Les Cahiers du GERAD ISSN: 0711–2440

An exact optimization approach for an integrated process configuration, lot-sizing, and scheduling problem

K. Pérez Martínez, Y. Adulyasak,
R. Jans, R. Morabito, E. A. Vitor Toso

G–2018–71

August 2018

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2018
– Bibliothèque et Archives Canada, 2018

Legal deposit – Bibliothèque et Archives nationales du Québec, 2018
– Library and Archives Canada, 2018

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

An exact optimization approach for an integrated process configuration, lot-sizing, and scheduling problem

Karim Pérez Martínez [a]

Yossiri Adulyasak [b]

Raf Jans [b]

Reinaldo Morabito [a]

Eli Angela Vitor Toso [c]

[a] *Department of Production Engineering, Federal University of São Carlos, São Carlos–SP, Brazil*

[b] *GERAD & Department of Logistics and Operations Management, HEC Montréal, Canada*

[c] *Department of Production Engineering, Federal University of São Carlos, Sorocaba–SP, Brazil*

karim@dep.ufscar.br
yossiri.adulyasak@hec.ca
raf.jans@hec.ca
morabito@ufscar.br
eli@ufscar.br

August 2018
Les Cahiers du GERAD
G–2018–71

Abstract: We study an integrated process configuration, lot-sizing, and scheduling problem, which appears in a real production environment in the packaging industry. Products are produced by alternative process configurations. The production quantities and capacity consumption depend on which process configurations are used, how long they are used for, and in which sequence. For the particular case studied here, configuration decisions are generated at the same time as lot-sizing and sequencing decisions, which involve sequence-dependent setup costs and times. Due to dependency of these decisions, the model is nonlinear. Even though a linearization technique can be applied, the problem is still difficult to solve by a mixed integer programming (MIP) solver due to its complexity. This paper aims to develop efficient solution methods to deal with this integrated production planning problem. An exact branch-and-check (B&Ch) algorithm based on a relaxed formulation and using logic-based Benders cuts is proposed to find optimal solutions. In addition, symmetry-breaking constraints are applied to strengthen the formulations. Results show that in general, the B&Ch outperforms the linearized models solved by an MIP solver. To efficiently solve large instances, an MIP-based heuristic is then proposed to find good quality solutions in shorter computing times. Although the problem studied here is based on the packaging industry, the logic of the B&Ch and the proposed heuristic can be adapted to other applications where lot-sizing must be determined simultaneously with process configuration decisions.

Keywords: Lot-sizing and scheduling, process configuration, logic-based Benders decomposition, branch-and-check, molded pulp industry.

Acknowledgments: We are grateful to the Brazilian molded pulp company that collaborated and provided data for this research. The first author acknowledges the financial support from FAPESP- São Paulo Research Foundation [grant numbers 2013/23700-8, 2015/24917-6].

1 Introduction

We consider an integrated optimization problem which combines process configuration, lot-sizing, and sequencing decisions inspired by a real-world production system in the packaging industry. In this integrated problem, products are produced by alternative process configurations or operation modes. A single product can be produced by alternative process configurations, and a single process configuration can produce several products simultaneously. The amount of products produced over the planning horizon depends on how long and how many times each process configuration is used. Therefore, lot-sizing and sequencing decisions, which may include sequence-dependent setups, are related to the processes instead of products, as in the classical lot-sizing problems (Johnson and Montgomery, 1974).

The set of possible process configurations by which products are produced can be either determined beforehand or generated at the same time as the lot-sizing and scheduling decisions. For some production environments, the number of configurations might be large, so complete enumeration approaches become intractable. On the other hand, considering only a subset of them may lead to sub-optimal solutions. Thus, integrated approaches capable of dealing with process configuration, lot-sizing, and scheduling are desirable for these integrated problems. A well known similar problem that may be considered as an integration of process configuration and lot-sizing is the cutting-stock problem, where the amount of each item depends on how many times the cutting patterns are used. A general review of this integrated optimization problem can be found in Melega et al. (2018).

Production planning problems which include process configuration decisions also appear in other industrial applications, as in refineries, the printing industry, and flexible manufacturing systems. For these problems, machines or production stages can be set according to alternative process configurations, which produce different products simultaneously. In lumber companies (Gaudreault et al., 2011), for instance, the amounts and mix of products depend on the configurations used in the drying and finishing stages for each period. In the foundry industry (Luche et al., 2009), production quantities of electrofused grains depend on the furnace configuration and the combination of shaking sieves used. In refineries (Göthe-Lundgren et al., 2002; Shi et al., 2014), production quantities depend on the choice of operation modes for each production unit. In the offset printing industry (Baumann et al., 2015), the mix and production quantities depend on the designs allocated to the printing-plate slots and the number of rotations of each plate.

This paper studies the integration of production planning decisions in the molded pulp packaging industry. In this real application, products are shaped by a molding machine with a specific combination of molds attached, which is called the *molding pattern*. The amounts produced depend on the molding patterns used and the speed of the molding machine (e.g., rotations per hour). Setup times and costs depend on how these molding patterns are sequenced over a finite planning horizon. As the number of all possible molding patterns can be very large and laborious to identify beforehand, integrated approaches are expected to be advantageous for this context. Alternative mathematical formulations to represent this optimization problem have been proposed in our previous works. Martínez et al. (2016) proposed a linear formulation for this problem, which assumes that some preset settings for the production line are known in advance to generate feasible molding patterns. Martínez et al. (2018) proposed a general nonlinear formulation, which integrates the main decisions and models the operational constraints of the problem. Results showed the capability of the general model to represent production environments of different sizes and its superiority over enumerating approaches.

The main contribution of this paper is on the development and performance of exact and heuristic solution approaches for the studied problem. These contributions are fourfold. First, we provide enhancements to improve the mathematical formulation presented in Martínez et al. (2018). Valid inequalities are proposed to deal with different types of symmetries of the original formulation. Second, we propose a reformulation, which is efficiently solved by an exact algorithm based on a branch-and-check (B&Ch) framework. Third, we perform extensive computational experiments to demonstrate the effectiveness of the symmetry-breaking constraints and the performance of the B&Ch algorithm compared to the linearized MIP model solved by CPLEX. Fourth, an MIP-based heuristic is proposed to achieve competitive solutions in shorter computing times for larger instances. Although the integrated problem studied here is a particular application in the molded pulp packaging industry, we believe the proposed approaches can be adapted to similar applications which combine lot-sizing with process configuration decisions, and for production planning optimization problems with nonlinear constraints.

This paper is organized as follows. The next section describes the planning problem. Section 3 presents the formulations for two different variants of the studied problem. Section 4 describes the symmetries in the models and proposes a set of valid inequalities to eliminate them. Section 5 describes the solution methods proposed to solve the problem. Section 6 presents the results of the computational experiments. Finally, Section 7 presents concluding remarks. We also provide an online supplement with additional information and detailed results of the approaches proposed here.

2 Problem description

In the molded pulp packaging industry, the process configuration, lot-sizing, and scheduling problem involves the decisions of generating the molding patterns to be used, defining how long each one should be used for, and the sequence in which they should be scheduled. Demands are dynamic and assumed to be deterministic, based on a combination of customer orders and forecasts of future requirements. The sales and marketing department provides these requirements (weekly demand) on a monthly basis to the production planning department. A production plan is then created at the beginning of each month to deal with the total demand of the entire month. Furthermore, some rescheduling decisions could be made every one or two weeks to improve the initial production plan. Deadlines may be renegotiated with customers, but this is considered as strategically undesirable. Therefore, backlogging is allowed but with a penalty.

We consider a single production line environment. A production line consists of one rotary molding machine followed by several conveyors that work simultaneously and synchronously. One mold produces only one type of product and a product is produced by only one type of mold. The molding machine has a set of F faces of width L, where molds of different types are attached. Each machine face is assigned to one specific conveyor, so that the output of the face is exclusively downloaded and transported by such conveyor. The assignment of faces to conveyors has been decided on beforehand and constitutes an input for the models. Thus, we consider that the machine faces are divided into disjoint subsets dedicated to each conveyor. These subsets are called here *faces for conveyor k*, denoted by F_k, where $F = \bigcup_{k\in K} F_k$ and $\bigcap_{k\in K} F_k = \emptyset$.

Decisions on the process configuration include how many molds of each type are attached to the molding machine, i.e., determining the molding patterns. However, not all possible combinations of molds are feasible, since some operational constraints must be satisfied. These constraints include the synchronization of the molding machine and conveyors. Conveyors are considered identical, when they have the same total width, but they may still be configured differently. The configuration of each conveyor is a decision to make which consists of determining the number of lanes and the width of each lane. Products must be transported on the conveyors in such a way that only one type of product is assigned to each lane. This is mandatory to properly carry out the subsequent stages of the process, i.e., sorting, pressing, and packing, which are set up according to the specific configuration of products on the conveyors. Therefore, the molding pattern attached to the machine must ensure that the products produced are transported properly on the corresponding conveyors (i.e., one type of product on each lane).

Figure 1 illustrates a production line and the synchronization of the molding pattern attached to the machine and the configuration of the conveyors. The molding machine has 6 faces in total. The output from faces 1, 3, and 5 are transported to conveyor 1, and the output of faces 2, 4, and 6 are transported to conveyor 2. Conveyor 1 has three separate lanes to feed the packaging stage. Conveyor 2 has two separate lanes. The sequence of products A, B, and C on the faces 1, 3, and 5 is arranged in such a way that product A is always transported in the first lane, product B in the second lane and product C in the third lane.

Besides the synchronization between the molding machine and conveyors, the following operational constraints must also be satisfied by the process configuration decisions. First, the number of molds of each type is limited, so each molding pattern must not exceed the total availability. In addition, the capacity of the

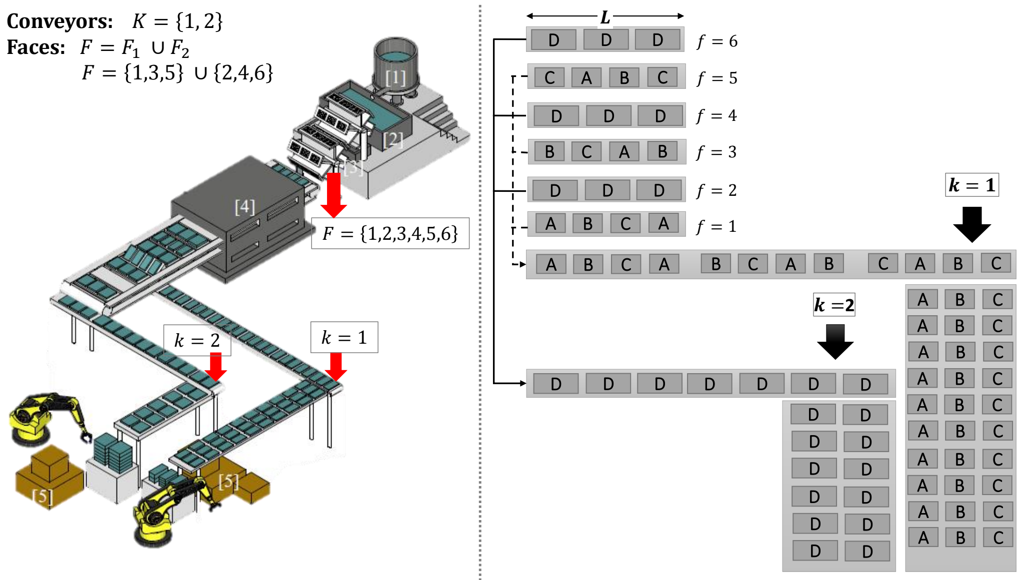

Figure 1: Production line in large scale molded pulp companies

machine faces must be totally used. This implies that all the faces must be used and fully filled with molds. Next, some products are incompatible and cannot be produced simultaneously. This is imposed because they may have incompatible physical and chemical characteristics, e.g., different colors, different weights, or they require a different pulp composition.

