CIGI QUALITA MOSIM 2023
A real-life batch-sizing and sequencing problem with buffer & sequence-dependent setup times

Nicola La Palombara12, Walid Behiri1, Sana Belmokhtar-Berraf1, Chengbin Chu1, and Mustapha Sali3
1Université Gustave Eiffel, France
{nicola.lapalombara, walid.behiri, sana.belmokhtar-berraf, chengbin.chu}@univ-eiffel.fr
2REDLOG SAS, France
{nicola.lapalombara}@planisense.com
3Université PSL-Paris-Dauphine, France
{mustapha.sali}@dauphine.psl.eu

Abstract: This study was inspired by a real-life problem and concerns an injection machine that can produce several types of products, each requiring a specific configuration. Depending on which configuration is set up, the production rate will be distinct, and the minimum batch size is to be respected. The time horizon is one day and it is decomposed into very small time slots. The problem is organizing batch production attempting to meet a given demand by keeping buffer levels within predefined intervals and maximizing resource efficiency. Sequence-dependent setup time and stock coverage are taken into consideration. An integer linear programming (ILP) model is proposed to solve the problem with an original modeling approach that determines the optimal sequence of machine states over the time horizon. We distinguish three possible types of states the machine can hold: production, setup, and idle. The ILP is validated by performing experiments with different-sized instances. Finally, we conclude this study by presenting future extensions to consider parallel machines that share common resources required to perform setup operations.

Keywords: *Single machine, batch-sizing, sequencing, setup times, capacitated buffers*

1. Introduction

Today's environmental and economic issues require all industries to efficiently use resources, whatever their nature. In this context, we note that in several manufacturing systems, significant and incompressible setup times (e.g., machine cleaning or tool changes) are required to move from one product to another. Especially when setup times are sequence-dependent, scheduling jobs is a crucial activity for planners to maximize profits and reduce costs at the same time *(Allahverdi, 2015)*.

For example, plastic injection systems are employed in the automotive industry to efficiently turn raw plastics into differently shaped pieces with complex geometry. Usually, these pieces which are then assembled to obtain finished products, require a proper configuration to be set on the machine, such as a mold to be mounted, material, and color used. In this context, a long setup time is required to prepare the machine and many studies have shown that setup times can be globally reduced by batching, i.e., the choice to produce similar pieces contiguously *(Potts & Van Wassenhove, 1992)*. For instance, to minimize changeovers that cause a loss in productivity, it is preferred that the manufacturing sequence should be followed from light-colored parts to dark-colored parts. For a similar reason, it is requested that raw materials and molds on the machinery should be changed as little as possible. On the other side, demand fulfillment can be improved by having smaller batches, which also reduces throughput time and average inventory levels. The need to fulfill demands at different points in the planning horizon, with different levels of urgency caused by different orders, add complexity to the system. In summary, we have conflicting goals between reducing setup times, i.e., favoring large production lots, and high order fulfillment, ideally achieved by one-piece-flow production.

Many studies address the molding process in terms of the design process, manufacturing, and tuning of the parameters of the molding machines to improve the quality of the manufacturing molds *(Low & Lee, 2008)*. Satisfying the due dates is a matter of satisfaction as well and plays an important role in determining competitiveness in the mold industry. The growing interest in plastic injection processes for both academics and industries is highlighted by the recent review of *Echchakoui and Barka (2020)*, which highlighted the opportunity to improve the decision-making process by applying operational research techniques.

The present study is aligned with this relevance because our contribution is an integer linear programming (ILP) model to deal with a real-life batch sizing and sequencing problem where sequence-dependent setup times are necessary when changing the production from one type of product to another. This problem arises in the automotive industry and concerns an injection machine that is able to produce several types of products, each of which requires a specific configuration to be set on the resource. The problem is to determine the start time and duration of each batch so that an internal demand is met by buffers. Thus, batch production has to be organized in such a way that the buffer levels are kept within a predefined range, set in advance. In addition, it is critical to strive for maximum machine efficiency, which translates into maximizing productivity, when demand is met. This type of problem has been identified to be more difficult to address when resource capacities are tight *(Fleischmann et al., 2005)*.

One of the related problems in the literature is the lot-sizing and scheduling problem (LSSP) which aims to minimize setup, production, and inventory holding costs *(Karimi et al., 2003)*. Typically, this is done within a monthly or weekly planning horizon, but in our case, we are trying to arrange production for the next 24 hours. Many authors have addressed the problem of simultaneous batch size and scheduling by the need to find solutions that can be applied in real-life scenarios *(Ibarra-Rojas et al., 2011)*. In the same vein, we are motivated by the research opportunity to define an efficient scheduling procedure for systems where many resources are involved in manufacturing activities and their efficient utilization is crucial to improve company's performance.

