ORIGINAL PAPER

Single Machine Campaign Planning under Sequence Dependent Family Setups and Co-Production

Serkan Kalay[a] and Z. Caner Taşkın[b]

[a]Department of Industrial Engineering, Boğaziçi University, 34342 Bebek, İstanbul, Turkey;
[b]Department of Industrial Engineering, Boğaziçi University, 34342 Bebek, İstanbul, Turkey

ARTICLE HISTORY
Compiled May 20, 2020

ABSTRACT
We investigate tactical level production planning problem in process industries, with float glass manufacturing being the specific application domain. In the presence of high sequence dependent family setup costs, the need for planning production in batches, or campaigns as named in the float glass industry, arises. Moreover, the float glass manufacturing has some unique properties such as partially controllable co-production and uninterruptible production. The motivation of our work is a real life problem encountered at a major float glass manufacturing company in Turkey. We develop two mixed integer programming formulations and investigate some variants to solve the problem. Our formulations are capable of handling different resolutions in input data such as demand forecast expressed in discrete time and setup durations expressed in continuous time. We compare formulations both theoretically and by running computational experiments. Furthermore, we conduct additional experiments to gain insights about characteristics of the generated campaign plans from a business perspective.

KEYWORDS
Sequence dependent family setup; MILP; campaign planning; process industry; co-production

1. Introduction and Literature Review

In this paper, we study a production planning problem in process industries in the presence of sequence dependent family setups and co-production on a single machine. Since process industries are usually capital intensive, cost effectiveness is critical within the manufacturing process and its planning.

Manufacturing, transportation, inventory holding and demand satisfaction related costs are often directly considered in supply chain planning. Nevertheless, loss of efficiency in production line capacity usage can have significant impact on the overall effectiveness especially in process industries. For instance, furnaces used in float glass manufacturing need to be up and running 24/7 due to the continuous production nature of the process even if the produced glass is not conforming with respect to product specifications, or there is insufficient demand. Thus, lost capacity is highly undesirable. Setup times and, if the nature of the process imposes, co-production need to be dealt with to improve effectiveness. Co-production is the phenomenon of producing several

CONTACT Serkan Kalay. Email: serkan.kalay@boun.edu.tr

different products simultaneously usually due to physical or chemical properties of the system Ağralı (2012). Setup can be defined as time, cost and possibly material needed to start manufacturing of a production unit.

Setups can be categorized in different aspects. Sequence dependency is a phenomenon with a significant effect in terms of solution performance of the problem we study and has been investigated by numerous works such as Almada-Lobo, Oliveira, and Carravilla (2008); Haase and Kimms (2000); Toledo, da Silva Arantes, Hossomi, and Almada-Lobo (2016). Regardless of whether the setup type is sequence dependent or independent, we can further categorize setups as either being product or family setups. In family setups, products are grouped into families with respect to certain attributes affecting the setup time and setups arise between production units belonging to different families. There are several studies where authors investigate family setups in both multiple and single machine environments as in Almada-Lobo et al. (2008); Günther (2014) respectively. In discrete time formulations, depending on the timing of setup operations with respect to period boundaries, setup approach is also categorized further as carryover or crossover Fiorotto, Jans, and de Araujo (2017). Moreover, Wittrock (1990) define minor setup such as time incurred on machines of moderate length due to switch from one part to another and major setup of long length due to a switch between parts belonging to different families. However, in float glass manufacturing major and minor setups are both related to family setups, former being related to a change in color whereas latter related to a change in coating or thickness. Stefansdottir, Grunow, and Akkerman (2017) develop a classification scheme for setups, however including cleanings, which can be a key cost driver in process industries such as food and pharmaceuticals. Cleanings can be viewed as another setup type, required due to quality and safety considerations and can further be categorized. A generic mathematical model, which can accurately represent cleanings is presented. In float glass manufacturing, on the other hand, cleanings translate into setups since switching from one color to another requires the furnace and the molten solution to be stabilized in terms of quality of the destination color. Finally, startup is another category of setup, which corresponds to resources spent to start producing any product. However, we do not further elaborate on details of startup setups, since in float glass manufacturing lines operate on a 24/7 basis and startup setups practically only exist when a new production line starts operating for the first time.

Lot sizing and scheduling problem with sequence dependent setup consideration is studied by Guimarães, Klabjan, and Almada-Lobo (2014) with two different approaches. One formulation is based on decisions for setup between products whereas in the other uses a collection of pre-defined sequences. The latter selects a sequence to be executed in the production. However, authors do not explicitly model family setups but only products and longer setups between products corresponding to family aggregation is not analyzed in detail but only present in a single instance of computational experiments. Moreover, the models proposed do not allow for setup crossover, which can be necessary in environments where some of input data is not an integer multiple of micro-period lengths such as setup durations. Miegeville (2005) studies extensively the float glass manufacturing process and develops a MIP for production planning. The model in this study determines whether a product is produced in a time period and that at most one product is allocated to periods. The author do not explicitly address sequence dependent family setup phenomenon. Lime, Grossmann, and Jiao (2011) model the transition between adjacent periods permitting the changeovers between products occur before, across and after the period boundaries. However, fixed number of slots, similar to micro-periods in Guimarães et al. (2014), can result in sub-optimal solutions in cases where input data is sensitive to discretization. Clark, Morabito, and Toso (2010) study production planning for animal nutrition products under sequence-dependent family setups, and formulate a mathematical model based on asymmetric traveling salesman problem. The study shows the model can be efficient for some certain cases but needs further algorithmic development for variants of the problem.