Lot-sizing decisions consist of deciding how long each molding pattern should be used for. The production quantity of each product in a given period is defined as the number of the corresponding molds, multiplied by how long the molding pattern is used for, and the speed of the molding machine (i.e., rotations per time unit).

Scheduling decisions specify how the molding patterns should be sequenced and involve sequence dependent setup times and costs. To properly represent these decisions, the setup operations are divided in three different types, namely, Setups I, II and III. Setup I involves the action of stopping and restarting the production line, and it is done every time a new molding pattern is attached to the machine. This setup is performed when at least one mold has to be replaced and it does not depend on the patterns and their sequence. Setup II is related to the steps to attach and detach molds to the machine faces. The total time for Setup II depends on the number of molds exchanged to set up a different pattern. Setup III is related to the changeovers in the configuration of the conveyors, i.e., the set of different lanes of different width, and it is sequence-dependent. This setup operation is usually the longest one since it can last from 10 to 48 hours, and it is only required when the replaced molds have different sizes.

3 Mathematical formulations

The formulations in this section are based on the general lot-sizing and scheduling problem (GLSP), originally proposed by Fleischmann and Meyr (1997). The planning horizon is divided into $|T|$ non-overlapping large time buckets (i.e., one week). This time horizon is further subdivided into $|S|$ non-overlapping subperiods of variable length. The production line can only be set up for a single molding pattern per subperiod and the models implicitly determine the length of the subperiod by the time that each pattern is used. The objective is to minimize the total production costs which include the setup, inventory holding, and backlogging costs.

We present two MIP formulations for two variants of the problem. The first model assumes a production line where all the molds have the same size, therefore only Setup I and II are required in the scheduling decisions. The second model represents a generalization of the first model where molds of different sizes can be used and thus Setup III is possibly required.

3.1 Model 1: MIP formulation for molds of identical size

In order to define the problem, we present the following sets, parameters, and decision variables.

Sets	
T	set of time periods (indexed by t);
S_t	set of subperiods belonging to period t (indexed by s);
S	set of subperiods, i.e., $\bigcup_{t \in T} S_t$;
N	set of products (indexed by i);
O	set of incompatible pairs of products (i, i');
K	set of conveyors (indexed by k);
F_k	set of faces for conveyor k (indexed by f);
F	set of faces of the molding machine, i.e., $\bigcup_{k \in K} F_k$;
P	set of lanes on each conveyor (indexed by p).
Parameters	
h_i^+	unit inventory holding cost for product i;
h_i^-	unit backlogging cost for product i;
sc^I	setup cost for stopping/restarting the production line (Setup I);
sc_i^{II}	setup cost for attaching/detaching one mold for product i (Setup II);
d_{it}	demand of product i in period t;
Q_t	total time capacity in period t;
M_i	number of molds for product i available;
l	width of molds;
L	total width of every face of the molding machine;
R	speed of the molding machine (i.e., rotations per hour);
H_k	proportion parameter between the size of the machine faces and conveyor k; this parameter takes the value $\left\lfloor \frac{L}{\text{width of conveyor } k} \right\rfloor$;
st^I	setup time for stopping/restarting the production line (Setup I);
st_i^{II}	setup time for attaching/detaching one mold for product i (Setup II).
Decision variables	
Process configuration decisions	
y_{ipks}	equals 1 if product i is assigned to lane p of conveyor k in subperiod s; 0, otherwise;
x_{ifs}	number of molds for product i attached to face f in subperiod s.
Lot-sizing decisions	
w_s	production time in subperiod s;
I_{it}^+	inventory quantity of product i at the end of period t;
I_{it}^-	backlogged quantity of product i at the end of period t.
Scheduling decisions	
v_s	equals 1 if a new molding pattern is set up in subperiod s; 0, otherwise;
u_{iks}	number of molds for product i attached or detached to the faces for conveyor k, in subperiod s.
Auxiliary variables	
Y_{is}	equals 1 if product i is produced in subperiod s; 0, otherwise;
α_{iks}	equals 1 if product i is assigned to all lanes of conveyor k in subperiod s; 0, otherwise;
β_{ks}	integer variable to impose a logical relationship between the configuration of conveyor k in subperiod s, and the molds attached to the faces for this conveyor in the same subperiod.

Objective function Objective function (1) minimizes the total production costs, which in this case involves costs for Setups I and II, inventory holding, and backlogging costs.

$$\text{Min } \sum_{s \in S} sc^I v_s + \sum_{s \in S} \sum_{k \in K} \sum_{i \in N} sc_i^{II} u_{iks} + \sum_{t \in T} \sum_{i \in N} (h_i^+ I_{it}^+ + h_i^- I_{it}^-) \tag{1}$$

Process configuration decisions The set of constraints (2)–(12) are all related to the process configuration decisions. Constraints (2)–(5) define the configuration of the conveyors. Specifically, constraints (2) enforce that only one product is assigned to each lane of each conveyor in each subperiod. Constraints (3) and (4) ensure that product i is produced in a given subperiod if, and only if, this product is assigned to at least one lane in any of the conveyors. Constraints (5) guarantee that incompatible products are not produced simultaneously in any of the subperiods.

$$\sum_{i \in N} y_{ipks} = 1 \qquad \forall s \in S;\ k \in K;\ p \in P \tag{2}$$

$$Y_{is} \leq \sum_{k\in K}\sum_{p\in P} y_{ipks} \qquad \forall s \in S;\ i \in N \tag{3}$$

$$Y_{is} \geq y_{ipks} \qquad \forall s \in S;\ i \in N;\ k \in K;\ p \in P \tag{4}$$

$$Y_{is} + Y_{i's} \leq 1 \qquad \forall s \in S;\ (i,i') \in O \tag{5}$$

Constraints (6)–(7) define the molding patterns attached to the machine. Constraints (6) ensure that as many molds as possible are attached to each machine face. Therefore, the sum of the width of these molds must not exceed the total width of the machine face and there must not remain enough space to attach any other mold. Note that, as this model assumes that all the molds have the same width, we can state that the number of molds attached to each face is exactly $\lfloor \frac{L}{l} \rfloor$. Constraints (7) guarantee that for each subperiod, the required number of molds to be attached to the complete machine does not exceed the number of molds available of each type.

$$\sum_{i\in N} x_{ifs} = \left\lfloor \frac{L}{l} \right\rfloor \qquad \forall s \in S;\ f \in F \tag{6}$$

$$\sum_{f\in F} x_{ifs} \leq M_i \qquad \forall s \in S;\ i \in N \tag{7}$$

Constraints (8)–(12) link the configurations of conveyors and the molding machine to synchronize the process. Constraints (8) ensure that the molds attached to the machine correspond to the products assigned to the conveyors. Therefore, if molds for product i are attached to any of the faces for conveyor k (i.e., any $f \in F_k$), then product i must be assigned to at least one lane of conveyor k. Constraints (9) are valid inequalities and ensure that if product i is transported on conveyor k, then the number of molds for this product attached to each face for the same conveyor (i.e., $\forall\, f \in F_k$) is at least H_k multiplied by the number of lanes filled by product i. Our results show that these valid inequalities significantly improve the lower bounds of the formulations.

$$\sum_{f\in F_k} x_{ifs} \leq M_i \sum_{p\in P} y_{ipks} \qquad \forall s \in S;\ k \in K;\ i \in N \tag{8}$$

$$x_{ifs} \geq H_k \sum_{p\in P} y_{ipks} \qquad \forall s \in S;\ k \in K;\ f \in F_k;\ i \in N \tag{9}$$

Constraints (10) determine whether conveyors transport different products at the same time or not. They impose that $\alpha_{iks} = 1$ if and only if product i is assigned to all the lanes in conveyor k in subperiod s (i.e., $\sum_{p\in P} y_{ipks} = |P|$).

$$|P| - \sum_{p\in P} y_{ipks} \leq |P|(1-\alpha_{iks}) \qquad \forall s \in S;\ i \in N;\ k \in K \tag{10}$$

Constraints (11) and (12) ensure that the products will be arranged in the same order on the conveyor, as demanded by the operational constraints for this problem. To be able to satisfy this synchronization, we have to impose a logical relationship between the configuration of a conveyor and the molds attached to the faces for this conveyor. This is imposed particularly for the cases in which a conveyor is configured to transport different products simultaneously. The nonlinear constraints (11) and (12) guarantee that there is an integer β_{ks} so that for any product the number of molds attached to the faces for conveyor k equals the number of lanes assigned to this product, multiplied by β_{ks}. When the conveyor k transports only one type of product (i.e., $\sum_{i\in N} \alpha_{iks} = 1$), this condition can be relaxed, so the number of molds attached are only defined by (6)–(9). Note that B_k in constraints (11) and (12) is an upper bound of $\sum_{f\in F_k} x_{ifs}$, which can be defined as $B_k = |F_k| \lfloor \frac{L}{l} \rfloor$.

$$\sum_{f\in F_k} x_{ifs} \leq \beta_{ks} \sum_{p\in P} y_{ipks} + B_k \sum_{i'\in N} \alpha_{i'ks} \qquad \forall s \in S;\ i \in N;\ k \in K \tag{11}$$

$$\sum_{f\in F_k} x_{ifs} \geq \beta_{ks} \sum_{p\in P} y_{ipks} - B_k \sum_{i'\in N} \alpha_{i'ks} \qquad \forall s \in S;\ i \in N;\ k \in K \tag{12}$$

Lot-sizing decisions Constraints (13) ensure that the total time capacity of each period, which is consumed by setup times and production times of the patterns used in that period, is not exceeded. Constraints (14) are the demand balance constraints, which link the process configuration and lot-sizing decisions in order to determine the production quantities in each period. Note that for this problem, the quantity produced of each product is a function of the molding patterns defined, the speed of the molding machine R, and the time each pattern is used (i.e., w_s), which results in another set of nonlinear constraints. These constraints, as well as constraints (11)–(12), can be linearized as presented in Appendix A of the online supplement.

$$\sum_{s \in S_t} w_s + \sum_{s \in S_t} st^I v_s + \sum_{s \in S_t} \sum_{k \in K} \sum_{i \in N} st_i^{II} u_{iks} \leq Q_t \qquad \forall t \in T \tag{13}$$

$$I^+_{i(t-1)} + \sum_{s \in S_t} Y_{is} w_s R \sum_{f \in F} x_{ifs} + I^-_{it} = d_{it} + I^-_{i(t-1)} + I^+_{it} \quad \forall t \in T;\, i \in N \tag{14}$$

Scheduling decisions Constraints (15) and (16) model Setup II by determining how many molds of each type are attached and detached to the set of faces for each conveyor. Note that, as $u_{iks} \geq 0$, constraints (15) are active when molds of type i are attached to the faces $f \in F_k$ in subperiod s, while constraints (16) are active when molds of type i need to be detached in this subperiod. Constraints (17) ensure that the Setup I is done when at least one mold of any type needs to be attached or detached, i.e., every time a new molding pattern is set up.

$$u_{iks} \geq \sum_{f \in F_k} x_{ifs} - \sum_{f \in F_k} x_{if(s-1)} \qquad \forall s \in S : s \geq 1;\, i \in N;\, k \in K \tag{15}$$

$$u_{iks} \geq \sum_{f \in F_k} x_{if(s-1)} - \sum_{f \in F_k} x_{ifs} \qquad \forall s \in S : s \geq 1;\, i \in N;\, k \in K \tag{16}$$

$$u_{iks} \leq M_i v_s \qquad \forall s \in S;\, i \in N;\, k \in K \tag{17}$$

Finally constraints (18) define the domain of the variables in this formulation.

$$\begin{aligned} &Y_{is},\, v_s,\, y_{ipks},\, \alpha_{iks} \in \{0,1\}; \quad x_{ifs},\, \beta_{ks} \in \mathbf{Z}^+; \quad u_{iks},\, w_s,\, I^+_{it},\, I^-_{it} \in \mathbf{R}^+ \\ &\qquad\qquad \forall t \in T;\, s \in S;\, i \in N;\, k \in K;\, f \in F;\, p \in P \end{aligned} \tag{18}$$

3.2 Model 2: MIP formulation for molds of different size

This formulation is a generalization of the previous one because it assumes that the molds attached to the machine may be of different sizes. In this case, the conveyors might require adjustments in their lanes, which involve an additional setup operation named Setup III. To model the decisions about the configuration of conveyors, in addition to defining which product is assigned to each lane, it is also necessary to know which *arrangement* must be set up for each conveyor. An arrangement defines how many lanes are set up on the conveyor and the width of each one. The set of possible arrangements E as well as the data related to them are determined in advance, i.e., the number of lanes ($|P_e|$) and the width of each lane (g_{pe}). As we considered that conveyors are identical (i.e., they have the same total width), it implies that all conveyors in set K can be configured to any possible arrangement $e \in E$. Thus, the configuration of a conveyor is defined by deciding which arrangement is set up and which product is assigned to each lane. Model 1 can be considered as a special case of Model 2, when the number of possible arrangements equals one ($|E| = 1$) and $l_i = l$, $\forall i \in N$. In addition to the data and variables defined for Model 1, the following data and variables are also required for Model 2.