This paper is structured as follows: Section 2 provides a literature review of the problems that present some analogies with the scheduling issue considered in this paper, such as batching and lot-sizing. Section 3 describes the main characteristics of our case study. Section 4 introduces our optimization problem and its mathematical formulation, while Section 5 presents the numerical experiments. Finally, Section 6 concludes this study and discusses future works.

2. Literature review

The studied problem falls on the border of two research domains: lot-sizing and scheduling problems (LSSP) and batch-scheduling problems (BSP). Accordingly, in this section, the most related works are analyzed by highlighting the differences between the studied industrial problem and those considered in the literature, which demonstrates the need for a new optimization model. In parallel, we provide an overview of studies addressing the mold injection process, as being the context of the studied company.

Lot-sizing and scheduling problems are crucial for manufacturing systems with different characteristics such as multiple planning periods, sequence-dependent setup times, and auxiliary assets used for production, among others. For a deeper understanding, readers are referred to the reviews by *Copil et al. (2017)*, *Karimi et al. (2003)*, *Drexl and Kimms (1997)*. According to *Fleischmann and Meyr (1997)*, the lot-sizing and scheduling problem (LSSP) is a tactical problem dealing with the minimization of production and inventory costs by simultaneously optimizing lot sizes and production schedules within a planning horizon lasting several days, weeks, or months. Studies about LSSP on a single machine are paid more attention by both researchers and companies, given their applicability to the real world *(Guimarães et al., 2014)*. In fact, results obtained from these models are commonly used for scheduling bottleneck machines in more complex systems.

By considering the maximum number of setups per period, the formulations for LSSPs can be decomposed into two categories, small-bucket and large-bucket models *(Copil et al., 2017) (Ferreira et al., 2012)*. In the first category, the planning horizon is divided into relatively short intervals or micro periods within which one item can be produced at most. For the large-bucket models, multiple items can be produced during each macro period. *Fleischmann (1990)* proposed a model in which the planning horizon is divided into periods, each of which with a demand to be met, and only one setup per period is allowed. In this sense, the discrete lot-sizing and scheduling problem (DLSP) is a small bucket model with the so-called all-or-nothing assumption. This means, in each period, if production holds it should use the entire period capacity, otherwise the period could not be used to produce at all. Almost all small buckets models consider setup costs, but only a few consider setup times. *Copil et al. (2017)* show that it is probably due to standard formulations, which assume a complete setup to be executed within a single period. However, for short microperiods, this assumption usually does not hold. Then, more complex formulations such as those in *Drexl and Haase (1995)* and *Suerie (2006)* would be required to ensure that a setup can span over multiple microperiods. *Günther (2014)* highlights that in short-time planning horizons, holding costs are negligible and proposes a block planning approach where production events are scheduled first, followed by determining quantities of products. This approach allows as early as possible production using a makespan criterion.

In this study, we take the problem as a whole and this issue may thus be referred to as a batch scheduling problem (BSP). Batching is related to the decisions of whether or not to schedule similar jobs contiguously, to avoid setup times or setup costs *(Potts & Van Wassenhove, 1992)*. Depending on how jobs are released in the system, we classify batch scheduling problems between job availability, which allows each product to be released independently from the other products in the same batch, and batch availability, which states that a product cannot be released before its batch has been fully processed *(Potts & Kovalyov, 2000)*. *Potts and Van Wassenhove (1992)* have proposed a classification of integrated models that combine batch scheduling and lot-sizing. *Jordan and Drexl (1998)* highlight in both DLSP and BSP models, we save setups by batching jobs. In the DLSP, decisions regarding what is to be done are made in each individual period and all parameters are based on the period length, while in the BSP, we decide how to schedule jobs, with a completion time-related objective function. The authors have shown the DLSP can be solved as a BSP if the DLSP instances are transformed.

Considering the injection molding industry, *Wassenhove and Bodt (1983)* was among the first to apply the lot-sizing problem to solve a real-life problem in this field. Studies such as *Ibarra-Rojas et al. (2011)* address the problem of maximizing the production of parts using molds mounted on machines. This study proposes a MILP that determines the batch size of every part and the assignments of parts to mold and machines. An update of this study was proposed by *Rios-Solís et al. (2020)* which aims to solve a lot-sizing and scheduling problem where pieces are produced using molds and then assembled in finished products. The authors indicate that this problem combines both approaches of lot-sizing and scheduling techniques and it is scarcely addressed in the literature. Lot-sizing and scheduling considering the molding context in the automotive sector is addressed by some researchers like *Andres et al. (2021)* and *Diaz-Madronero et al. (2018)* who propose original MILP models to take into account the peculiarities of their case studies, such as stock coverage constraints and backorders, which are recurrent features in this industry. Similar problems arise in other industries like shoe companies, whose requirements and schedules are generally considered by pairs instead of units *(Huang et al., 2012)*. In the same way, *Mula et al. (2021)* considers the bi-part injection molding problem, where a mold has two different cavities that are managed as the same part in their model.