Regarding the capacity efficiency concern in process industries, an elaborated setup decision within the plan cycle is necessary. In the presence of high associated costs, the duration of a production run for a given setup needs to be long enough so that the balance between setup and inventory holding costs for products involved is ensured. Therefore, products belonging to a certain family are usually produced in campaigns. In glass manufacturing for instance, products that have the same color, which is the main driver of setup, are produced in campaigns. For a specific color, the plan usually contains one or two campaigns in a year, in order to minimize the changeovers Taşkın and Ünal (2009). Hence, we can define campaign planning as the process of determining the timing and the length of such production run decisions.

We refer to Ağralı (2012) to embody the formal definition of co-production, which is producing several different products in a single production run by necessity. Co-production processes exist in various industries including petroleum, semiconductor, glass etc. Main difference of co-production from by-production is that co-products are primary products themselves and that a certain combination of products needs to be produced conforming to the necessities of the process along with intended products. On the other hand, a by-product is not primarily produced itself but rather produced as a result of producing another product. Co-production needs to be dealt with in process industries since it can result in undesired production, which means production and inventory holding costs incurred unintentionally.

There are numerous studies in the literature related to production planning with setup considerations. Copil, Worbelauer, Meyr, and Tempelmeier (2017) gives definitions for General Lot Sizing and Scheduling (GLSP), Capacitated Lot Sizing with Sequence-Dependent-Setup (CLSD), Proportional Lot Sizing and Scheduling (PLSP), Continuous Lot Sizing and Scheduling (CLSP) and Discrete Lot Sizing and Scheduling (DLSP). Authors categorize referenced works with respect to being extension to one of these fundamental models. Another review study is provided by Allahverdi, Ng, Cheng, and Kovalyov (2008), in which they categorize the literature based on shop environment type including single machine, parallel machines and flow shops, batch and non-batch setup indications and sequence dependency. Allahverdi (2015) proposes an updated version of Allahverdi et al. (2008) with a review of around 500 papers. The classification of the reviewed papers is exactly the same and this newer version covers problems involving static, dynamic, deterministic and stochastic environments for different shop types.

Capacitated Lot Sizing Problem (CLP) is the basic production planning problem and is known to be NP-hard Florian, K. Lenstra, and H. G. Rinnooy Kan (1980). In this study, we focus on single machine case, which is essentially a single-level General Lot Sizing Problem (GLSP) with sequence dependent family setup and co-production extensions. Finding a feasible solution for this problem, which is single-level special case of General Lot Sizing Problem for Multiple Production Stages (GLSPMS), is NP-complete Fleischmann and Meyr (1997).

In the existence of pre-defined jobs, heuristic algorithms are frequently used. Herr and Goel (2016) apply Variable Neighbourhood Search (VNS) with six different moves. Jin, Song, and Wu (2009) apply a batch-based Simulated Annealing (SA) algorithm with a neighbourhood definition aimed to increase efficiency in neighbour detection by eliminating non-promising neighbours. Non sequence dependent but family dependent job scheduling without pre-emption is solved with six different heuristics by Uzsoy and Vel¡asquez (2008) while Guo and Tang (2015) apply Scatter Search to solve the problem having sequence dependency existing in the same problem. Bektur and Saraç (2019) aim to minimize total weighted tardiness for scheduling unrelated parallel machine scheduling problem with sequence dependent setup times and machine eligibility restrictions. They propose a simulated annealing (SA) and a tabu search (TS) algorithm. Numerical experiments show TS with long-term memory yields better solutions.

Mathematical formulation based solution modeling is also widely applied. Gören and Tunalı (2015) formulate capacitated lot sizing problem under sequence independent setup as a Mixed Integer Linear Program (MILP) with setup carryover. Fiorotto et al. (2017) state the main contribution of their work as enabling setup crossover between periods without adding binary variables. Toso, Morabito, and Clark (2009) study animal feed compound production, where some products might serve for cleansing as long as they are produced a certain amount, which results in violation of triangular inequality of sequence-dependent setups. They apply a Relax and Fix heuristic, which is shown to be computationally and economically effective compared to current practice in the industry. Araujo and Clark (2013) argue that it is not possible to solve MILP based formulation to optimality in reasonable time and hence they propose a solution procedure encapsulating a combination of Descent Heuristic (DH), Diminishing Neighbourhood Search (DNS) and SA. Setup formulation approach is based on setup carryover in Haase and Kimms (2000), and an efficient and fast sequence enumeration proposed along with a lower bound generation scheme. Ghirardi and Ameiro (2019) study generalization of lot scheduling problem including backordering and setup carryover on unrelated parallel machines. They formulate three different matheuristics inspired by local search, local branching and feasibility pump (FP). Their tests show that their approach outperforms other approaches and two MIP solvers on base formulation. Günther (2014) proposes a change of paradigm in lot sizing and scheduling named block planning concept, which is based on continuous representation of time. The author further argues that since setup and inventory holding costs are hard to determine in most practical cases, timespan minimization is a reasonable objective.

Ağralı (2012) proves that the uncapacitated dynamic lot-sizing problem with co-production can reduce to single item lot sizing problem and consequently Dynamic Programming (DP) is applicable. Öner and Bilgiç (2008) study the effect of uncontrolled co-production on the production schedules and apply common cycle schedule method.

Figueira, Santos, and Almada-Lobo (2013) focus on short-term production planning and scheduling in pulp and paper industry with two stage. Their solution methodology consists of combination of hybrid VNS and Speeds Constraint Heuristic (SCH). Figueira et al. (2015) study the same problem again in pulp and paper industry to the extent of development of a decision support system containing some simplifying assumptions. The proposed system provides satisfactory results in reasonable running times, around 10 to 15 minutes. Furlan, Almada-Lobo, Santos, and Morabito (2015) formulate a MIP model for lot scheduling in pulp and paper industry in integrated mills. They propose a genetic algorithm (GA) to efficiently solve large instances. Toledo, da Silva Arantes, de Oliveira, and Almada-Lobo (2013) study a similar problem in glass container industry. Glass color, which causes a major setup in glass manufacturing environments, is assumed to remain constant in short-term and sequence dependent setups are associated with product changeover. The authors propose multi-population GA and SA as solution approaches.