Additional sets	
E	set of arrangements for conveyors (indexed by e);
P_e	set of lanes of a conveyor according to arrangement e; $\bigcup_{e\in E} P_e = P$ and $\bigcap_{e\in E} P_e \neq \emptyset$.
Additional parameters	
l_i	width of molds for product i;
g_{pe}	width of lane p if a conveyor is set up according to arrangement e;
a_{pe}	equals 1 if position $p \in P_e$; 0 otherwise;
$sc^{III}_{ee'}$	setup cost for a changeover from arrangement e to e' (Setup III);
$st^{III}_{ee'}$	setup time for a changeover from arrangement e to e' (Setup III).
Additional decision variables	
z_{kes}	equals 1, if conveyor k is set up according to arrangement e in subperiod s; 0, otherwise;
$b_{ee'ks}$	equals 1, if there is a changeover from arrangement e to e' in conveyor k, in subperiod s; 0, otherwise.

Objective function As in Model 1, the objective (19) is to minimize the total costs, which in this case also include the cost for Setup III.

$$\text{Min} \sum_{s\in S} sc^I v_s + \sum_{s\in S}\sum_{k\in K}\sum_{i\in N} sc^{II}_i u_{iks} + \sum_{s\in S}\sum_{k\in K}\sum_{e,\,e'\in E} sc^{III}_{ee'} b_{ee'ks} + \sum_{t\in T}\sum_{i\in N} (h^+_i I^+_{it} + h^-_i I^-_{it}) \tag{19}$$

Process configuration decisions Constraints (20)–(22), in addition to (3)–(5), define the setup state of the conveyors. Constraints (20) ensure that each conveyor is set up for only one arrangement in each subperiod. Constraints (21) ensure that a single type of product is assigned to each lane configured on each conveyor. The number of lanes in a conveyor depends on the arrangement set up, which is a decision to be made. Constraints (22) guarantee that, if product i is assigned to lane p, then the lane and the mold that produces this product have the same width. These constraints are valid because of (21) and the assumption that each product is obtained by only one type of mold.

$$\sum_{e\in E} z_{kes} = 1 \qquad \forall s \in S;\ k \in K \tag{20}$$

$$\sum_{i\in N} y_{ipks} = \sum_{e\in E} a_{pe} z_{kes} \qquad \forall s \in S;\ k \in K;\ p \in P \tag{21}$$

$$\sum_{i\in N} l_i y_{ipks} = \sum_{e\in E} g_{pe} z_{kes} \qquad \forall s \in S;\ k \in K;\ p \in P \tag{22}$$

Constraints related to the configuration of the molding machine, i.e., constraints (7), and the synchronization between the machine and conveyors, i.e., (8)–(9) and (11)–(12), remain the same as in Model 1. Constraints (23)–(24) are included in this model to guarantee that each machine face is completely filled with molds. As the size of molds might be different in this case, the number of molds to be attached to each face is not known in advance as in Model 1 (i.e., $\lfloor \frac{L}{l} \rfloor$). Therefore, constraints (23) impose that the sum of the width of the molds attached to each face does not exceed the total width of the face, whereas constraints (24) ensure that the unused space in each face is smaller than the size of the smallest lane of the corresponding conveyor. As the lanes of conveyors and molds attached to their faces are of the same width (imposed by constraints (22)), this also implies that there should not be enough space left to attach any mold of any type set up in a specific subperiod. The value of ξ is set up as 10e-4 for this implementation.

Similar to constraints (10) for Model 1, constraints (25) ensure $\alpha_{iks} = 1$ if all the lanes of conveyor k are assigned to the same product i.

$$\sum_{i\in N} l_i x_{ifs} \leq L \qquad \forall s \in S;\ f \in F \tag{23}$$

$$L - \sum_{i\in N} l_i x_{ifs} \leq \sum_{e\in E} \min_{p\in P_e} \{g_{pe}\} z_{kes} - \xi \qquad \forall s \in S;\ k \in K;\ f \in F_k \tag{24}$$

$$\sum_{e\in E} |P_e| z_{kes} - \sum_{p\in P} y_{ipks} \leq |P|(1-\alpha_{iks}) \qquad s \in S;\ i \in N;\ k \in K \tag{25}$$

Lot-sizing decisions These sets of constraints include the inventory balance equalities (14) as in Model 1 and constraints (26). These are the capacity consumption constraints, which include the time required for Setup III, where $st^{III}_{ee} = 0$, $\forall e \in E$.

$$\sum_{s\in S_t} st^{I} v_s + \sum_{s\in S_t}\sum_{k\in K}\sum_{i\in N} st^{II}_i u_{iks} + \sum_{s\in S_t}\sum_{k\in K}\sum_{e,e'\in E} st^{III}_{ee'} b_{ee'ks} + \sum_{s\in S_t} w_s \le Q_t \qquad \forall t\in T \tag{26}$$

Scheduling decisions In addition to constraints (15)–(17) in Model 1 for Setups I and II, constraints (27) and (28) for Setup III are added. These constraints model the changeovers of the arrangements on each conveyor, which are sequence-dependent. Constraints (29) properly link Setups I and III, so that Setup I must be performed if Setup III occurs in a given subperiod. Constraints (29) can be omitted because they are redundant with constraints (15)–(17) and (27)–(28). However, preliminary experiments for this formulation showed improved results when these valid inequalities are added.

$$\sum_{e\in E} b_{ee'ks} = z_{ke's} \qquad \forall s\in S;\, k\in K;\, e'\in E \tag{27}$$

$$\sum_{e'\in E} b_{ee'ks} = z_{ke(s-1)} \qquad \forall s\in S : s\ge 2;\, k\in K;\, e\in E \tag{28}$$

$$1 - \sum_{e\in E} b_{eeks} \le v_s \qquad \forall s\in S;\, k\in K \tag{29}$$

Finally the domains of the new variables are defined in (30) in addition to the ones defined in (18).

$$z_{kes},\, b_{ee'ks} \in \{0,\,1\} \qquad \forall s\in S;\, k\in K;\, e,e'\in E \tag{30}$$

4 Symmetry-breaking constraints

We propose valid constraints to deal with three different types of symmetry. The first one is related to the setup state of the whole production line in each subperiod, the second one is related to the assignment of products to conveyors' lanes, and the third one is related to the set up of arrangements to conveyors. The two first symmetries may appear in both Models 1 and 2, whereas the last one only appears in Model 2.

The first type of symmetry is the same issue as in the formulation for the original GLSP in Fleischmann and Meyr (1997). The idea is to place subperiods with no setup operations at the end of each period; so constraints (31) can be included, where s^f_t represents the first subperiod in period t. The domain of these constraints is $\forall s \ge s^f_t + 2$ since the models 1 and 2 allow to keep the setup state from one period to the next one, as the classical GLSP. Note that for the cases when the setup state is carried over from a specific period to the next one (e.g, from period $t-1$ to period t), there is not setup operation performed in the first subperiod of t, i.e., $v_{s^f_t} = 0$. Therefore, constraints (31) would not be valid inequalities if defined $\forall s \ge s^f_t + 1$ because this consequently forces all the Setup I variables to be null (i.e., $v_s = 0\ \forall s\in S_t$), and therefore these constraints may eliminate optimal solutions.

$$v_s \le v_{s-1} \qquad \forall t\in T;\, s\in S_t : s \ge s^f_t + 2 \tag{31}$$

The second type of symmetry appears since multiple identical assignments of products to the lanes of the conveyor exist for a given molding pattern. The assignment of products to the lanes of the conveyor can hence be permuted without affecting the feasibility or the total cost. This is illustrated in Figure 2. One can observe that, for cases 1, 2, and 3, the number of molds of each type attached to machine faces for conveyor $k = 1$ is the same, whereas the values of y_{ipks} are different.

Constraints (32) ensure that products are assigned to the conveyors' lanes in ascending order, so that the product with the smallest index is placed in the first lane and the one with the largest index in the last lane. These constraints can be added to both Models 1 and 2. However, for Model 2, where molds and products

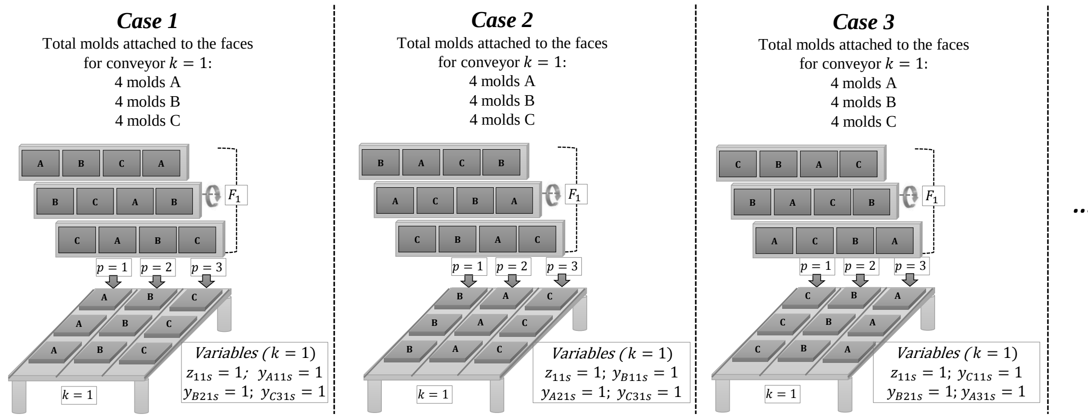

Figure 2: Symmetric solutions avoided by constraints (32)

are of different size, the input data must be ordered so that products are enumerated from 1 to $|N|$, and lanes from 1 to $|P|$, according to their width, from the smallest to the largest. Otherwise constraints (32) are not valid inequalities for Model 2, as they may cut optimal solutions.

$$\sum_{\substack{i' \in N: \\ 1 \le i' \le i}} y_{i'(p-1)ks} \ge y_{ipks} \qquad \forall s \in S;\ i \in N;\ k \in K;\ p \in P : p \ge 2 \tag{32}$$

Symmetries related to arrangements and conveyors only appear in Model 2. For a given molding pattern, the assignment of arrangements (using the z_{kes} variables) to the various conveyors can be permuted. This is shown in Figure 3. Constraints (33) ensure that arrangements are assigned in ascending order to conveyors, i.e., the arrangement with the smallest index is assigned to the first conveyor and the arrangement with the largest index is assigned to the last conveyor. These constraints are valid since we assume that all conveyors may be configured to any of the arrangements in set E.