Comparatively to BSP which considers job due dates in the input data, we start from a given demand to simultaneously determine batch sizes and their sequence. Moreover, all of the above models do not take into consideration the presence of decoupling buffers to satisfy demand. Distinct from classical LSSP models whose aim is to fulfill a forecast demand as late as possible (just in time), our work aims at minimizing backorders while trying to maximize machine efficiency. The main difference from lot-sizing lies in the lengths of periods compared with the setup times, which are much shorter in our problem. Indeed setup requires several consecutive time slots to be completed. The same applies to minimum batch production. As seen in this literature review there are no works dealing with so small periods with respect to setup times and minimum batch size. In addition, it is important to note that compared to the LSSP literature, our problem ignores stock holding costs and admits backorders. Furthermore, our problem deals with all-or-nothing assumptions, therefore at each time slot, the decision is whether and which type of product should be manufactured, while the quantity of products produced is determined by the production rate based on the type of product.

To summarize, this study represents a contribution to extending the studies on small bucket production with setup times and minimum batch sizes. Most DLSP models assume that the setup state is lost. We consider setup state conservation, inspired by the industrial application we are dealing with, coherently with the research opportunities identified by *Clark et al. (2014)*. In fact, in real systems, it is often critical to be able to stop batch production for a few moments (e.g., because the buffer is temporarily full) and then restart the same production without requiring additional setup time. The proposed novel ILP considers an objective function that indirectly allows idle times to be minimized, which is a key feature for real cases and has been ignored by previous studies *(Andres et al., 2021)*. In the following sections, we describe our case study and provide a mathematical model that will be useful to understand the features of our approach.

3. Case study

This research deals with a real-life scheduling problem in the context of production lines with a two-stage flow-shop configuration. Each line consists of an injection machine that feeds the downstream assembly process, with a decoupling buffer between them ensuring smooth and continuous operation of the assembly process (as shown in fig. 1).

The injection process is triggered by demand from the assembly phase with an update frequency of 5 minutes (which means that demand for injection changes each 5 min). In addition, the changeovers of the injection machine affect the quality conformity of the first products of a batch. Depending on the type of material and colors that are used, the number of non-conform products is subject to variability. In order to face these two uncertainties the decoupling buffer between injection and assembly is absolutely necessary. Thus, minimal and maximal thresholds are imposed on the buffer level for each product in such a way that the available buffer should always be kept in this interval. The maximal buffer limitation corresponds to a physical space limitation whereas the minimal one is called stock coverage and corresponds to the quantity necessary to meet the demand for the next 2 hours. This work represents the first step in developing a reactive scheduling approach aimed at absorbing demand fluctuation with high computation frequency. There are shared crane resources carrying out mold changes on the injection machines. A setup (or changeover) to prepare the machine for a new batch production can consist of dismounting a mold, mounting a new one, and cleaning the machine (as a trial run for production). The cleaning process is dependent on the change of color and/or material type being injected, as any residual material from the previous production run could contaminate the new batch. In addition, the mold must be heated before starting the injection process, but this operation can be done in masked time.

This work focuses on the injection process, as it feeds the assembly machine that should never stop. Moreover, because each injection resource has a unique set of assigned molds, each product type can be manufactured by only one line. These considerations prompted us to decompose the overall problem into several independent sub-problems, initially assuming shared resources with unlimited capacity. Fig. 1 introduces all the elements and their relationships within the scope of the problem studied. Notice that each type of product stored in buffers requires a proper configuration, which is schematized as a mold-color combination. In addition, each configuration presents a specific production rate, i.e. the number of products released at each time slot when that product type is manufactured. A batch represents the number of identical products that are manufactured over consecutive time slots, using the same production configuration. Once the machine is set up (with high product sequence-dependent setup times related to tools changing and cleaning tasks), the production batch duration is to be determined while respecting a minimum batch size (depending on the product type), due to the technical requirements of the molds. Essentially, the studied problem turns out to be a batch sizing and sequencing problem. The considered time horizon is 24 hours, which is divided into 5 minutes time slots.