Lot sizing and scheduling problem has drawn much attention from researchers. Moreover, process industries such as glass manufacturing, steel, pulp and paper, petroleum etc. are also popular due to their specific planning complexities. However, sequence dependent family setups are not thoroughly studied as stated by Allahverdi (2015) with an emphasis on the need for research on single machine environments. Moreover, the effect of the dimension of the attributes that form a setup family for a campaign is not studied to the best of our knowledge. Furthermore, co-production is a phenomenon that has very limited past research. In this paper, we will develop efficient formulation for campaign planning under co-production in single machine case.

The rest of the paper is organized as follows. We first provide a definition of the problem with float glass manufacturing being the specific application domain and discuss planning issues in Section 2. Two mixed integer linear programming formulations are described in Section 3. We propose two mathematical models similar to models proposed by Guimarães et al. (2014) and Lime et al. (2011) in terms of setups between adjacent periods allowed to occur before, after or during period transition. The main difference of our formulations is setups being sequence dependent between families but not the products. Moreover, our formulations model co-production which stems from a natural characteristic of glass production. We present the results of the computational experiments providing insights about both mathematical and business aspects in Section 4. Finally, Section 5 concludes the paper.

2. Problem Definition

Float glass manufacturing is an example of a process industry, where the main driver in planning process is the cost and the effectiveness in capacity usage. Furthermore, float glass manufacturing has some special characteristics making it difficult to obtain high quality plans.

The term float refers to the physical nature of the glass production. Molten solution, consisting of raw materials such as sand, limestone and soda ash, is fed into a tin bath and transforms into its flat form by floating over liquid tin. The primary characteristics of the finished product is determined by raw materials fed into the mixture Taşkın and Ünal (2009). The most important and primary attribute of float glass is its color. Switching from one color to another requires several days, significant amount of time and energy consumption and hence is very costly. In order to compensate the setup cost incurred for the changeover and also for efficiency purposes, each family has a corresponding minimum production duration. Moreover, setups depend on other attributes of glass such as coating. The problem hence contains the phenomenon of sequence-dependent family setups.

Due to the chemical nature of the process, random errors on the glass surface appear during production. Depending on the cutting decisions regarding the size, different size and quality combinations can be produced. Using the historical data reflecting the characteristics of a specific production line, we can determine the percentages up to which a specific combination of size and quality can be manufactured at most. For example, producing high quality glass in big sizes on a specific production line might eventually result in an increase in production of moderate and/or low quality glass in lower sizes. We can define this as partially controllable co-production. For a more detailed explanation on float glass manufacturing fundamentals, we refer to Taşkın and Ünal (2009).

Float glass manufacturing is a process having setup and co-production attributes as discussed above. Furthermore, it is a continuous process and the furnace needs to operate 24/7 until it reaches the end of its lifetime, which can take more than ten years. Moreover, production capacity depends highly on the mix of products allocated to lines since product attributes affect the production rate.

Figure **??** illustrates the main components of the campaign planning problem. Tactical planning in float glass manufacturing is typically executed by the planning specialists implementing a manually pre-determined campaign plan. The tactical plans are generated at a monthly level since the demand forecasts are available on discrete time with monthly availability. However, critical information that drives planning activities such as setup durations and production speed is available in continuous time. Hence, the campaign planning problem needs to incorporate continuous timeline while ensuring the demand responsiveness on discrete time. The main output of the planning is the campaign plan, which we can define as a sequence of families, start and end times of setups and productions. The planning process yields production quantity per product and period based on campaign decisions, which in turn provides demand satisfaction and backlog plan as well as inventory projection.

Figure **??** illustrates an example of a campaign plan for four periods. With the help of this illustration, we can observe synchronization of input and output data, which are available on different time resolutions. For each period, a production amount and demand for a single product from family FM is available. On the other hand, the campaign plan is available on continuous time. For example, a campaign of family FM starts in Period 1 and ends in Period 2. Production quantity within this campaign is associated with Period 1 and Period 2 with respect to time overlapping with each one of them. As a result, the production quantity is disaggregated to discrete time. With the help of the dotted lines, we can also observe the illustration of demand satisfaction schema. For example, the demand of Period 2 is satisfied from productions in Period 1 and Period 2. whereas the demand of Period 3 is partially satisfied from Period 2 and Period 4, which results in backlogging. Moreover, for each period, considering the production quantity and demand satisfaction plan, one can obtain ending inventory projections.

Let us note the main characteristics of the problem as follows:

- Demand forecast per product is available on a discrete time (monthly).
- Input master data, which consists of the parameters of the decision process such as inventory holding or demand backlog cost, production speed per item and setup duration between families, is available on a continuous time.
- Main cost items are inventory holding, demand backlog/unsatisfaction and setup. Production costs are ignored since the problem is on a single machine.
- Setups are costly such that the furnace consumes as much as energy as in production without yielding any glass in order of days in duration. Hence, setups are important in terms of ensuring cost effectiveness of the plan.
- Due to significant setup duration and costs, campaigns are encouraged to have relatively long durations. However, since this will also effect the demand satisfaction plan and backlog is another major expense item, obtaining an optimal campaign plan is crucial.
- Due to the fact that sequence-dependent setup times are expressed in continuous time, the campaign plan needs to be on continuous time.

To elaborate on the last item, we note that our problem differs from aggregate planning. Lot sizing decisions need to be discrete to match the availability of demand forecast. On the other hand, as stated sequencing decisions considering sequence-dependent setup times and production speed is in continuous time. Hence, the synchronization between discrete and continuous information is challenging in terms of formulation. To the best of our knowledge a model that can efficiently incorporate continuous time input data with discrete time data without harming the optimality due to discretization is not present in the literature.