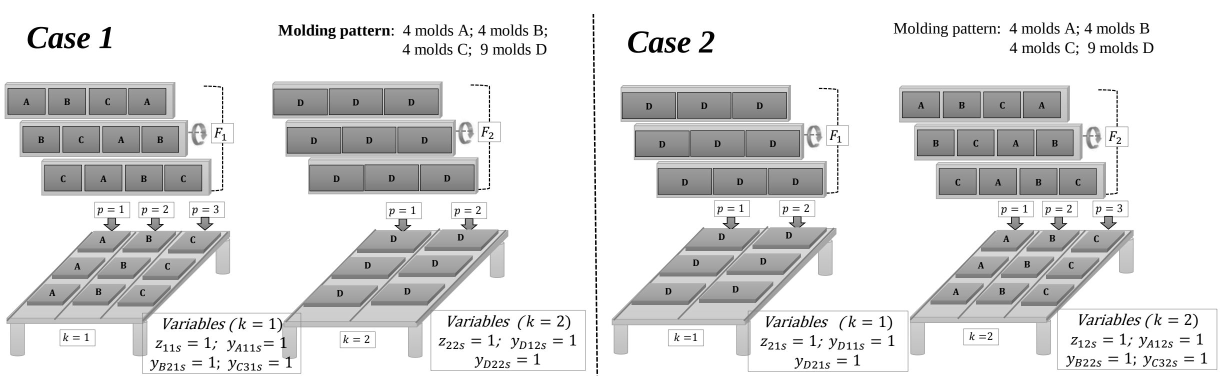

Figure 3: Symmetric solutions avoided by constraints (33)

$$\sum_{\substack{e' \in E: \\ 1 \le e' \le e}} z_{(k-1)e's} \ge z_{kes} \qquad s \in S;\ e \in E;\ k \in K;\ k \ge 2 \tag{33}$$

5 Solution approaches

5.1 Linearized MIP models solved by CPLEX

We employ a linearization technique for the nonlinear constraints (11)–(12) and (14), and solve the linearized models using a standard MIP solver. Since these constraints involve the product of variables where at least one of them is integer, we can use a binary representation to substitute these integer variables and define additional inequalities to linearize the nonlinear constraints. We then solve the linearized formulations using CPLEX. An improved version of the linearization approach proposed in Martínez et al. (2018) is described in Appendix A of the online supplement.

5.2 Reformulation solved by a branch-and-check algorithm

As an exact solution approach to solve the problem, we propose a branch-and-check algorithm (B&Ch) based on logic-based Benders decomposition (LBBD), which are formally introduced by Hooker (2000), Thorsteinsson (2001) and Hooker and Ottosson (2003). In this framework, a relaxed model is solved by a branch-and-bound and cuts based on the LBBD are generated and added during the branch-and-bound process. Some applications of logic-based Benders implementations on production management problems have been studied in Hooker (2007); Fazel-Zarandi and Beck (2009); Cao et al. (2010), and most recently in Côté et al. (2014), and Delorme et al. (2017). In our study, we use LBBD in a branch-and-bound for a nonlinear formulation, specifically to deal with the demand balance constraints, where quantities produced are defined as a nonlinear function of other decision variables of the problem. We then proposed a *relaxed formulation* solved by the B&Ch algorithm which comprises two main elements: a *cut generation routine* and a *procedure to find feasible solutions* for the original nonlinear formulation.

This algorithm uses the branch-and-bound callback of CPLEX for MIP problems and this is implemented as follows. Firstly, the linear program of the *relaxed formulation* is solved. If the solution of this relaxation is not an integer feasible solution, then branching starts. If an integer feasible solution is found, the *cut generation routine* is performed to detect whether this solution represents a feasible integer solution for the original nonlinear problem. If the solution is proven to be infeasible for the original problem, some valid inequalities are added to the current formulation, and then the branch-and-bound process continues. After the cut generation is performed, a *procedure to find a feasible solution for the original problem* is also called and, if the solution obtained is better than the incumbent one, this one is provided as an upper bound to the branch-and-bound tree.

5.2.1 Relaxed formulation

This relaxed formulation consist on relaxing the demand balance constraints (14) by representing the production quantities as single variables limited by valid bounds. So, the quantity of product i produced in subperiod s is defined as the product of the production time w_s, the machine speed R, and an approximation of the number of molds for this product in the pattern defined for such subperiod. For this relaxed formulation, the demand balance constraints become linear, and consequently no linearization is needed for these constraints. Optimal solutions for this reformulation are valid lower bounds for the original problem, which can be strengthened by cutting planes through the branch-and-bound tree.

Denote $q_{is} \in \mathbf{R}^+$ as the quantity produced of product i in subperiod s. The relaxed formulation can be obtained by replacing constraints (14) in Model 1 or 2 by constraints (34)–(37).

$$I^+_{i(t-1)} + \sum_{s\in S_t} q_{is} + I^-_{it} = d_{it} + I^-_{i(t-1)} + I^+_{it} \qquad \forall t \in T;\, i \in N \tag{34}$$

$$q_{is} \le w_s R M_i \qquad \forall s \in S;\, i \in N \tag{35}$$

$$q_{is} \le Q_t R \sum_{f\in F} x_{ifs} \qquad \forall t \in T;\, s \in S_t;\, i \in N \tag{36}$$

$$q_{is} \ge \min_{k\in K}\{|F_k| H_k\} w_s R - BM^{II}_{is}(1 - Y_{is}) \qquad \forall s \in S;\, i \in N \tag{37}$$

where BM^{II}_{is} is a large enough number that can be approximated as an upper bound for q_{is}. We set $BM^{II}_{is} = Q_t R M_i, \forall t \in T;\, s \in S_t;\, i \in N$ in our experiments. Note that the original demand balance constraints are equivalent to equations (34) if $q_{is} = Y_{is} w_s R \sum_{f\in F} x_{ifs}$, $\forall s \in S;\, i \in N$. However, for the relaxed formulation q_{is} is an approximation of the true amount, where valid upper bounds are defined by constraints (35) and (36), and a valid lower bound by constraints (37).

The bounds imposed for q_{is} are always valid. If product i is produced in s, then the quantity produced is at most equal to $w_s RM_i$, which means that the pattern defined in such subperiod contains all the molds available for this product. This upper bound is imposed by constraints (35) and it is valid because constraints (7) ensure that the maximum number of molds used is at most M_i. If product i is not produced in s, then there is no molds for this product on the machine (i.e., $\sum_{f\in F} x_{ifs} = 0$), hence q_{is} becomes zero via constraints (36). If product i is produced in s, then (36) still provides a valid upper bound, which is the amount produced by using the defined molding pattern during the whole corresponding time period. Likewise, if product i is produced (i.e., $Y_{is} = 1$), the quantity of this product in s is greater than or equal to the production time in this subperiod w_s, times the speed of the molding machine R, times the minimum number of molds for this product attached to the molding machine, as imposed by (37).

If product i is produced in a given subperiod, then the number of molds for this product in the molding pattern is at least one. However, constraints (3) and (9) allows us to define the minimum number of molds in any molding pattern as $\min_{k\in K}\{|F_k|H_k\}$. Note that $Y_{is} = 1$ implies that product i is assigned to at least one lane of one of the conveyors, as constraints (3) ensure. Consequently, the molds for product i must be attached H_k times to each face for the conveyor to which this product was assigned (see constraints (9)).

5.2.2 Cut generation routine

This routine is performed in the B&Ch every time a feasible integer solution of the relaxed formulation is found. As the relaxed formulation only defines valid upper and lower bounds for q_{is}, solutions provided may be not feasible for the original model since q_{is} may not be equal to $Y_{is} w_s R \sum_{f\in F} x_{ifs}$. Therefore, the values of the variables q_{is}, $\forall s \in S$; $i \in N$ are verified to see whether they are valid for the original problem or not. If they are not valid, cuts are added at the current node for all violations regarding the original value of variables q_{is}, and then the branching process continues. These valid cuts are shown in constraints (38) and (39), where $\hat{Y}_{is}$, $\hat{x}_{ifs}$, $\hat{z}_{kes}$, and $\hat{y}_{ipks}$ are the current values of variables Y_{is}, x_{ifs}, z_{kes}, and y_{ipks}, $\forall s \in S$; $i \in N$; $k \in K$; $p \in P$; $e \in E$, respectively, and $\mathbf{S}$ is the set of pairs (i, s) corresponding to the variables $\hat{q}_{is}$ whose value are not valid for the original program.

$$
\begin{aligned}
q_{is} \geq & \hat{Y}_{is} w_s R \sum_{f\in F} \hat{x}_{ifs} - BM^{II}_{is} \sum_{k\in K} \sum_{\substack{e\in E:\\ \hat{z}_{kes}=1}} (\hat{z}_{kes} - z_{kes}) - BM^{II}_{is} \sum_{k\in K} \sum_{\substack{e\in E:\\ \hat{z}_{kes}=0}} z_{kes} \\
& - BM^{II}_{is} \sum_{p\in P} \sum_{\substack{k\in K:\\ \hat{y}_{ipks}=1}} (\hat{y}_{ipks} - y_{ipks}) - BM^{II}_{is} \sum_{p\in P} \sum_{\substack{k\in K:\\ \hat{y}_{ipks}=0}} y_{ipks} \quad \forall (i,s) \in \mathbf{S}
\end{aligned} \tag{38}
$$

$$
\begin{aligned}
q_{is} \leq & \hat{Y}_{is} w_s R \sum_{f\in F} \hat{x}_{ifs} + BM^{II}_{is} \sum_{k\in K} \sum_{\substack{e\in E:\\ \hat{z}_{kes}=1}} (\hat{z}_{kes} - z_{kes}) + BM^{II}_{is} \sum_{k\in K} \sum_{\substack{e\in E:\\ \hat{z}_{kes}=0}} z_{kes} \\
& + BM^{II}_{is} \sum_{p\in P} \sum_{\substack{k\in K:\\ \hat{y}_{ipks}=1}} (\hat{y}_{ipks} - y_{ipks}) + BM^{II}_{is} \sum_{p\in P} \sum_{\substack{k\in K:\\ \hat{y}_{ipks}=0}} y_{ipks} \quad \forall (i,s) \in \mathbf{S}
\end{aligned} \tag{39}
$$

Inequalities (38)–(39) are based on the logic-based Benders decomposition approach. These constraints ensure that if the setup state in a subperiod s is the same as the current relaxed solution (the values of variables z_{kes} and y_{ipks} are the same as $\hat{z}_{kes}$ and $\hat{y}_{ipks}$), we have $q_{is} = \hat{Y}_{is} w_s R \sum_{f\in F} \hat{x}_{ifs}$, $i \in N$, and consequently the solution is also feasible for the original problem. Otherwise, q_{is} is relaxed and only bounded by constraints (35)–(37) of the relaxed formulation. For Model 1, the cuts to be added are similar to constraints (38) and (39), but they do not include the terms related to variable z_{kes}, i.e., the terms $BM^{II}_{is} \sum_{k\in K} \sum_{\substack{e\in E:\\ \hat{z}_{kes}=1}} (\hat{z}_{kes} - z_{kes})$ and $BM^{II}_{is} \sum_{k\in K} \sum_{\substack{e\in E:\\ \hat{z}_{kes}=0}} z_{kes}$.

5.2.3 Procedure to find feasible solutions for the original problem

The main idea of this procedure is to generate feasible solutions for the original problem using information provided by the integer solutions of the relaxed formulation. In the relaxed problem, the total inventory holding and backlogging costs are not accurate because the demand balance constraints are relaxed. Thus, this procedure consists of optimizing the lot-sizing decisions by considering as input parameters the information provided by the relaxed solution about process configurations and sequencing decisions. Given a relaxed feasible solution, the linear model (40)–(43) is solved to reoptimize and make the values of variables w_s, q_{is}, I^+_{it}, and I^-_{it} feasible for the nonlinear formulation.