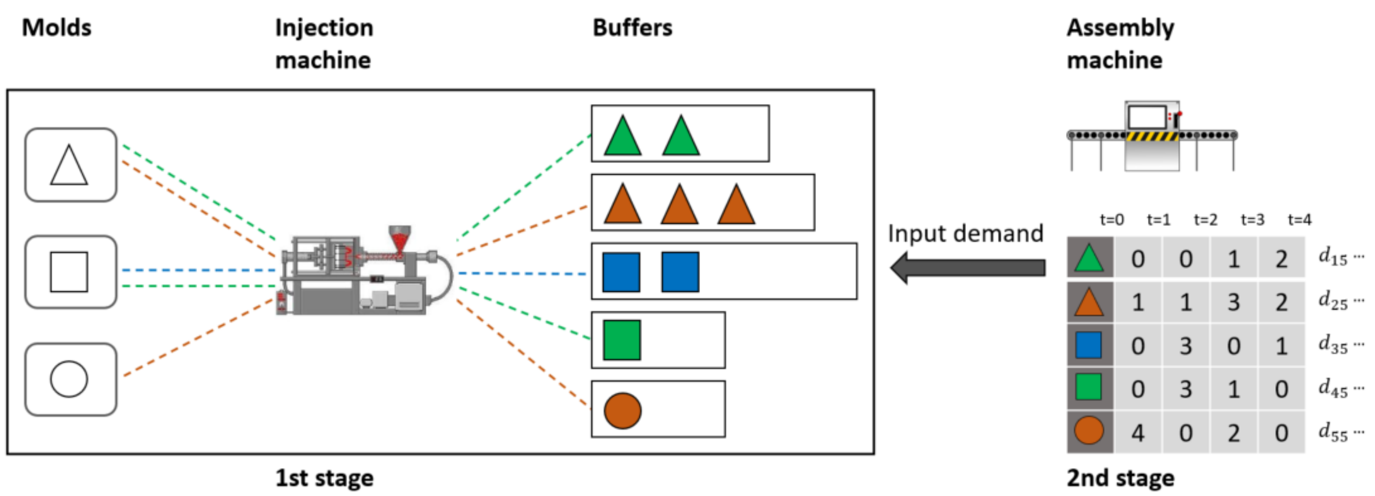

Fig. 1: A schema of the plastic injection process

Currently, the company uses a manual spreadsheet procedure based on Kanban control for scheduling the batch needed based on real consumption. This process is extremely time-consuming as re-scheduling is often needed and batch sizes must often be adjusted manually in case of unexpected events. Batch-size decisions are often based on human experience and empirically determined. It typically consists of a standard 4 hours batch production. In addition, replenishment based on the current buffer level without considering future consumption does not allow any anticipation. Indeed, to avoid backorders, schedulers are often forced to order urgent production batches or to fill buffers with additional stock. Both cases represent an extra cost for the company.

It is therefore evident there is a need for an efficient method to determine a work schedule that can maximize demand fulfillment while considering all the particular conditions described above.

4. Mathematical formulation

In this section, to address all the characteristics and objectives of this case study, a simultaneous batch-sizing and sequencing model is formulated using a linear programming model aiming at minimizing backorders and coverage stockouts while maximizing machine efficiency.

A. Problem description and assumptions

This model considers K physical products, each of them requiring a specific configuration to be set on the machine, over a discrete time horizon $t \in [1, T]$. From a modeling point of view, the terms configuration and product are synonymous since in practice there is a unique correspondence between them. Setups between products are known beforehand and are sequence-dependent because cleaning operations depend on the change of material and color injected between one batch and the next one. There are a total of $K * (K-1)$ possible setups which include the time to remove a mold, mount a new one and clean the machine. In addition, we consider K fictitious dummy products which are used to indicate an idle state of the resource even though a certain configuration is set. Coherently, we define $j \in J / J = [1, K^2 + K]$ the set of all the possible states allowed on the resource where J is composed of:

- $J^k = [1, K]$ is the set of production states,
- $J^c = [K+1, K^2]$ is the set of setup states,
- $J^i = [K^2+1, K^2+K]$ is the set of idle states.

To be more explicit, we use the index k when j is in J^k. Let $d_{k,t}$ be the demand for item k, which has to be fulfilled from buffer at time slot t. For each product type k, $BMax_k$ represents the buffer capacity, while $BMin_{k,t}$ represents the stock coverage required at time t which is calculated as the demand of the next 2 hours. Moreover, $b_{k,t}$ represents the buffer level k at time t. In this model, backorders are allowed, while holding costs are ignored by the company given the considered short time horizon *(Günther, 2014)*. The nominal production rate q_k represents the number of products k that can be produced and released in the buffer at each time slot, without waiting for the completion of the batch, if the product k is running on the machine. If at time t no production takes place but the machine was already set up for a product k, i.e., the machine is in an idle state, then the setup state is conserved, meaning no setup time should be required to continue producing k after. A binary matrix $F_{(J \times J)}$ is introduced to define the rules of the machine states' transition. If $f_{j,j'} = 1$ then for two successive time slots the transition from the state j to the state j' is allowed. On the contrary, if $f_{j,j'} = 0$ it would be forbidden to switch from state j to state j' for the machine. In this model, a batch corresponds to a consecutive number of time slots with the same production state $j \in J^k$ that is activated on the machine such that a minimum batch size must be respected. The number of products in a batch result from multiplying the production coefficient q_k by the number of consecutive time slots a production state is activated. The vector v_j represents the minimum number of time slots associated with state $j \in J$. It contains the information for minimum batch size and setup times when we refer to a production or setup state, respectively. However, $v_j = 1$ for all the idle states $j \in J^i$.