Despite focusing on float glass manufacturing as the specific application domain, we note that our approach for dealing with sequence dependent family setups leading to campaign planning can be generalized to other process industries.

3. Mathematical Models

In this section, we develop two mathematical models for the campaign planning problem described in the previous section. Both models are mainly based on the state decisions of the machine in each time bucket and they mainly differ from each other with respect to the formulation of the state transition over period boundaries. We name the models Pattern Transition Based Model (PTBM) and Family Transition Based model (FTBM) respectively.

In order to clarify the formulations, we first define the concept of pattern in Section 3.1 and then introduce the formulations in Sections 3.2 and 3.3 in the remainder of this Section. In addition, Table 1 illustrates symbols used in both PTBM and FTBM.

3.1. Pattern

3.1.1. Definition

We can define a pattern as an ordered list of families that will be produced consecutively within a period. The concept of pattern is similar to sequence in Guimarães et al. (2014) with the difference that they define sequence by product order but we define patterns by family order.

An important issue to address in pattern definition is that setup times are respected. Each adjacent pair within the pattern needs to be feasible in terms of setup changeover. Let FM, MV and BR be three families available. We can define Pattern 1, a pattern with single family as FM, Pattern 2, a pattern with two families FM-MV, Pattern 3, a pattern with three families BR-MV-FM and Pattern 4, another pattern with three families MV-BR-MV. Figure **??** illustrates these four example patterns. Notice that these represent sequence of the families that the furnace will produce in a period. In addition, the setup from family FM to MV (for Pattern 2), BR to MV, MV to FM (for Pattern 3), MV to BR and BR to MV (for Pattern 4) should be feasible.

Moreover, we distinguish the amount produced at the beginning, in the middle and at the end of a period. As an example, in Pattern 3, family BR corresponds to the beginning, MV to the middle and FM to the end. Note that as in Pattern 4, a family can appear in multiple sequences also. We assume that for patterns having at most two families, the set of families produced in the middle is empty.

In our formulations we will assign a pattern to each period. Consequently it's also important that the setup between the last family of a predecessor pattern and the first family of its successor pattern is also feasible. Setup data is known and hence is given as an input. We can efficiently represent this data as a matrix having families

Table 1. Symbols used in both formulations.

Set	Description
J	Set of products
Q	Set of quality groups
S	Set of size groups
T	Set of time periods
P	Set of campaign patterns
F	Set of product families
O	Set of orders for timing of production in a period (b: beginning, m: middle, e: end)
$P(f)$	Set of patterns containing family f at least once
$F^o(p)$	Set of families appearing in order o in pattern p
$P^o(f)$	Set of patterns containing family f in order o
$J(f)$	Set of products belonging to family f
$\Gamma(f, g)$	Set of product family couples that are infeasible, $f, g \in F$
Parameter	**Description**
D_{jt}	Demand of product j in period t
$I_{j(-1)}$	Beginning inventory of product j
v_j	Production speed of product j, machine-days required for unit production
A_t	Available capacity of the machine in period t in days
$S(j)$	Index of the size group of product j
$Q(j)$	Index of the quality group of product j
R_{fqs}	Maximum production ratio/percentage for quality group q and size group s for family f
MD_f	Minimum production duration for family f in days
NT_{fp}	Number of times family f appears in the middle order of pattern p
MD_{fp}	Minimum production duration for family f in middle order of pattern p, can similarly be expressed as $MD_f NT_{fp}$
ST_p	Setup time needed for family order within pattern p in days
ST_{fg}	Setup time needed for switching from product family f to family g in days
h_j	Inventory holding cost for product j
b_j	Cost of backlogging a demand of product j for a single period
c_{fg}	Setup cost of switching from family f to family g
c_p	Total setup cost of family order within pattern p
Variable	**Description**
I_{jt}	Inventory of product j at the end of period t
S_{jtk}	Satisfied quantity of demand from period t of product j in period k
U_{jt}	Unsatisfied quantity of demand from period t of product j
X_{jt}	Production quantity of product j in period t
δ_{pt}	Binary indicator variable for selection of pattern p in period t
d^o_{ft}	Number of days spent for production of family f in order o in period t

in columns and rows. Each cell in the matrix corresponds to the setup duration/cost between the corresponding couple. Notice that, for infeasible family couples, which can be due to some technical properties, cells can be filled up with a sufficiently large value being larger than maximum number of days in a month.

Let us explain our approach regarding the representation of the setup over period boundaries in more detail with the help of illustrations as in Figure **??**. Case (a) is an example where the setup time spent between families MV and BR crosses over period boundary. The Case (b) represents an example where the setup time is spent at the beginning of successor period. Note that depending on the production quantity and consequently duration decisions, it might well be also spent at the end of the predecessor period as in case (c). Finally, case (d) is an example for no-setup instance as the production within the same family continues. Note that with this approach the model can decide on allocating patterns such that setup is executed during period boundaries, which is not possible with sequence decisions in Guimarães et al. (2014).

3.1.2. Pattern Generation

As explained in Section 3.1.1, a pattern is simply an ordered list of families that we can assign to a period on the production line. We can generate patterns with Algorithm 1 which is not explained in Guimarães et al. (2014) how to obtain the pre-defined sequences.

The algorithm works with the set of families F and the corresponding setup matrix M, which we use as input to a recursive procedure called Extend. At each call to Extend, the procedure evaluates each family f with respect to three criteria: i) f should be different than the last family of the current sequence, ii) by inserting f to the end of the sequence, minimum possible duration of this new sequence should not exceed the duration of a period, iii) if by adding f to the end of the sequence minimum possible duration exceeds the duration of a period, then there should be at least a strictly positive amount of time for producing f in addition to minimum possible duration of the sequence.