Variables

$\tilde{w}_s$	production time in subperiod s;
$\tilde{I}^+_{it}$	inventory quantity of product i at the end of period t;
$\tilde{I}^-_{it}$	backlogged quantity of product i at the end of period t

$$\text{Min} \sum_{t \in T} \sum_{i \in N} \left(h^+_i \tilde{I}^+_{it} + h^-_i \tilde{I}^-_{it} \right) \tag{40}$$

$$\tilde{I}^+_{i(t-1)} + \sum_{s \in S_t} \hat{Y}_{is} \tilde{w}_s R \sum_{f \in F} \hat{x}_{ifs} + \tilde{I}^-_{it} = d_{it} + \tilde{I}^-_{i(t-1)} + \tilde{I}^+_{it} \quad \forall t \in T;\, i \in N \tag{41}$$

$$\sum_{s \in S_t} \tilde{w}_s \leq Q_t - \hat{ST}_t \qquad \forall t \in T \tag{42}$$

$$\tilde{w}_s,\, \tilde{I}^+_{it},\, \tilde{I}^+_{it} \in \mathbf{R} \qquad \forall t \in T;\, s \in S_t;\, i \in N \tag{43}$$

where $\hat{Y}_{is}$ and $\hat{x}_{ifs}$ are the values of variables Y_{is} and x_{ifs} in the current integer relaxed solution, and $\hat{ST}_t$ is the total setup time in period t for the current relaxed solution, determined as:

$$\text{Model 1: } \hat{ST}_t = \sum_{s \in S_t} st^I \hat{v}_s + \sum_{s \in S_t} \sum_{i \in N} \sum_{k \in K} st^{II}_i \hat{u}_{iks}$$

$$\text{Model 2: } \hat{ST}_t = \sum_{s \in S_t} st^I \hat{v}_s + \sum_{s \in S_t} \sum_{i \in N} \sum_{k \in K} st^{II}_i \hat{u}_{iks} + \sum_{s \in S_t} \sum_{k \in K} \sum_{e,e' \in E} st^{III}_{ee'} \hat{z}_{kes}$$

Given an optimal solution of the sub-problem (40)–(43), a feasible solution for the original problem is constructed by taking the integer solution of the relaxed problem and substituting the values of variables $\hat{w}_s$, $\hat{I}^+_{it}$, and $\hat{I}^-_{it}$ by the values for the optimal solution of the corresponding sub-problem (40)–(43). Backlogging and inventory holding costs must be adjusted.

5.3 MIP-based heuristic

Computational experiments using the solution approaches described above show the complexity of the problem and indicate how challenging it is to find optimal solutions for real size instances in affordable computing times. An MIP based heuristic, which consists of three phases as presented in Algorithm 1, is developed to quickly determine a good solution in shorter computing times compared to the exact methods proposed for this problem.

This heuristic consists of the following three phases: (i) defining a set of feasible process configurations; (ii) finding a feasible solution using the choices defined in phase (i); and (iii) applying an improvement procedure to the solution obtained in phase (ii). A description of each phase is presented as follows and complemented by further details in Appendix B of the online supplement.

Algorithm 1 3-phase MIP-based heuristic

1: **Phase 1:** Create an empty set of possible patterns $\xi \leftarrow \emptyset$
2: Perform the approaches described in Section 5.3.1 to generate feasible molding patterns for the original problem.
3: Append to the list ξ the molding patterns generated in this stage and save the information related to each $\epsilon \in \xi$ which will be required for next phases.
4: **Phase 2:** Take the list ξ and the parameters associated to this set as input data for the an MIP model for lot sizing and scheduling, and solve it using CPLEX (this MIP model is prsented in Appendix B (B.2)–(B.12) of the online supplement).
5: **Phase 3:** Transform the solution obtained in the previous step into a feasible solution for the original formulation.
6: Perform the fix-and-optimize heuristic presented in Section 5.3.3
7: **return** A feasible solution for the original problem

5.3.1 Phase I

In this phase a set of feasible molding patterns are generated for each problem instance. The aim is not only to generate patterns that will be used for the next phase of the heuristic, but to try to generate the ones which are likely to be used in good quality solutions for the original problem. In order to achieve this, the algorithms used in this phase take into account information about the lot-sizing and scheduling decisions, besides the technological requirements which ensure that the generated molding patterns are feasible.

We perform two different relax-and-fix (R&F) heuristics based on the original MIP formulation, which uses CPLEX and the B&Ch to solve the subproblem partitions. The R&F approach has been widely used to solve many production problems, particularly lot-sizing and scheduling problems Wolsey (1998). This is a construction heuristic which find an initial solution by solving several small integer problems. This is done by dividing the integer variables into 3 partitions: a subset of fixed variables, a subset of integer variables to be optimized, and a subset of variables whose integrality condition is dropped. At each iteration, these partitions are defined according to a particular strategy and the resultant subproblem is solved, so that a complete solution for the original problem is obtained at the end of the last iteration.

The first R&F approach set partitions of the integer variables according to the main decisions of the problem. First, only the variables which determine the products to be produced are considered as integer, then the ones related to the configuration of the conveyors, then the ones related to the molds attached to the machine, and finally the setup variables are considered integer. Some overlapping in these subsets of variables are allowed to ensure that a feasible solution can be found. The second R&F approach sets partitions of variables according to the time structure of the original formulation, so that subproblems are created according to each subperiod $s \in S$.

The first R&F is performed once, then the patterns used in the solution obtained are appended to a list of patterns denoted by ξ, and next the second R&F is performed. Some local branching inequalities are added to the formulation before the second R&F is performed to avoid that patterns already generated appear in the solution of the second R&F approach.

5.3.2 Phase II

An MIP formulation for lot-sizing and scheduling is solved in this phase. This formulation uses the molding patterns generated in Phase I and some data related to them as input parameters. The MIP model then decides which molding patterns among the ones defined as input are used, how long they are used for, and how they are sequenced. Although the MIP model in this phase involves lot-sizing and scheduling decisions, which is by itself an NP-hard problem Bitran and Yanasse (1982), CPLEX is efficient in solving problem instances of reasonable size. Using an MIP model for this phase highlights even more the importance of generating patterns likely to appear in good quality solutions, so that the set of generated patterns can be kept small to avoid large CPU times at this phase.

5.3.3 Phase III

This phase is used to further improve and refine the solution found in Phase II by using a fix-and-optimize (F&O) heuristic. The solution from Phase II is set as an initial solution. The original formulation is divided into several partitions, so that at each iteration of the F&O, one partition of variables are reoptimized whereas the others remain fixed. We set partitions of variables according to the set of time periods T of the planning horizon. Therefore, at each iteration the variables indexed to a given period $t^* \in T$ and the subperiods belonging to this period (i.e., $s \in S_{t^*}$) are reoptimized whereas the others remain fixed. As this F&O approach uses the original model of the problem, different patterns from the ones considered as input for Phase II can be generated, and therefore a better solution for the problem may be obtained as a result.

6 Computational results

Computational experiments were performed to analyze and compare the solution approaches proposed in this paper, as well as the effect of the symmetry-breaking constraints in the formulations. The data sets tested are based on real data provided by a company, which produces molded pulp packages for eggs and fruits, and sector information on molded pulp equipment manufacturers. We present results for 10 different groups of 10 instances each, which represent production environments in the same industry with different characteristics and equipment sizes. Demand levels were defined based on the company data. Characteristics of the production line, i.e., the number of conveyors, number of machine faces, width of products, faces, and conveyors, were set according to the company environment and information collected from equipment manufacturers in the molded pulp industry. For all the experiments we use Python 2.7 and solver CPLEX 12.6 on a computer with processor Intel(R) Xeon(R) X5675 / 3.07GHz and 16 GB of RAM.

Sections 6.1 and 6.2 present the results of the computational experiments performed for Model 1 and 2, respectively. These experiments include the performance of the exact methods for different formulations of the models that include symmetry-breaking constraints (SBC) presented in Section 4, as Table 1 summarizes. We performed in total 1,280 runs to analyze the performance of the exact methods and valid inequalities proposed. All these experiments for models 1 and 2 were limited to 10,800 seconds per instance. For Model 1, we performed 300 test runs as a result of solving the 30 corresponding instances using 5 different variants of this formulation (i.e., formulations **A**, **B**, **C**, **D**, and **E**) and two different solution methods (i.e., the linearized model solved by CPLEX and the B&Ch). For Model 2, we performed 980 test runs as a result of solving the 70 corresponding instances by using 7 variants of this formulation and the two exact solutions methods presented in this paper. Section 6.3 presents the results of the 3-phase MIP-based heuristic. This heuristic was tested for the data sets for Model 2, which comprise 70 problem instances. The detailed results of the approaches presented in these sections are provided in Appendix C of the online supplement.

Table 1: Summary of the computational experiments of the exact methods

Variant	Solution method		Formulation	Group (# inst)	Total run
Model 1	Linearized MIP model solved by CPLEX	**A**	Model 1	T2P5_F6K2A1 (10)	300
		B	Model 1+SBC (31)	T2P8_F6K2A1 (10)	
		C	Model 1+SBC (32)	T2P8_F6K3A1 (10)	
	Reformulation solved by the B&Ch	**D**	Model 1+SBC (31) & (32)		
		E	Model 1+SBC (31) & (32)+ bp*		
Model 2	Linearized model solved by CPLEX	**A**	Model 2	T2P5_F6K2A2 (10)	980
		B	Model 2+SBC (31)	T2P8_F6K2A2 (10)	
		C	Model 2+SBC (32)	T2P8_F6K3A3 (10)	
		D	Model 2+SBC (33)	T4P5_F6K2A2 (10)	
	Reformulation solved by the B&Ch	**E**	Model 2+SBC (32) & (33)	T4P5_F6K3A3 (10)	
		F	Model 2+SBC (31),(32) & (33)	T4P8_F6K2A2 (10)	
		G	Model 2+SBC (31), (32) & (33)+ bp*	T4P8_F6K3A3 (10)	

bp*: branching parameters are set to give higher priority to variables v_s

6.1 Results for Model 1

We tested three groups of instances where each group contains 10 instances in order to compare the performance of the solution approaches for Model 1. All these instances represent a single production line, where the molds attached to the machine are of the same width.

The major characteristics of each group of instances for Model 1 are described in Table 2. The group names are defined as follows: "T" indicates the number of periods, "P" the number of products, "F" the number of machine faces, "K" the number of conveyors, and "A" the number of possible arrangements. Each time period is considered as a week, in which the production line operates seven days per week, 24 hours per day. The groups "F6K2" represent the production line of the real company, which consists of a 6-faces machine and two conveyors. The relationship between the width of the machine faces and conveyors is such that $l^* < L < 2l^*$, where l^* is the conveyors width, so the parameter $H_k = 1$, $\forall k \in K$ in these cases. Groups "F6K3" consist of problem instances which represent a larger production setting with a 6-faces machine and three conveyors. The machine faces of these groups are bigger compared to the ones in "F6K2", so that the width of the faces is exactly twice the conveyor width, i.e., $L = 2l^*$ and $H_k = 2$. Note that the number of possible arrangements of all groups is equal to 1 and this thus meets the assumptions of Model 1. This arrangement consists of three separating lanes of the same width.

Table 2: Characteristic of the groups of instances for Model 1

		Planning horizon		Products	Production Line			
Group	# inst.	#per. $\vert T\vert$	#subper. $\vert S\vert$	#prod. $\vert N\vert$	#faces $\vert F\vert$	#conv. $\vert K\vert$	#arrang. $\vert E\vert$	#lanes $\vert P\vert$
T2P5_F6K2A1	10	2	8	5	6	2	1	3
T2P8_F6K2A1	10	2	8	8	6	2	1	3
T2P8_F6K3A1	10	2	8	8	6	3	1	3

Table 3 presents the average results for each group of 10 instances, for each formulation tested (i.e., **A**, **B**, **C**, **D**, and **E** described in Table 1). The branching priority considered in variant **E** was selected based on preliminary tests, whose results showed an improved performance when a higher priority is given to variables v_s. The information reported are the averages of the upper bounds, lower bounds, optimality gap, elapsed time, and number of nodes. The number of optimal solutions found in each group and information related to the reason for stopping the program is also included. We indicate how many instances were stopped due to (1) finding a solution within the optimality tolerance, (2) reaching the time limit, and (3) running out of memory. For the B&Ch, the average number of added cuts is also reported in the last column of Table 3.