Table I. Nomenclature of the model

Indexes and sets	
i, j	Indexes of machine states
t	Index of time slots
J^k	Set of production states
J^c	Set of setup states
J^i	Set of idle states
J	Set of all possible machine states
Data	
$d_{k,t}$	Demand, i.e., the requirement for product k at time t
v_j	Minimum number of consecutive time slots the state j must be active
q_k	Production rate for product k
$F_{j,i}$	Binary matrix which regulates machine states transition
$BMax_k$	Maximum buffer capacity for product k
$BMin_{k,t}$	Stock coverage level of product k at time t
$I_{k,0}$	Initial buffer level for product k
i_0	State of the machine before solving the problem
θ_h	The unitary weight of backorders
θ_w	The unitary weight of coverage stockouts
θ_b	The unitary weight of products in the buffers
Decision variables	
$x_{j,t}$	Binary variable which is 1 if state j is active at time t, 0 otherwise
$h_{j,t}$	Backorders of $j \in J^k$ at time t
$w_{j,t}$	Units of coverage stockouts of $j \in J^k$ at time t
$b_{j,t}$	Buffer level $j \in J^k$ at time t

The optimization problem deals with the determination of the optimal machine state at each time slot. In fact, at each time slot, a single state is activated by a binary variable $x_{j,t}$, which is 1 if state j is activated at time t, 0 otherwise. Finally, the variables $h_{k,t}$ represent the backorders of product k at time t which we will try to satisfy at time $t+1$, and the variables $w_{k,t}$ represent the number of products k missing from the buffer to reach the desired stock coverage level. All relevant data for the scheduling process are assumed to be deterministic which is justified by having a very short-term operational problem on hand.

B. Integer Linear Programming formulation

In order to build a mathematical model for the problem described, we use the notation summarised in Table I. Thus, the ILP model can be formulated as follows:

$$MIN\Big(\sum_{t \in T}\sum_{j \in J^k}(\theta_h h_{j,t} + \theta_w w_{j,t}) \quad - \sum_{j \in J^k}\theta_b b_{j,T}\Big) \tag{1}$$

$$\sum_{t=1}^{1+v_j} x_{j,t} \;\geq\; v_j x_{j,1}, \qquad \forall j \in J \tag{2}$$

$$\sum_{t'=t}^{t+v_j} x_{j,t'} \geq v_j(x_{j,t} - x_{j,t-1}),$$
$$\forall t \in \{2, \ldots, T - v_j\}, \quad \forall j \in J \quad (3)$$

$$\sum_{t'=t}^{T} x_{j,t'} \geq (T - t)(x_{j,t} - x_{j,t-1}),$$
$$\forall t \in \{T - v_j + 1, \ldots, T\}, \quad \forall j \in J \quad (4)$$

$$\sum_{j \in J} x_{j,t} = 1, \qquad \forall t \in \{1, \ldots, T\} \quad (5)$$

$$\sum_{j \in J} f_{i_0,j} x_{j,1} \ \geq \ 1, \qquad (6)$$

$$\sum_{i \in J} f_{i,j} x_{i,t-1} \geq x_{j,t} - x_{j,t-1},$$
$$\forall t \in \{2, \ldots, T\}, \quad \forall j \in J \quad (7)$$

$$b_{j,1} \ = \ I_{j,0} + q_j x_{j,1} - d_{j,1} + h_{j,1}, \qquad \forall j \ \in \ J^k \quad (8)$$

$$b_{j,t} = b_{j,t-1} + q_j x_{j,t} - d_{j,t} + h_{j,t},$$
$$\forall t \in \{2, \ldots, T\}, \quad \forall j \in J^k \quad (9)$$

$$w_{j,t} \geq BMin_{j,t} - b_{j,t}, \qquad \forall t \in \{1, \ldots, T\}, \quad \forall j \in J^k \quad (10)$$

$$b_{j,t} \ \leq \ BMax_j, \qquad \forall t \ \in \ \{1, \ldots, T\}, \quad \forall j \ \in \ J^k \quad (11)$$

$$x_{j,t} \ \in \ \{0, 1\}, \qquad \forall t \ \in \ \{1, \ldots, T\}, \quad \forall j \ \in \ J \quad (12)$$

$$h_{j,t}, w_{j,t}, b_{j,t} \in N, \qquad \forall t \in \{1, \ldots, T\}, \quad \forall j \in J^k \quad (13)$$

The objective function (1) minimizes backorders and coverage stockouts while maximizing buffer levels. In other words, the objective function tries to meet two conflicting criteria as demand satisfaction and resource efficiency. It should be noted that this function enables the reduction of setup and idle states. The weights calibration of the three factors contained in the objective functions allowed us to express their relative importance.