We define the minimum possible duration of a sequence as the sum of minimum production duration of appearing families and the setup required for the sequence. Also note that, with criteria iii), we make sure that even if a sequence is not feasible to be executed in a period with respect to its minimum duration, we do not eliminate it since our formulations can handle such a case. We explain this further in Section 3.2.1 and 3.3.1 in detail. Note that, the algorithm generates all possible sequencing combinations so that the mathematical models can allocate sequences to periods to optimize the plan taking setup costs into account.

3.2. Pattern Transition Based Model

Table **??** lists the symbols used in PTBM in addition to common symbols listed in Table 1 along with their brief descriptions. We present the constraints in Section 3.2.1. First, we define the fundamental constraints of GLSP followed by the constraints related to business model, which are tied to specifics of float glass manufacturing. Finally, we present the campaign defining constraints. We define the objective function and give the complete model in Section 3.2.2.

In order to facilitate the understanding of the formulation logic, we present Figure **??** as an illustrative example. We have patterns *FM-MV* and *BR-MV-FM* assigned to periods t and *t+1* respectively, and the relations between periods in terms of variables

Algorithm 1: Generate all patterns p for a given set of families F

```
GeneratePatterns (F, M)
    inputs : Set F of all families and setup matrix M
    output: List of patterns P
    LL ← ∅ (LL is a list)
    return Extend(LL, F, M)

Extend (LL, F, M)
    inputs : A list to be extended with new family insertions, set of families
             and setup matrix
    output: List of patterns P
    P ← ∅
    foreach family f ∈ F do
        if tail(LL) ≠ f and CanAdd(LL, f, M) then
            LL ← LL ∪ f
            P ← P ∪ LL
            P ← P ∪ Extend(LL, F, M)
    return P

CanAdd (LL, f, M)
    inputs : A list and a family f and setup matrix
    output: Indicator whether family f can be inserted to given list
    D ←MinDuration(LL, M)
    if D ≥ length of a period then
        return FALSE;
    S ← M[tail(LL), f]
    D ← D + S
    if D ≥ length of a period then
        return FALSE;
    else
        return TRUE;

MinDuration (LL, M)
    inputs : A list and setup matrix
    output: Minimum possible duration of given ordered family list LL
    D ← 0
    foreach family f ∈ LL do
        D ← D + M[prev(f), f] + MD_f
    return D
```

can be seen on the figure. Moreover, considering pattern BR-MV-FM assigned to period $t + 1$, let us note that family BR is produced in order b at the beginning, MV in m in the middle and FM in e at the end.

3.2.1. Constraints

We permit backlog for demand satisfaction since the demands of products can be spread over the planning horizon whereas the duration and the timing of production campaigns are restricted. Eq. (1) ensures the consistency of demand satisfactions.

$$\sum_{\substack{k \in T \\ k \geq t}} S_{jtk} + U_{jt} = D_{jt} \quad \forall\, j \in J,\ t \in T \tag{1}$$

Eq. (2) is the inventory balance constraint that links production quantity X, ending inventory I and demand satisfaction S variables across time periods.

$$I_{j(t-1)} + X_{jt} - \sum_{\substack{k \in T \\ k \leq t}} S_{jkt} = I_{jt} \quad \forall\, j \in J,\ t \in T \tag{2}$$

Production cannot be interrupted since the furnace needs to be up and running in 24/7 operating mode. Available capacity must hence be fully utilized, which is ensured by Eq. (3). Note that in addition to time spent for production, Eq. (3) incorporates the setup time required due to the pattern selection.

$$\sum_{o \in O} d^{o}_{ft} + \sum_{p \in P} (ST_p \delta_{pt} + F_{pt} + B_{pt}) = A_t \quad \forall\, t \in T \tag{3}$$

We define the auxiliary variables d^o corresponding to the number of days allocated for production of family f at the beginning, in the middle or at the end of a period t. We relate d^o to the production quantity variables X with Eq. (4).

$$\sum_{j \in J} v_j X_{jt} = \sum_{o \in O} d^{o}_{ft} \quad \forall\, f \in F,\ t \in T \tag{4}$$

Due to the physical and the chemical nature of the glass production, random errors are observed on glass surface. Moreover, products can be substituted with respect to their size s and quality q attributes. For example, a glass sheet of size s can be cut into smaller sizes. Similarly, a sheet of quality q can be substituted as an item of lower quality. Furthermore, depending on the characteristics of the production line, production amount of a specific size group s and quality q cannot exceed a certain percentage of the total production quantity within a time period. Consequently, various production compositions are feasible. We denote this phenomenon as partially controllable co-production as explained in Section 2. Eq. (5) ensures that the production quantities in a time period yield a feasible composition within a specific family. The rates R_{fqs} depend on the characteristics of each furnace and are driven from the historical production data. Note that this approach is defined in Taşkın and Ünal (2009).

$$\sum_{\substack{j \in J(f) \\ Q(j) \leq q \\ S(j) \leq s}} X_{jt} \leq \sum_{j \in J(f)} X_{jt} \; R_{fqs} \quad \forall \; f \in F, \; q \in Q, \; s \in S, \; t \in T \tag{5}$$

Our main approach for the campaign planning is based on assigning patterns to time periods. Eq. (6) ensures that a single pattern is assigned to each period.

$$\sum_{p \in P} \delta_{pt} = 1 \quad \forall \; t \in T \tag{6}$$

To ensure the efficiency and the stability of the manufacturing process, a minimum production duration should be ensured for each production run of a product family. Eq. (7) models this requirement, ensuring a lower bound for production duration of families that are produced in the middle of a pattern. Considering the period boundaries, in an optimum solution we can have the minimum duration split into two adjacent periods. In order to enable our formulation take such a decision, we introduce Eq. (8). On the other hand, we need to make sure that we set a proper upper bound on the production duration variables. Eq. (9) ensures that spending time for producing family f in order o is permitted only if a corresponding pattern is assigned in that period.