First we analyze the effect of the symmetry-breaking constraints within each solution approach separately, i.e., we look at the differences among the formulations **A**, **B**, **C**, **D**, and **E** for the linearized model and the B&Ch. We observe that, for Model 1, the performance of B&Ch is significantly improved when all the symmetry constraints are added to the reformulation. In addition to that, a reduction in the average CPU time is achieved for two out of the three groups when the branching rule is set to give higher priority to variables v_s. On the other hand, for the linearized model solved by CPLEX, the best performing formulation is different among the groups of tested instances. In this case, the best results are achieved for most of the instances when the symmetry-breaking constraints are added individually, i.e., cases **B** and **C**, instead of including all of them in the formulation.

The B&Ch results show that its best performance is achieved by including all the symmetry-breaking constraints in Model 1, i.e., formulations **D** and **E**. Its performance for the cases **A** and **B** is very poor and involves memory issues for most instances. The behavior of this B&Ch method for the cases with no symmetry-breaking constraints can be explained by the form of the logic-based Benders cuts (38)–(39). As described, these cuts force variable q_{is} to be feasible for the original nonlinear formulation if the setup state in a subperiod s is the same as a given solution of the relaxed formulation. As these cuts are stated in terms of variable y_{ipks} in Model 1, and different values for these variables may lead to the same setup state as we described in Section 4, many symmetric solutions appear and therefore many cuts need to be added for these symmetric solutions. Thus, memory issues may arise and the convergence of the B&Ch is indeed very slow for the cases where the symmetry-breaking constraints related to variable y_{ipks} are not considered. When the B&Ch runs out of memory, no information about the number of nodes and number of cuts are provided; however, the upper bounds, gap, and elapsed time reported are the ones achieved just before the interruption.

Comparing the best formulation of each approach, highlighted by bold text in Table 3, clearly shows that the B&Ch algorithm outperforms the linearized version using CPLEX. For the smallest group tested T2P5_F6K2A1, the B&Ch found optimal solutions for all the instances within approximately 65 seconds, whereas using the linearized model only an optimal solution for the first instance is found before reaching the time limit. For group T2P8_F6K2A1, the B&Ch has a better performance and the differences go beyond better lower bounds and computing times. In this case the B&Ch found optimal solutions for 8 out of the 10 instances, whereas no optimal solutions are found by solving the linearized model using CPLEX. The average gap using the B&Ch is reduced by 26.5% compared to the average gap of the linearized model solved by CPLEX for this group.

For the last group of instances, both approaches found optimal solutions for the 10 instances. However, the B&Ch solves the problems more efficiently, since its average computing time is approximately 3.3 times less than the average time of the linearized model. Note that group T2P8_F6K3A1 seems easier to be solved than the group of instances T2P8_F6K2A1, even though the former represents a bigger production environment.

Table 3: Summary of the solution methods' performance for Model 1

		Linearized MIP model solved by CPLEX							Reformulation solved by the B&Ch algorithm							
Group	Case	Upper B.*	Lower B.*	Gap*	Time (s)*	# nodes*	# opt.	Stop crit.	Upper B.*	Lower B.*	Gap*	Time (s)*	# nodes*	# opt.	Stop crit.	# cuts*
T2P5_F6K2A1	**A**	17336.0	14896.6	14.78%	10181.8	132776.2	1	1/9/0	17336.0	14112.4	19.28%	6900.2	-	0	0/2/8	-
	B	17336.0	14083.9	19.73%	10680.5	181914.1	1	1/9/0	17336.0	14331.4	17.94%	5857.4	-	0	0/0/10	-
	C	**17336.0**	**14935.6**	**14.57%**	**9909.5**	**259102.5**	**1**	**1/9/0**	17336.0	17335.3	0.00%	215.8	13086.3	10	10/0/0	304.1
	D	17336.0	14882.1	14.89%	9768.4	241317.3	1	1/9/0	17336.0	17335.9	0.00%	85.3	4872.9	10	10/0/0	290.9
	E	17336.0	14884.0	14.88%	9810.7	322230.6	1	1/9/0	**17336.0**	**17335.8**	**0.00%**	**64.7**	**2938.5**	**10**	**10/0/0**	**293.7**
T2P8_F6K2A1	**A**	27057.4	19220.4	28.93%	10800.1	33582.1	0	0/10/0	26729.0	23120.9	13.48%	10800.0	268447.1	0	0/10/0	460.9
	B	27211.3	19728.0	27.49%	10800.1	17118.1	0	0/10/0	26668.2	24559.0	7.89%	9423.8	-	0	0/7/3	-
	C	26844.5	19225.0	28.36%	10800.0	21406.2	0	0/10/0	26668.2	23245.5	12.83%	10800.3	74370.8	0	0/10/0	346.4
	D	26696.5	19230.5	27.95%	10800.0	14892.6	0	0/10/0	**26689.0**	**26617.6**	**0.26%**	**7306.8**	**53320.8**	**8**	**8/2/0**	**336.1**
	E	**26686.9**	**19538.6**	**26.80%**	**10800.3**	**13763.3**	**0**	**0/10/0**	26668.2	26502.6	0.62%	6733.7	44527.5	8	8/2/0	303.9
T2P8_F6K3A1	**A**	23328.4	23125.6	0.87%	10800.0	26017.0	1	1/9/0	23555.0	20717.7	12.04%	5026.1	-	0	0/0/10	-
	B	**23555.0**	**23553.9**	**0.00%**	**6736.0**	**18472.6**	**10**	**10/0/0**	23555.0	20762.4	11.85%	6668.3	-	0	0/2/8	-
	C	23555.0	20593.0	12.59%	10800.0	16517.4	0	0/10/0	23555.0	23153.1	1.70%	5595.4	55532.3	8	8/2/0	411.2
	D	23555.0	22947.1	2.57%	7601.1	11480.2	6	6/4/0	23555.0	23553.5	0.01%	2646.1	27758.9	10	10/0/0	407.0
	E	23555.0	23447.0	0.46%	6823.0	10821.7	8	8/2/0	**23555.0**	**23554.1**	**0.00%**	**2012.3**	**21062.7**	**10**	**10/0/0**	**380.5**

A: Model 1 without symmetry-breaking constraints; **B**: Model 1 + Symmetry-breaking constraints (31); **C**: Model 1 + Symmetry-breaking constraints (32); **D**: Model 1 + Symmetry-breaking constraints (31) & (32); **E**: Model 1 + Symmetry-breaking constraints (31) & (32) + Branching priority on variable v_s

$*$ Average

$\circ$ (1)/(2)/(3): (1) # instances with a solution within the optimality tolerance; (2) # instances where the time limit was reached; (3) # instances which ran out of memory.

This may be explained by the fact that all instances in both groups have the same demand levels and time capacity, but T2P8_F6K3A1 involves a bigger machine where a greater number of molds can be attached, and it therefore can produce larger production quantities per time unit. This hence suggests that the tightness of the machine capacity has a significant effect in the complexity of the instances, even more than the problem size in this case.

6.2 Results for Model 2

Experiments for Model 2 comprise 7 different groups of instances as described in Table 4. The first 3 groups are similar to the ones tested for Model 1. However, theses instances include products of different sizes, so that conveyors can be set up according to more than one possible arrangement. These three first groups comprise a planning horizon of 2 periods subdivided in 8 subperiods, and the demand and inventory levels were generated uniformly based on the average actual demand. The last four groups of instances represent a longer planning horizon with 4 periods sub-divided into 16 subperiods in total. These groups consider real data regarding the demand and initial inventory levels, according to the data provided by the company. They also involve more expensive backlogging costs, compared to the instances for Model 1 and the first three groups for Model 2.

Table 4: Characteristic of the groups of instances for Model 2

		Planning horizon		Products	Production Line			
Group	# inst.	#per. $\|T\|$	#subper. $\|S\|$	#prod. $\|N\|$	#faces $\|F\|$	#conv. $\|K\|$	#arrang. $\|E\|$	#lanes $\|P\|$
T2P5_F6K2A2	10	2	8	5	6	2	2	≤ 3
T2P8_F6K2A2	10	2	8	8	6	2	2	≤ 3
T2P8_F6K3A3	10	2	8	8	6	3	3	≤ 3
T4P5_F6K2A2	10	4	16	5	6	2	2	≤ 3
T4P5_F6K3A3	10	4	16	5	6	3	3	≤ 3
T4P8_F6K2A2	10	4	16	8	6	2	2	≤ 3
T4P8_F6K3A3	10	4	16	8	6	3	3	≤ 3

Table 5 presents the average results for the first three groups of instances for Model 2. These experiments also include tests with symmetry-breaking constraints (33), so both solution approaches are tested for 7 different formulations as described in Table 1.

The B&Ch is able to prove the optimality of most solutions reported in Table 5, i.e., 28 out of the 30 solutions are proved optimal, whereas the linearized model solved by CPLEX proved the optimality of only 17 out of the 30 solutions in these data sets. Furthermore, the B&Ch algorithm clearly outperforms the linearized model solved by CPLEX with respect to computing time, particularly for groups T2P5_F6K2A2 and T2P8_F6K2A2. In the case where both methods found optimal solutions for all the instances, i.e., group T2P8_F6K2A2, the B&Ch solved these problem instances approximately 38 times faster than the linearized model resolution by CPLEX.

Based on our findings for these sets of instances and the results for Model 1 presented in Table 3, we can indicate that the integrated formulation proposed in this paper is able to provide good quality solutions for these sets of real-size instances. However, using CPLEX to solve the linearized model is not effective to prove the quality of those found solutions. The large average gaps reported for the linearized model are due to weak lower bounds rather than the quality of the upper bounds. Although these lower bounds are improved by including the symmetry-breaking constraints, solving the linearized model by CPLEX still fails to prove the optimality of these solutions. On the other hand, the B&Ch approach is more effective and determined most optimal solutions in short computing times. The reformulation solved by this approach is less complex than the linearized model and the lower bounds are quickly improved by including the logic-based Benders cuts. The symmetry-breaking constraints significantly improve the convergence and effectiveness of the B&Ch.

Table 6 presents the average results for the last four groups of instances for Model 2. Solving these sets of instances became very challenging, due to the instance sizes. In addition, these instances particularly consider the demand variations that may surge in practice, which affect the capacity consumption and consequently influence the complexity of the instance.

Similar to the results for the three first groups, the best performance of the methods for most instances is achieved when all the symmetry-breaking constraints are included. These results show how challenging it is to find optimal solutions for instances with a longer planning horizon and different production environments. Note that the B&Ch solved 6 out of the 40 instances to optimality, whereas the linearized model solved by CPLEX found only 1 optimal solution for these data sets.

Although these instances are more complex for both exact approaches, the performance of the B&Ch algorithm is superior compared to solving the linearized model by CPLEX, since better upper bounds and stronger lower bounds are achieved for all the four groups. On average, the B&Ch lower bounds are between 2.15% and 12.21% better than the lower bounds obtained by the linearized model. The B&Ch algorithm found upper bounds which are, on average, 0.99% to 3.57% better than the ones obtained by the linearized model solved by CPLEX.

In general, the improvements achieved by adding the symmetry-breaking constraints are consistent over all the instance groups. They improved the lower bounds of the linearized model and are fundamental for the efficiency of the B&Ch algorithm, particularly for the first six groups of instances. The most significant improvements are achieved for the cases where constraints (32) were included in the formulation. This can be explained by the fact that a larger number of symmetric solutions are expected for multiple assignments of products to the conveyors lanes (i.e., symmetry type related to variables y_{ipks}) rather than for the other types of symmetry described here.

The superiority of the B&Ch algorithm over the linearized model is also consistent over all the results. In particular, for the first six groups of instances, the B&Ch found optimal solution for 56 out of the 60 instances in acceptable computing time whereas the linearized model found optimal solutions for only 28 instances after 3 hours. Note that the B&Ch is particularly effective to solve the data sets with 5 products and 2 time periods. The number of nodes explored by this algorithm is far smaller than in the linearized model for these groups, as solutions are found and proved optimal by the B&Ch within the first two minutes. On the other hand, although the B&Ch yields better results than the linearized model for the last four groups of instances, these data sets are difficult to be efficiently solved by both approaches. The cuts added in the B&Ch do not appear as effective as for the smallest and medium instances, as they still have large optimality gaps after 3 hours and the number of nodes in the search tree increased significantly. This may be explained by the size of these problems, which implies a larger number of variables to branch on in the search tree.