Constraints (2), (3) and (4) make sure the continuation of a certain state for a minimum number of consecutive time slots to ensure the minimum batch size and the respect of setup times. Constraints (5) ensure the activation of one and only one state at each time slot. Constraints (6) and (7) regulate the machine states transition, according to what is the initial state and what has been defined in matrix F. We are saying a state is always preceded by a compatible state. Constraints (8) and (9) define the flow conservation constraints that describe the evolution of buffer levels over the considered time horizon. Constraints (10) define coverage stockouts, i.e., the stock coverage quantity used at each time slot to meet demand. Constraints (11)-(13) define the range of variables. The objective is to find a sequence of states to obtain a production schedule that minimizes (1).

The computational complexity of this problem can be evaluated by considering the following special case:

- all minimum batch sizes are equal to 1: $v_j = 1 \quad \forall j \in j^k$
- $\theta_w = \theta_b = 0$
- all initial buffer levels are equal to 0: $I_{j,t} = 0 \quad \forall j \in j^k$

Here, the problem of batch sizing and sequencing is reduced to a single-machine scheduling problem, with job due date and setup times and the objective is to minimize the number of late jobs. Note that jobs are obtained by transforming input demand. The complexity of scheduling problems with batch setup times is investigated by *Bruno and Downey (1978)* and *Monma and Potts (1989)*, which have shown the feasibility problem is NP-hard if setup times are not zero.

Because the sequencing problem, even for a single machine with sequence-dependent setups and makespan as the objective is equivalent to the traveling salesman problem (TSP) which is NP-hard *(Pinedo, 2012)*, the studied problem is therefore also NP-hard.

5. Computational experiments

In this section preliminary experiments are carried out to validate the ILP and evaluate the model performances in terms of running time. Indeed, to determine whether our proposal can effectively optimize industrial problems in a reasonable amount of time, we aim to investigate the effect of dataset size on computational results. The following section shows how data are generated, while the results of computational experiments are presented in the final section.

A. Data generation

We randomly generated instances that combine the number of products K and the number of time slots T since these factors specifically define the size of our problem, i.e., the number of decision variables and constraints of a class of instances. A total of 250 instances are considered by randomly generating 25 instances for each class analyzed. Table II is used to report the details corresponding to each instance class.

Concerning the other parameters of the problem, their generation is independent of each class of instances and it has been implemented as follows:

- The demand of a product k in a given time slot t, $d_{k,t}$, is an integer random number with a uniform distribution within the interval $[0, 4]$;
- The setup time needed to switch production from one type of product to another is an integer that is uniformly distributed between $[4, 8]$ time slots;
- Similarly, minimum batch sizes are expressed in integer numbers of consecutive time slots, uniformly distributed within the interval $[10, 14]$;
- Parameters v_j when $j \in J^i$, involved in the definition of idle states, is set to one;
- The production rate for product k, q_k, is a random integer in the interval $[4, 7]$;
- Maximum buffer capacities are integers uniformly distributed belonging to the interval $[150, 300]$;
- The initial buffer levels are integers uniformly distributed within the interval $[0, 100]$.
- Stock coverage is set as 24 time slots of demand (2 hours);
- i_0, the machine state before starting the simulation, is an integer between $[1, K^2 + K]$;
- The weights of parameters in the objective function, θ_h,θ_w and θ_b are respectively set to 100, 1 and 0.1.

Table II. Preliminary results of the analyzed instances

Class	Number of products	Number of time slots	Number of variables	Number of constraints	CPLEX time (sec)	Standard deviation
1	3	72	1515	3949	2.04	1.77
2	4	72	2308	6101	5.62	5.68
3	5	72	3245	8683	10.64	11.88
4	6	72	4326	11695	18.81	21.47
5	7	72	5551	15137	35.27	54.57
6	8	72	6920	19009	47.81	65.55
7	5	90	4055	10861	55.61	70.50
8	6	90	5406	14629	211.34	246.35
9	5	108	4865	13039	436.40	453.37
10	6	108	6486	17563	993.26	1081.13

B. ILP numerical results

Computational experiments have been carried out to assess the performances of the proposed ILP using CPLEX 12.5.1. These experiments are run on a personal computer Dell Inc. 2022 with a Processor 12th Gen Intel(R) Core(TM) i5-1235U, 1300 MHz, 10 Core Logical Processor(s), and 16 GB RAM. The model is coded in Java using Eclipse IDE. Table II presents the experimental results obtained. Each row of this table shows the average indicators over the 25 solved instances for each class. The first five columns give the parameters of each class. Column "CPLEX time (sec)" refers to the average time in seconds that CPLEX took to solve ILP, while column "Standard deviation" represents the dispersion of computation times. For all the analyzed instances, the solver found the optimal solution.

