science* **126**, pp. 1507–1515 (2018). <https://doi.org/10.1016/j.procs.2018.08.123>
- [26] R. Wu, Y. Li, S. Guo and W. Xu, "Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm". *Adv. Mech. Eng.*, **10** (2018). <https://doi.org/10.1177/1687814018804096>
- [27] N. Al-Hinai and S. Piya, "Jobshop scheduling for skill-dependent Make-to-order system," *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, 2015, pp. 1-5, doi: 10.1109/IEOM.2015.7093814.
- [28] C. López Martínez, H. Hernández Riaño and M. Soto de la Vega. "Un Squirrel Search Algorithm discreto aplicado al problema Job Shop con operadores calificados", *INGE CUC*, **15**, 2, pp. 143–154, (2019). DOI: <http://doi.org/10.17981/ingecuc.15.2.2019.14>
- [29] C. Gomes da Silva, J. Figueira, J. Lisboa and S. Barman, "An interactive decision support system for an aggregate production planning model based on multiple criteria mixed integer linear programming", *Omega: The International Journal of Management Science*, **34**, 2, pp. 167-177 (2006). <https://doi.org/10.1016/j.omega.2004.08.007>
- [30] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times/costs". *European Journal of Operational Research* **246**, pp.345-378 (2015). <https://doi.org/10.1016/j.ejor.2015.04.004>
- [31] A. Guinet., "Textile production systems: a succession of non-identical parallel processor shops". *J. Oper. Res. Soc.* **42**, 8, pp. 655–671 (1991). <https://doi.org/10.1057/jors.1991.132>
- [32] N. Balakrishnan, J.J. Kanet and S.V Sridharan, "Early/tardy scheduling with sequence dependent setups on uniform parallel machines", *Comput. Oper. Res.* **26**, 2, pp. 127–141 (1999). [https://doi.org/10.1016/S0305-0548\(98\)00051-3](https://doi.org/10.1016/S0305-0548(98)00051-3)
- [33] E. Vallada and R. Ruiz, "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times". *Eur. J. Oper. Res.* **211**, 3, pp. 611–622 (2011). <https://doi.org/10.1016/j.ejor.2011.01.011>
- [34] O. Avalos-Rosales, A. Alvarez and F. Angel-Bello, "A reformulation for the problem of scheduling unrelated parallel machines with sequence and machine dependent setup times", in *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling (ICAPS2013)*, pp. 278–282. International Journal of Management Science (2013).
- [35] P. Wirojanagud, E. S. Gel, J. W. Fowler and R. Cardy, "Modeling Inherent Worker Differences for Workforce Planning", *Int. Journal of Production Research*, **45**, 3, pp.525-553 ,(2007). <https://doi.org/10.1080/00207540600792242>
- [36] L. Meng, C. Zhang, Y. Ren, B. Zhang and C. Lv "Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem". *Computers & Industrial Engineering*, **142** (2020). <https://doi.org/10.1016/j.cie.2020.106347>
- [37] LINDO Systems, Inc. <http://lindo.com>