$$d^m_{ft} \geq MD_{fp}\delta_{pt} \quad \forall \; p \in P, \; f \in F^m(p), \; t \in T \tag{7}$$

$$d^e_{f(t-1)} + d^b_{ft} \geq MD_f \delta_{pt} \quad \forall \; p \in P, \; f \in F^b(p), \; t \in T, \; t \geq 1 \tag{8}$$

$$d^o_{ft} \leq \sum_{p \in P^o(f)} A_t \delta_{pt} \quad \forall \; f \in F, \; o \in O, \; t \in T \tag{9}$$

In order to properly handle setup crossover, we need to relate θ variables with δ variables. This can be formulated as in Eq. 10, which is a non-linear constraint.

$$\theta_{prt} = \delta_{p(t-1)}\delta_{rt} \quad \forall \; p, r \in P, \; t \in T, \; t \geq 1 \tag{10}$$

Note that we can linearize Eq. 10 as in Eqs. (11)–(13). Hence, we do not consider Eq. (10) any further. Moreover, Eqs. (11)–(13) permit relaxation of θ variables as $\theta_{prt} \geq 0$

$$\theta_{prt} \leq \delta_{p(t-1)} \quad \forall \; p, r \in P, \; t \in T, \; t \geq 1 \tag{11}$$

$$\theta_{prt} \leq \delta_{rt} \quad \forall \; p, r \in P, \; t \in T \tag{12}$$

$$\theta_{prt} \geq \delta_{p(t-1)} + \delta_{rt} - 1 \quad \forall \; p, r \in P, \; t \in T, \; t \geq 1 \tag{13}$$

Setup time spent at the beginning and at the end of a period t are managed with Eqs. (14)–(15). Note that these are big-M type constraints with MST_{fg} being the tightest big-M value. When a pattern transition is active through θ variable, setup time for the corresponding family pair is binding for the sum of setup time variables B and F. Otherwise, both upper bound and lower bound become redundant. Notice that it may or may not be the case that the setup time spans period boundaries with our approach.

$$ST_{fg} + MST_{fg}(1 - \theta_{prt}) \geq B_{p(t-1)} + F_{rt} \quad \forall\, p, r \in P, f = f_p^T, g = f_R^H,\ t \in T,\ t \geq 1 \tag{14}$$

$$ST_{fg} - MST_{fg}(1 - \theta_{prt}) \leq B_{p(t-1)} + F_{rt} \quad \forall\, p, r \in P, f = f_p^T, g = f_R^H,\ t \in T,\ t \geq 1 \tag{15}$$

It is also imperative that the variables for setup time at the beginning and at the ending of a period are zero unless the corresponding pattern is selected. Eqs. (16)–(17) ensure this requirement.

$$F_{pt} \leq STS_f \delta_{pt} \quad \forall\, p \in P, f = f_p^H,\ t \in T,\ t \geq 1 \tag{16}$$

$$B_{pt} \leq STP_f \delta_{pt} \quad \forall\, p \in P, f = f_p^T,\ t \in T,\ t \geq 0 \tag{17}$$

It might be the case that, switching from a certain product family f to another g is not possible due to some technical restrictions or business practice. Eq. (18) ensures that the model does not generate such an output.

$$\delta_{p(t-1)} + \delta_{rt} \leq 1 \quad \forall\, p, r \in P, f = f_p^T, g = f_r^H,\ (f, g) \in \Gamma(f, g),\ t \in T,\ t \geq 1 \tag{18}$$

3.2.2. Objective Function

We define the objective function as cost minimization. We assume that production cost for each product j remains constant within the planning horizon. Inventory holding costs for each product is driven from its production cost. Hence, production costs are implicitly included in the model and do not appear in the objective. We sum inventory holding and demand satisfaction costs over products and periods as the first three components. Our approach for demand unsatisfaction is based on the assumption that it is favorable to satisfy a demand, no matter how long the backlog period is, over unsatisfying. To achieve this, the cost associated with unsatisfaction is calculated as $b_j\ (|T| - t + 1)$, which reflects our assumption that demand can be satisfied from an infinite capacity after the planning horizon ends with a corresponding backlog cost associated. In addition, having the coefficient set as $(|T| - t + 1)$ earlier demands will be satisfied more preferably. Moreover, the cost associated to each family setup is significant and we incorporate this cost into the objective function with both pattern selection and pattern transition variables as with last two components. Model 1 in Appendix A represents the complete formulation for PTBM.

PTBM Objective

$$\text{Minimize} \sum_{\substack{j\in J\\ t\in T}} \left[h_j\ I_{jt} + b_j\ (|T| - t + 1)\ U_{jt} + \sum_{\substack{k\in T\\ k\leq t}} (b_j\ (t-k)\ S_{jkt}) \right]$$
$$+ \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ f=f_p^T\\ g=f_r^H\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{prt}$$

3.3. Family Transition Based Model

In PTBM, an auxiliary variable θ_{prt} is introduced for each feasible pattern pair and time period. This approach may be inefficient in cases where there are multiple pattern couples such that the predecessor's last family and the successor's first family are same. This leads us to the main idea in FTBM. The main difference in FTBM is the way we formulate the transition between periods. Instead of introducing an auxiliary variable for each feasible pattern couple, we introduce variables for a distinct set of family pairs corresponding to one or more pattern pair transition.

Table **??** lists the symbols used in FTBM in addition to the common symbols listed in Table 1 along with their brief descriptions. We present the constraints in Section 3.3.1. Figure **??** illustrates the formulation logic and the variable mapping to a possible campaign plan. Notice that the campaign plan is the same as the one illustrated for PTBM in Figure **??**.