Overall, we observe that calculation times increase as K and/or T increase. However, the computation time is not strictly related to the size of the class, in terms of the number of variables and constraints generated (for example, classes 6 and 10). This result is partly related to the intrinsic nature of the optimization problem. Over a limited time horizon with respect to the number of products to be manufactured, as we mainly minimize backorders, there are fewer interesting combinations of sequences and batch sizes to be made. In such cases, the minimum batch size parameter, which is randomly generated in the same way for all classes, may allow a faster convergence on the best sequence of machine states.

As we can see in Table II, for the 72 time slot instances, the ILP model achieves optimal results at a maximum of 48 seconds on average. Then, considering larger time horizons, we note a consistent increase in the computation time required to find optimal solutions. The largest class analyzed requires on average more than 15 minutes to be solved. This can be explained by the fact that adding one product while considering the same relatively large horizon, considerably increases the possible sequences and batch size combinations. Finally, the large values for standard deviation may be due to the non-negligible range of uniform distributions used and also it shows that probably not all parameters generated regardless of class size have the same impact on computational complexity.

These results represent a preliminary experiment that should be deepened and they are significant in understanding the real possibilities of implementing an optimal model in real-world scheduling systems, where usually we have to deal with larger instances.

Additionally, in Table III we analyzed a class of instances that replicates the real-world data managed by the company on a daily basis. We randomly generated 75 instances considering 24 hours time horizon (288 time slots) and an integer number of products that are uniformly distributed between [8, 16]. All the other parameters are identically generated as the before-mentioned instances. Moreover, a time limit of 60 minutes was set for the solver and an average duality gap equal to 31.18% was obtained.

Table III. Computational results for real-world size instances

Number of products	Number of time slots	Cplex time (sec)	Gap (%)
[8,16]	288	3600	31.18%

From this experimental section, we conclude that the ILP model is able to solve small and medium instances within a reasonable amount of time. Nevertheless, regarding the operational context related to the high rescheduling frequency which is 5 minutes, a heuristic should be developed in order to efficiently solve the problem.

6. Conclusions & future works

In this paper, a real-world scheduling problem from a plastic injection company has been studied and solved. Buffer capacity, stock coverage, minimum batch sizes, different production rates, and sequence-dependent setup times lead to a very challenging optimization problem. A relevant feature of our problem concerns the length and the granularity of the time horizon. Given the internal demand to be met every 5 minutes, we divided the daily time horizon into 5 minutes time slots.

We propose a novel Integer Linear Programming (ILP) formulation that determines the optimal sequence of machine states, aiming at minimizing backorders and coverage stock-outs while maximizing resource efficiency. We distinguished three possible states to define the global schedule: production, setup, and idle states. The main contribution of this model compared to others found in the literature is that it considers a simultaneous batch and sequencing problem within a very short time horizon with very small time slots and both setups and minimum batches extend over several consecutive time slots. The first experiments were carried out using CPLEX solver to optimally solve small to medium size instances within some minutes. For real-size instances, we set 60 minutes time limit, and the average duality gap obtained leads us to conclude that efficient heuristics should be developed to obtain near-optimal solutions. The main limitation of this work lies in the absence of comparison with other existing models and the reduction of some practical aspects of the real case.

As the first step for future works, we believe that numerical experimentation should be further investigated, as the preliminary results obtained are worth further understanding, given the specificity of our problem. In particular, it would be interesting to include a sensitivity study on important parameters or the weight of binary variables in this model. Given the problem's practical implications, we consider that an interesting research direction is to implement multi-objective hierarchical functions that can better prioritize industry goals. Afterward, we will develop a meta-heuristic approach to deal with large instances based on real data. The literature shows that genetic algorithms, Lagrangian relaxation, and dynamic programming represent excellent approaches for dealing with such constrained systems *(Wolsey, 2002)*, *(Copil et al., 2017)*. Moreover, incorporating demand (or process) uncertainty while considering that scheduling in practice has to be done on a rolling horizon basis is the main purpose of this project. As future work, we also aim to extend our study to parallel machine environments, that is, contemplate the existence of common cranes and operators for setup operations. A further extension of the model is considering multipart molds, that is, simultaneously processing products of different shapes. In literature, some bi-part injection molding problems were found but to the best of our knowledge, there is no work dealing with more than 2 references produced at the same time *(Mula et al., 2021)*. In conclusion, this study represents the first step towards the development of a user-friendly Decision Support System (DSS) to improve the company scheduling activities and to address other realistic industrial problems.

References

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, *246*(2), 345–378.