We also run some computational tests to analyze the performance of the solution procedures with respect to critical parameters as demand variance and backlogging costs, and present them in Appendix D of the online supplement. These results show that the B&Ch algorithm still outperforms the linearized model for a set of instances with variations in these parameters. Higher backlogging costs and a higher demand variance make the instances more difficult to be solved and affect the performance to both production approaches. However, the results for a sample of instances showed that the B&Ch is still able to find optimal and good quality solutions for these cases whereas the linearized model presents large optimality gap after 3 hours (i.e., solutions with average gaps up to 21%). Instances with lower backlogging costs appear to be less complex and can be effectively solved by both solution approaches. Results show that the linearized model and the B&Ch could find optimal solutions in a few second for these cases.

6.3 Heuristic results

This section presents the results of the 3-phase heuristic described in Section 5.3. This method aims to find competitive solutions in shorter computing times, particularly for the instances tested for Model 2 where finding optimal solutions is challenging, even for the best performing exact approach.

Table 7 presents the average results for the first three groups and Table 8 presents the average results for the last four groups of instances for Model 2 described in Table 4. Results show the objective function values obtained at the end of each phase of the heuristic, the accumulated CPU time after each phase and the deviation of the solution value, compared with the solution values obtained by the best performing formulation of the B&Ch algorithm reported in Tables 5 and 6.

Table 5: Summary of the solution methods' performance for Model 2 (Groups T2P5_F6K2A2, T2P8_F6K2A2 and T2P8_F6K3A3)

		Linearized MIP model solved by CPLEX							Reformulation solved by the B&Ch algorithm							
Group	Cases	Upper B.*	Lower B.*	Gap*	Time (s)*	# nodes*	# opt.	Stop crit.	Upper B.*	Lower B.*	Gap*	Time (s)*	# nodes*	# opt.	Stop crit.	# cuts*
T2P5_F6K2A2	**A**	22271.1	20646.7	7.29%	10800.0	289636.2	0	0/10/0	22271.1	17765.8	20.24%	9397.9	-	0	0/5/5	-
	B	22271.1	18583.4	16.52%	10800.0	238911.1	0	0/10/0	22271.1	17765.8	20.21%	7668.4	-	0	0/10/0	-
	C	22271.1	20044.5	9.96%	10800.0	784074.4	0	0/10/0	22271.1	22270.2	0.00%	211.2	24239.5	10	10/0/0	305.2
	D	**22271.1**	**20843.9**	**6.35%**	**10157.6**	**290161.3**	**2**	**2/8/0**	22271.1	17715.4	20.46%	9401.2	-	0	0/5/5	-
	E	22271.1	20060.4	9.90%	10800.0	736585.1	0	0/10/0	22271.1	22270.0	0.01%	146.1	16254.6	10	10/0/0	316.5
	F	22271.1	20041.2	9.98%	10800.0	959027.2	0	0/10/0	22271.1	22270.5	0.00%	94.7	10926.9	10	10/0/0	318.5
	G	22271.1	20082.0	9.81%	10800.0	888296.7	0	0/10/0	**22271.1**	**22270.4**	**0.00%**	**85.7**	**10033.0**	**10**	**10/0/0**	**312.2**
T2P8_F6K2A2	**A**	28263.4	28110.5	0.55%	8161.9	38568.7	7	7/3/0	28263.4	25935.3	8.24%	9755.0	-	0	0/5/5	-
	B	28263.4	27656.4	2.16%	9513.6	100895.6	3	3/7/0	28263.4	26029.7	7.89%	8553.7	-	0	0/1/9	-
	C	28263.4	27480.3	2.75%	10800.0	129852.5	0	0/10/0	28263.4	28263.0	0.00%	359.7	16788.7	10	10/0/0	346.9
	D	28263.4	27954.1	1.09%	8140.8	100507.9	6	6/4/0	28263.4	25405.0	10.11%	10800.7	-	0	0/0/10	-
	E	28263.4	28102.8	0.57%	4791.4	26418.3	9	9/1/0	28263.4	28261.8	0.01%	1547.8	65420.3	10	10/0/0	362.2
	F	28263.4	28199.1	0.23%	5822.1	104996.0	8	8/2/0	28263.4	28262.9	0.00%	270.5	9472.1	10	10/0/0	362.4
	G	**28263.4**	**28263.3**	**0.00%**	**2937.2**	**148931.6**	**10**	**10/0/0**	**28263.4**	**28263.1**	**0.00%**	**77.8**	**2522.7**	**10**	**10/0/0**	**331.8**
T2P8_F6K3A3	**A**	33700.4	29072.7	13.74%	10800.0	40175.2	0	0/10/0	33700.4	27559.1	18.22%	10264.8	-	0	0/8/2	-
	B	33700.4	31233.8	7.30%	10800.0	39241.7	0	0/10/0	33700.4	27559.1	18.22%	9725.6	-	0	0/6/4	-
	C	33700.4	28533.9	15.32%	10800.0	34596.9	0	0/10/0	33700.4	31321.0	7.05%	10800.2	128411.2	0	0/10/0	869.7
	D	33700.4	30295.3	10.12%	10800.0	49188.1	0	0/10/0	33700.4	27529.2	18.31%	10414.0	-	0	0/8/2	-
	E	33700.4	30079.5	10.69%	10800.0	42576.0	0	0/10/0	33700.4	31632.7	6.13%	10800.0	150332.3	0	0/10/0	807.3
	F	33700.4	31440.6	6.66%	10492.9	36670.7	1	1/9/0	33700.4	33242.2	1.36%	8736.7	141686.4	7	7/3/0	837.8
	G	**33700.4**	**32791.0**	**2.68%**	**9157.6**	**31187.1**	**5**	**5/5/0**	**33700.4**	**33405.7**	**0.86%**	**7995.6**	**115460.1**	**8**	**8/2/0**	**832.6**

A: Model 2 without symmetry-breaking constraints; **B**: Model 2 + Symmetry-breaking constraints (31); **C**: Model 2 + Symmetry-breaking constraints (32); **D**: Model 2 + Symmetry-breaking constraints (33); **E**: Model 2 + Symmetry-breaking constraints (32) & (33); **F**: Model 2 + Symmetry-breaking constraints (31), (32) & (33); **G**: Model 2 + Symmetry-breaking constraints (31), (32) & (33) + Branching priority on variables v_s

∗ Average

∘ (1)/(2)/(3): (1) # instances with a solution within the optimality tolerance; (2) # instances where the time limit was reached; (3) # instances which ran out of memory.

Table 6: Summary of the solution methods' performance for Model 2 (Groups T4P5_F6K2A2, T4P5_F6K3A3, T4P8_F6K2A2 and T4P8_F6K3A3)

		Linearized MIP model solved by CPLEX							Reformulation solved by the B&Ch							
Group	Cases	Upper B.*	Lower B.*	Gap*	Time (s)*	# nodes*	# opt.	Stop crit.	Upper B.*	Lower B.*	Gap*	Time (s)*	# nodes*	# opt.	Stop crit.	# cuts*
T4P5_F6K2A2	**A**	105413.4	77768.5	27.80%	10800.0	57789.7	0	0/10/0	102522.5	82241.9	22.23%	9671.8	-	0	0/3/7	-
	B	106951.5	75719.2	29.79%	10800.0	59386.7	0	0/10/0	104636.6	82596.8	23.37%	9885.8	-	0	0/6/4	-
	C	103928.0	80675.0	24.71%	10800.0	58781.5	0	0/10/0	102063.2	83472.5	20.17%	10654.8	-	0	0/9/1	-
	D	105549.7	78073.5	28.30%	10800.0	58326.4	0	0/10/0	103229.9	82280.2	22.50%	9214.6	-	0	0/6/4	-
	E	104030.9	80324.7	25.42%	10800.0	70874.4	0	0/10/0	102442.5	86724.9	17.49%	10059.4	-	0	0/7/3	-
	F	104168.1	81739.3	23.82%	10800.0	63423.6	0	0/10/0	101925.0	90745.7	12.88%	9239.7	360307.1	2	2/8/0	512.0
	G	**104903.2**	**83266.1**	**22.68%**	**10800.0**	**67543.9**	**0**	**0/10/0**	**102213.3**	**93050.7**	**10.31%**	**8321.8**	**265494.1**	**4**	**4/6/0**	**512.4**
T4P5_F6K3A3	**A**	95909.1	65251.4	32.48%	10800.0	66098.6	0	0/10/0	97614.0	65027.3	33.44%	10800.0	214151.4	0	0/10/0	2886.2
	B	98603.8	70089.5	29.19%	10059.4	60927.7	1	1/9/0	96935.0	66862.4	30.90%	10800.0	205463.1	0	0/10/0	3246.3
	C	98045.2	64965.0	32.82%	10800.1	54328.3	0	0/10/0	97524.3	66136.1	31.74%	10293.7	191381.3	1	1/9/0	1214.4
	D	98528.4	68916.4	30.78%	10800.0	72608.9	0	0/10/0	101032.7	65864.3	33.73%	10336.2	-	0	0/6/4	-
	E	97209.1	68538.6	30.49%	10800.0	55799.1	0	0/10/0	97705.4	68394.7	30.36%	10039.1	225234.7	1	1/9/0	998.1
	F	97779.0	72487.7	26.04%	10254.9	53343.1	1	1/9/0	96682.0	74057.9	23.66%	9201.4	188706.6	2	2/8/0	1293.5
	G	**96914.3**	**72198.5**	**25.95%**	**10176.1**	**41785.9**	**1**	**1/9/0**	**95952.8**	**73749.2**	**22.72%**	**9128.1**	**137903.5**	**2**	**2/8/0**	**1467.5**
T4P8_F6K2A2	**A**	172048.7	99604.6	39.86%	10800.0	16867.7	0	0/10/0	159921.7	107097.7	31.27%	10800.0	160164.0	0	0/10/0	832.1
	B	167552.4	103101.3	36.53%	10800.1	15832.4	0	0/10/0	161096.3	110064.8	29.89%	10800.0	158038.7	0	0/10/0	951.2
	C	163979.8	99769.1	37.38%	10800.0	13424.7	0	0/10/0	158190.1	110588.2	28.67%	10800.0	96330.2	0	0/10/0	420.0
	D	166686.2	105139.1	35.65%	10800.0	19186.1	0	0/10/0	157942.9	112589.1	27.89%	10781.2	-	0	0/9/1	840.3
	E	**164245.5**	**105993.1**	**34.24%**	**10800.0**	**15970.9**	**0**	**0/10/0**	157274.6	112618.0	27.58%	10800.0	150353.2	0	0/10/0	490.7
	F	169313.7	107324.4	35.73%	10800.0	13829.6	0	0/10/0	159926.1	114770.0	26.86%	10800.0	127024.9	0	0/10/0	530.3
	G	165507.8	104982.8	34.54%	10800.0	13565.7	0	0/10/0	**158387.1**	**118939.4**	**23.51%**	**10800.7**	**90477.5**	**0**	**0/10/0**	**603.5**
T4P8_F6K3A3	**A**	154295.7	92999.4	38.91%	10800.0	18504.6	0	0/10/0	154215.6	96435.1	36.41%	10800.0	73800.3	0	0/10/2	1917.2
	B	154973.8	98817.6	35.35%	10800.0	17957.8	0	0/10/0	151949.5	99409.1	33.43%	10800.1	84952.0	0	0/10/0	2106.4
	C	152513.9	93909.8	35.55%	10800.1	14091.9	0	0/10/0	150168.2	96123.9	35.09%	10800.1	64257.5	0	0/10/0	1014.0
	D	150467.9	96778.8	35.19%	10800.0	18415.2	0	0/10/0	151441.6	100775.9	32.55%	10800.0	78745.9	0	0/10/0	1472.5
	E	153873.3	97778.9	35.57%	10800.0	17611.9	0	0/10/0	150120.8	99807.2	32.44%	10800.0	68040.1	0	0/10/0	872.6
	F	**155657.9**	**101898.3**	**34.04%**	**10800.0**	**14255.9**	**0**	**0/10/0**	**150091.7**	**104953.3**	**28.80%**	**10800.0**	**63494.9**	**0**	**0/10/0**	**1050.3**
	G	155479.3	98751.4	35.69%	10800.0	11846.0	0	0/10/0	152662.2	105746.3	29.72%	10800.1	43286.6	0	0/10/0	1403.1

A: Model 2 without symmetry-breaking constraints; **B**: Model 2 + Symmetry-breaking constraints (31); **C**: Model 2 + Symmetry-breaking constraints (32); **D**: Model 2 + Symmetry-breaking constraints (33); **E**: Model 2 + Symmetry-breaking constraints (32) & (33); **F**: Model 2 + Symmetry-breaking constraints (31), (32) & (33); **G**: Model 2 + Symmetry-breaking constraints (31), (32) & (33) + Branching priority on variables v_s

∗ Average

∘ (1)/(2)/(3): (1) # instances with a solution within the optimality tolerance; (2) # instances where the time limit was reached; (3) # instances which ran out of memory.