3.3.1. Constraints

First, we note that since FTBM differs from PTBM with respect to the formulation of the state transition over period boundaries, some other concepts remain the same. Hence, the corresponding constraints are still valid for FTBM. In particular, requirement and inventory balance constraints with Eqs. (1)–(2), Eq. (4), which relates production duration and quantity variables and Eq. (5) formulating the production composition regarding the size group and the quality are included in FTBM. Similarly, Eq. (6) ensuring assignment of a single pattern in each period and Eqs. (7)–(9) ensuring the minimum duration for producing family f are valid for FTBM.

Resource balance constraints, that are defined with Eq. (3) in Section 3.2.1 need to be modified due to the differences in the definitions of setup related variables F and B. Note that they do not depend on pattern p in FTBM but rather only on period t. Eq. (19) formulates resource balance as follows:

$$\sum_{o\in O} d_{ft}^o + \sum_{p\in P} ST_p \delta_{pt} + F_t + B_t = A_t \quad \forall\ t \in T \tag{19}$$

In order to determine the first and the last family produced in a period we set Eqs. (20)–(21). Notice that with Eq. (6) combined with Eqs. (20)–(21), variables (γ^S, γ^E)

can only have values from $\{0, 1\}$. Hence, we can relax them as $\gamma^S, \gamma^E \geq 0$.

$$\gamma^S_{ft} = \sum_{p \in P^S(f)} \delta_{pt} \quad \forall\, f \in F,\ t \in T \tag{20}$$

$$\gamma^E_{ft} = \sum_{p \in P^E(f)} \delta_{pt} \quad \forall\, f \in F,\ t \in T \tag{21}$$

θ variables indicate whether a changeover is performed from family f to family g at the beginning of period t, and hence are binary by nature. Similar to Eq. (10), θ variables should be equal to 1 if and only if both corresponding γ variables are 1, which is again non-linear. However, similar to Eqs. (11)–(13), Eqs. (22)–(24) allow us to linearize and relax θ as $\theta \geq 0$.

$$\theta_{fgt} \leq \gamma^E_{f(t-1)} \quad \forall\, f, g \in F,\ t \in T,\ t \geq 1 \tag{22}$$

$$\theta_{fgt} \leq \gamma^S_{gt} \quad \forall\, f, g \in F,\ t \in T \tag{23}$$

$$\theta_{fgt} \geq \gamma^E_{f(t-1)} + \gamma^S_{gt} - 1 \quad \forall\, f, g \in F,\ t \in T,\ t \geq 1 \tag{24}$$

Eq. (25) ensures that necessary setup time for color transition is spent.

$$n^P_{fgt} + n^S_{fgt} = ST_{fg}\theta_{fgt} \quad \forall\, f, g \in F,\ (f, g) \notin \Gamma(f, g),\ t \in T \tag{25}$$

We relate setup time variables for families (n^S, n^E) to period based variables (F, B) with Eqs. (26)–(27).

$$F_t = \sum_{(f,g) \notin \Gamma(f,g)} n^S_{fgt} \quad \forall\, f, g \in F,\ t \in T \tag{26}$$

$$B_t = \sum_{(f,g) \notin \Gamma(f,g)} n^P_{fg(t+1)} \quad \forall\, f, g \in F,\ t \in T \tag{27}$$

Eq. (28) ensures that no infeasible family transition is permitted. Note that this is the counterpart of Eq. (18).

$$\gamma^E_{f(t-1)} + \gamma^S_{gt} \leq 1 \quad \forall\, f, g \in F,\ (f, g) \in \Gamma(f, g),\ t \in T,\ t \geq 1 \tag{28}$$

3.3.2. Objective Function

The objective function is the same as PTBM. Model 2 in Appendix B represents the complete formulation for FTBM.

3.4. Comparison of Pattern Based and Family Based Formulations

As explained in detail in Sections 3.2 and 3.3, formulations differ from each other with respect to the formulation of the state transition over period boundaries. In PTBM, there is a θ variable for each pair of patterns whereas in FTBM θ variables are mapped to each pair of families. The FTBM associates state decision variables δ to setup duration through a convex hull reformulation with Eqs. (20), (21) and (25). Hence, we argue that FTBM is tighter than PTBM with the following proposition.

Proposition 1. Let S^{FTBM} and S^{PTBM} be the feasible regions of linear programming relaxations of FTBM and PTBM respectively. Then, $S^{FTBM} \subset S^{PTBm}$.

Proof. Let I be the set of family pairs (f', g') such that $\theta_{f'g't} > 0$ in a feasible solution to PTBMV. Then summing Eq. (25) over $(f', g') \in I$, we obtain

$$\sum_{(f',g')\in I} n^P_{f'g't} + \sum_{(f',g')\in I} n^P_{f'g't} = \sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \tag{29}$$

Note that, the first term is equal to F_{t+1} and the second term is equal to B_t on the left hand side of the equation. Moreover, from Eqs. (22)–(24), we obtain following inequalities respectively by again summing over $(f', g') \in I$.

$$\sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \leq \sum_{(f',g')\in I} ST_{f'g'}\gamma^E_{f't} \tag{30}$$

$$\sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \leq \sum_{(f',g')\in I} ST_{f'g'}\gamma^S_{g'(t+1)} \tag{31}$$

$$\sum_{(f',g')\in I} ST_{f'g'}\theta_{f'g't} \geq \sum_{(f',g')\in I} (\gamma^E_{f't} + \gamma^S_{g'(t+1)}) + |I| \tag{32}$$