Andres, B., Guzman, E., & Poler, R. (2021). A novel milp model for the production, lot sizing, and scheduling of automotive plastic components on parallel flexible injection machines with setup common operators. *Complexity*, *2021*.

Bruno, J., & Downey, P. (1978). Complexity of task sequencing with deadlines, set-up times and changeover costs. *SIAM Journal on Computing*, *7*(4), 393–404.

Clark, A., Mahdieh, M., & Rangel, S. (2014). Production lot sizing and scheduling with non-triangular sequence-dependent setup times. *International Journal of Production Research*, *52*(8), 2490–2503.

Copil, K., Worbelauer, M., Meyr, H., & Tempelmeier, H. (2017). Simultaneous lotsizing and scheduling problems: A classification and review of models. *OR spectrum*, *39*(1), 1–64.

Diaz-Madronero, M., Perez-Sanchez, M., Satorre-Aznar, J. R., Mula, J., & Lopez-Jimenez, P. A. (2018). Analysis of a wastewater treatment plant using fuzzy goal programming as a management tool: A case study. *Journal of Cleaner Production*, *180*, 20–33.

Drexl, A., & Haase, K. (1995). Proportional lotsizing and scheduling. *International Journal of Production Economics*, *40*(1), 73–87.

Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling—survey and extensions. *European Journal of operational research*, *99*(2), 221–235.

Echchakoui, S., & Barka, N. (2020). Industry 4.0 and its impact in plastics industry: A literature review. *Journal of Industrial Information Integration*, *20*, 100172.

Ferreira, D., Clark, A. R., Almada-Lobo, B., & Morabito, R. (2012). Single-stage formulations for synchronised two-stage lot sizing and scheduling in soft drink production. *International Journal of Production Economics*, *136*(2), 255–265.

Fleischmann, B. (1990). The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, *44*(3), 337–348.

Fleischmann, B., & Meyr, H. (1997). The general lotsizing and scheduling problem. *Operations-Research-Spektrum*, *19*(1), 11–21.

Fleischmann, B., Meyr, H., & Wagner, M. (2005). Advanced planning. In *Supply chain management and advanced planning* (pp. 81–106). Springer.

Guimarães, L., Klabjan, D., & Almada-Lobo, B. (2014). Modeling lotsizing and scheduling problems with sequence dependent setups. *European Journal of Operational Research*, *239*(3), 644–662.

Günther, H.-O. (2014). The block planning approach for continuous time-based dynamic lot sizing and scheduling. *Business Research*, *7*(1), 51–76.

Huang, J., Süer, G. A., & Urs, S. B. (2012). Genetic algorithm for rotary machine scheduling with dependent processing times. *Journal of Intelligent Manufacturing*, *23*(5), 1931–1948.

Ibarra-Rojas, O. J., Rios-Mercado, R. Z., Rios-Solis, Y. A., & Saucedo-Espinosa, M. A. (2011). A decomposition approach for the piece–mold–machine manufacturing problem. *International Journal of Production Economics*, *134*(1), 255–261.

Jordan, C., & Drexl, A. (1998). Discrete lotsizing and scheduling by batch sequencing. *Management Science*, *44*(5), 698–713.

Karimi, B., Ghomi, S. F., & Wilson, J. (2003). The capacitated lot sizing problem: A review of models and algorithms. *Omega*, *31*(5), 365–378.

Low, M., & Lee, K. (2008). Mould data management in plastic injection mould industries. *International Journal of Production Research*, *46*(22), 6269–6304.

Monma, C. L., & Potts, C. N. (1989). On the complexity of scheduling with batch setup times. *Operations research*, *37*(5), 798–804.

Mula, J., Dıaz-Madronero, M., Andres, B., Poler, R., & Sanchis, R. (2021). A capacitated lot-sizing model with sequence-dependent setups, parallel machines and bi-part injection moulding. *Applied Mathematical Modelling*, *100*, 805–820.

Pinedo, M. L. (2012). *Scheduling* (Vol. 29). Springer.

Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching: A review. *European journal of operational research*, *120*(2), 228–249.

Potts, C. N., & Van Wassenhove, L. N. (1992). Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *Journal of the Operational Research Society*, *43*(5), 395–406.

Rios-Solís, Y. Á., Ibarra-Rojas, O. J., Cabo, M., & Possani, E. (2020). A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production. *European Journal of Operational Research*, *284*(3), 861–873.

Suerie, C. (2006). Modeling of period overlapping setup times. *European journal of operational research*, *174*(2), 874–886.

Wassenhove, L. N. V., & Bodt, M. A. D. (1983). Capacitated lot sizing for injection moulding: A case study. *Journal of the Operational Research Society*, *34*(6), 489–501.

Wolsey, L. A. (2002). Solving multi-item lot-sizing problems with an mip solver using classification and reformulation. *Management Science*, *48*(12), 1587–1602.