Note that, although the first stage of this heuristic consists of generating feasible process configurations to be used in Phase II, it is also possible to obtain feasible solutions after this phase, since the algorithm developed uses MIP heuristics based on the original formulation. The time spent solving the MIP model of Phase II is very short, i.e., few seconds for all the instances, since the set of possible patterns generated in Phase I is usually small and CPLEX can solve the MIP problems efficiently. We observe that, for all the groups, the solutions obtained after Phase II are already very competitive in comparison to the ones provided by the B&Ch as most of them are in fact optimal and they are obtained in a much shorter computing time.

For the first groups of instances in Table 7, the heuristic developed obtained the same solutions as the B&Ch algorithm, which found optimal solutions for 28 out of the 30 instances in these data sets. For groups T2P5_F6K2A2 and T2P8_F6K2A2, the heuristic found the optimal solutions in approximately 25% and 68%, respectively, of the time needed by the best performing B&Ch. For group T2P8_F6K3A3, the heuristic found the same solutions as the B&Ch algorithm approximately 6 times faster (i.e., approximately 1.8 hours faster), on average.

Table 7: Heuristic average results for groups T2P5_F6K2A2, T2P8_F6K2A2, and T2P8_F6K3A3

	Phase 1			Phase 2			Phase 3		
Group	Obj.Val.	Time	Dev.^{a} (min;max)	Obj.Val.	Time^b	Dev.^{a}(min;max)	Obj.Val.	Time^b	Dev.^{a}(min;max)
T2P5_F6K2A2	31154.6	13.6	39.8%(35%;57%)	22271.1	13.7	0.0%(0%;0%)	22271.1	21.7	0.0%(0%;0%)
T2P8_F6K2A2	49331.1	38.1	29.8%(70%;81%)	28263.4	38.2	0.0%(0%;0%)	28263.4	52.8	0.0%(0%;0%)
T2P8_F6K3A3	51580.4	1304.2	53.1%(50%;58%)	33700.4	1304.3	0.0%(0%;0%)	33700.4	1327.0	0.0%(0%;0%)

Time in seconds; ^{a} Deviation compared to the best solution found by the B&Ch; ^{b} Cumulative time

For the last four groups, the heuristic developed found competitive solutions in significantly less average computing time. For the first set of instances T4P5_F6K2A2, the heuristic found competitive solutions in approximately 1% of the average computing time spent by the best performing exact approach. For the group T4P5_F6K3A3, the average deviation of the heuristic solutions is 2.04%, with an average time of approximately 4.1% of the average time for the B&Ch. Particularly in this group of instances, as well as groups T4P8_F6K2A2 and T4P8_F6K3A3, the heuristic found better solutions for some instances. In the best case for group T4P5_F6K3A3, the heuristic found a solution which is 5.47% better than the one provided by the B&Ch.

Table 8: Heuristic average results for groups T4P5_F6K2A2, T4P5_F6K3A3, T4P8_F6K2A2 and T4P8_F6K3A3

	Phase 1			Phase 2			Phase 3		
Group	Obj.Val.	Time	Dev.^{a}(min;max)	Obj.Val.	Time^b	Dev.^{a}(min;max)	Obj.Val.	Time^c	Dev.^{a}(min;max)
T4P5_F6K2A2	115968.9	57.6	14.0%(6%;27%)	103007.2	59.4	0.7%(0%;3%)	102656.1	85.8	0.4%(0%;3%)
T4P5_F6K3A3	110683.9	184.4	16.0%(5%;26%)	98359.9	186.8	2.6%(-2%;8%)	97638.7	371.0	2.0%(-6%;9%)
T4P8_F6K2A2	179152.3	740.5	13.5%(2%;26%)	158107.8	754.0	-0.1%(-4%;6%)	156246.8	901.6	-1.1%(-4%;3%)
T4P8_FK3A3	159788.4	856.8	7.6%(-5%;20%)	145894.5	876.2	-2.3%(-10%;2%)	145550.8	991.4	-2.5%(-10%;2%)

Time in seconds; ^{a} Deviation compared to the best solution found by the B&Ch; ^{b} Cumulative time

For groups T4P8_F6K2A2 and T4P8_F6K3A3, the heuristic found better solutions than the ones provided by the B&Ch whereas its computing times are at least 10.9 times shorter. In group T4P8_F6K2A2, the heuristic found solutions at least as good as the B&Ch for 8 out of the 10 instances. On average, solutions in this group are approximately 1.05% better than those of the B&Ch and the CPU time is approximately 12 times shorter than the B&Ch results. Finally for the group T4P8_F6K3A3, the heuristic provided better solutions for 9 out 10 instances. The best case happened for instance 5, where the heuristic solution is approximately 9.7% better than the solution found by the B&Ch. The average CPU time for this group is approximately 10.9 faster than the B&Ch results, with an average improvement in solution quality of 2.54%.

7 Concluding remarks and research directions

We presented an exact approach based on a branch-and-check framework and a 3-phase heuristic to solve an integrated problem, which combines process configuration, lot-sizing, and scheduling decisions with sequence-dependent setup times and costs. A particular application of this integrated problem in a molded pulp packages company is considered in this paper. Our previous work in Martínez et al. (2018) already showed that the solutions obtained by solving the linearized formulations using CPLEX are better than the schedules in practice. This work showed that, in general, the model solutions involve production plans with lower total costs, which also fulfill a higher percentage of the total demand, reduce significantly the total setup times and costs (i.e., above 50%), and reduce the backlogging and inventory levels compared to the production plans in practice. As the solution methods proposed in this paper significantly outperform the linearized model, we can therefore state that the branch-and-check algorithm and the 3-phase heuristic proposed here provide better solutions than the schedules in practice.

In the branch-and-check algorithm, a relaxed formulation is solved and the logic-based Benders cuts are generated and added during the branch-and-bound process. The exact framework presented in this paper can be applied to other applications of production planning problems, where the production amounts and process configurations must be simultaneously determined. The formulations and reformulations presented here, mainly the specific constraints related to the process configuration decisions, may inspire mathematical models for similar processes which includes molding stages, as injection molding and the production of plastic parts, or production processes which require to synchronize the setup state of different production equipments.

Comparisons of the reformulation solved by the B&Ch with the linearized model solved by CPLEX for a set of instances based on real-word operations show that the B&Ch clearly outperforms the linearized model. As the original formulations for this problem allow symmetric solutions, we proposed three sets of symmetry-breaking constraints to resolve this issue. The proposed symmetry-breaking constraints appear to be particularly effective for the branch-and-check algorithm and could significantly improve the performance of this method.

Results also show that the heuristic approach proposed in this paper provides competitive solutions in much shorter computing times, when compared with the best performing case of the exact algorithms. For the largest data sets, the heuristic developed found better solutions for many of the instances within these groups. In general for the largest instances, heuristic solutions with average deviations between 0.4% and -2.6% from the B&Ch results could be obtained within average computing times between 10.9 and 97 times shorter than the average computing times of the B&Ch.

Extensions of this research may include enhancements for the branch-and-check framework. Other applications for the exact and heuristic approaches will also be considered as future research.

References

Philipp Baumann, Salome Forrer, and Norbert Trautmann. Planning of a make-to-order production process in the printing industry. Flexible Services and Manufacturing Journal, 27(4):534–560, 2015.

Gabriel R Bitran and Horacio H Yanasse. Computational complexity of the capacitated lot size problem. Management Science, 28(10):1174–1186, 1982.

Jin Xin Cao, Der-Horng Lee, Jiang Hang Chen, and Qixin Shi. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. Transportation Research Part E: Logistics and Transportation Review, 46(3):344–353, 2010.

Jean-François Côté, Mauro Dell'Amico, and Manuel Iori. Combinatorial Benders' cuts for the strip packing problem. Operations Research, 62(3):643–661, 2014. doi: 10.1287/opre.2013.1248.

Maxence Delorme, Manuel Iori, and Silvano Martello. Logic based Benders' decomposition for orthogonal stock cutting problems. Computers & Operations Research, 78:290–298, 2017. doi: http://doi.org/10.1016/j.cor.2016.09.009.

Mohammad M Fazel-Zarandi and J Christopher Beck. Solving a location-allocation problem with logic-based Benders' decomposition. In International Conference on Principles and Practice of Constraint Programming, pages 344–351. Springer, 2009.

Bernhard Fleischmann and Herbert Meyr. The general lotsizing and scheduling problem. OR Spectrum, 19:11–21, 1997.

Jonathan Gaudreault, Jean-Marc Frayret, Alain Rousseau, and Sophie D Amours. Combined planning and scheduling in a divergent production system with co-production: A case study in the lumber industry. Computers & Operations Research, 38(9):1238–1250, 2011.

Maud Göthe-Lundgren, Jan T. Lundgren, and Jan A. Persson. An optimization model for refinery production scheduling. International Journal of Production Economics, 78:255–270, 2002.

John N Hooker. Logic-based methods for optimization: combining optimization and constraint satisfaction. Wiley-Interscience series in discrete mathematics and optimization. John Wiley & Sons, 2000.

John N Hooker. Planning and scheduling by logic-based Benders decomposition. Operations Research, 55(3):588–602, 2007.

John N Hooker and Greger Ottosson. Logic-based Benders decomposition. Mathematical Programming, 96(1):33–60, 2003.

Lynwood A. Johnson and Douglas C. Montgomery. Operations research in production planning, scheduling, and inventory control. Wiley, 1974.

José Roberto Luche, Reinaldo Morabito, and Vitória Pureza. Combining process selection and lot sizing models for production scheduling of electrofused grains. Asia-Pacific Journal of Operational Research, 26(3):421–443, 2009.

Karim Pérez Martínez, Eli Angela Vitor Toso, and Reinaldo Morabito. Production planning in the molded pulp packaging industry. Computers & Industrial Engineering, 98:554–566, 2016.

Karim Pérez Martínez, Reinaldo Morabito, and Eli Angela Vitor Toso. A coupled process configuration, lot-sizing and scheduling model for production planning in the molded pulp industry. International Journal of Production Economics, 204:227–243, 2018.

Gislaine Mara Melega, Silvio Alexandre de Araujo, and Raf Jans. Classification and literature review of integrated lot-sizing and cutting stock problems. European Journal of Operational Research, 271(1):1–19, 2018. doi: 10.1016/j.ejor.2018.01.002.

Lei Shi, Yongheng Jiang, Ling Wang, and Dexian Huang. Refinery production scheduling involving operational transitions of mode switching under predictive control system. Industrial & Engineering Chemistry Research, 53: 8155–8170, 2014.

Erlendur S. Thorsteinsson. Branch-and-Check: A Hybrid Framework Integrating Mixed Integer Programming and Constraint Logic Programming, pages 16–30. Springer Berlin Heidelberg, 2001. doi: 10.1007/3-540-45578-7_2.

Laurence A. Wolsey. Integer Programming. Wiley, 1998.