Left hand side of all these three inequalities can hence be replaced by $F_{t+1}+B_t$. On the other hand, when we sum Eqs. (16) and (17) followed by another sum over $(p', r') \in J$ where p' and r' correspond to patterns having f' as ending family and g' as starting family respectively, we obtain

$$\sum_{(p',r')\in J} (B_{p't} + F_{r'(t+1)}) \leq \sum_{(p',r')\in J} (STP_{f'}\delta_{p't} + STS_{g'}\delta_{r'(t+1)}) \tag{33}$$

which also has the left hand side equal to $F_{t+1}+B_t$. Summing Eq. (14) over $(p', r') \in J$ gives

$$\sum_{(p',r')\in J} ST_{f'g'} - \sum_{(p',r')\in J} MST_{f'g'} + \sum_{(p',r')\in J} \theta_{p'r't} \leq \sum_{(p',r')\in J} (B_{p't} + F_{r'(t+1)}) \tag{34}$$

Note that right hand side of the inequality (34) is also equal to $F_{t+1} + B_t$. Then from Eq. (30) and Eq. (31), we obtain following inequalities which are always true by definition of $STP_{f'}$ and $STS_{g'}$ with respect to $ST_{f'g'}$

$$\sum_{(f',g')\in I} ST_{f'g'}\gamma^{E}_{f't} \leq \sum_{(p',r')\in J} (STP_{f'}\delta_{p't} + STS_{g'}\delta_{r't}) \tag{35}$$

$$\sum_{(f',g')\in I} ST_{f'g'}\gamma^{S}_{g'(t+1)} \leq \sum_{(p',r')\in J} (STP_{f'}\delta_{p't} + STS_{g'}\delta_{r't}) \tag{36}$$

Finally from Eq. (34) we obtain

$$\sum_{(p',r')\in J} ST_{f'g'} - \sum_{(p',r')\in J} MST_{f'g'} + \sum_{(p',r')\in J} \theta_{p'r'(t+1)} \leq \sum_{(f',g')\in I} (\gamma^{E}_{f't} + \gamma^{S}_{g'(t+1)}) - |I| \tag{37}$$

The first to components of the left hand side is negative by definition of $ST_{f'g'}$ and $MST_{f'g'}$. The third component is further explored from Eq. (13) by summing over $(p', r') \in J$

$$\sum_{(p',r')\in J} \delta_{p't} + \sum_{(p',r')\in J} \delta_{r'(t+1)} - |J| \leq \sum_{(p',r')\in J} \theta_{p'r't} \tag{38}$$

Since, $\sum_{(p',r')\in J} \delta_{p't} = \sum_{(f',g')\in I} \gamma^{E}_{f't}$, $\sum_{(p',r')\in J} \delta_{r'(t+1)} = \sum_{(f',g')\in I} \gamma^{S}_{g'(t+1}$ and $|J| \geq |I|$, then (37) is also always true. Hence, for each fractional solution to S^{FTBM}, one can find a corresponding solution in S^{PTBM}.

On the other hand, let p^{FM1} and p^{FM2} be two patterns ending with family FM and allocated have corresponding δ variables equal to 0.5 and 0.5 in period t respectively in a feasible solution to PTBM. Similarly, let r^{FM3} and r^{MV4} be two patterns starting with families FM and MV respectively with corresponding δ variables equal to 0.4 and 0.6 in period $t+1$. Following Eqs. (11)–(13) variable $\theta_{p^{(FM3)(t+1)}r^{(MV4)(t+1)}} \geq 0 \geq (0.5+0.4-1)$. Then Eq. (14) and Eq. (15), will become redundant since θ can take value of zero. However, in FTBM, the corresponding θ variable, namely $\theta_{(FM3)(MV4)(t+1)}$, has a lower bound of 0.6 from Eq. (24). This triggers Eq. (25) such that the left hand side has to equal $ST_{(FM3)(MV4)} * 0.6$ which might results in different setup duration for PTBM and FTBM. Hence there exists a fractional solution of PTBM, which is not a feasible solution of FTBM.

□

3.5. Pattern Set Preprocessing

Notice that both formulations contain binary variables corresponding to patterns, (δ variables). Moreover, PTBM also contains auxiliary variables θ, which can increase up to the number of cartesian product of the number of patterns and the number of periods. Hence, it is crucial to reduce the number of patterns while ensuring optimality of the solution.

We observe that multiple patterns generated with the Algorithm 1 can result in the production of the same set of families for a given beginning and ending family pair. Let us elaborate with illustrative examples. Let f_1, f_2, f_3 and f_4 be a set of families and p_1 and p_2 be a couple of generated patterns containing these families. Let the sequence of p_1 be f_1 - f_2 - f_3 - f_4 and the sequence of p_2 be f_1 - f_3 - f_2 - f_4. If setup costs for pattern p_1 is less than that of p_2, then an optimal solution will favor p_1 to p_2 since both patterns have common starting and ending families, and the same set of families produced in only different sequences.

A similar redundancy appears in cases where a pattern contains as sub-sequence, the replication of a specific number of times of another pattern. Let f_1 and f_2 be a couple of families and p_1 and p_2 be a couple of generated patterns. Let the sequence of p_1 be f_1 - f_2 and the sequence of p_2 be f_1 - f_2 - f_1 - f_2. Notice that p_1 is a 'shrunk' version of p_2, and that since p_2 yields more setup time and setup cost having twice the setup f_1 to f_2 and one f_2 to f_1 whereas p_1 yields more useful production time, p_2 can be removed from the list of patterns, thus reducing the number of binary variables in both formulations.

Algorithm 2 groups all patterns with respect to their canonical representation and keeps the one from each group having the least associated cost. Since we need to keep all the patterns enabling all possible transitions over period boundaries, information about the beginning and the ending families should not be lost, which we ensure by sub procedure GetCanonicalRepresentation in Algorithm 2.

Algorithm 2: Pattern preprocessing

SimplifyPatterns (P)
- **inputs :** Set of patterns P
- **output:** List of simplified patterns P'
- $P' \leftarrow \emptyset$
- $G \leftarrow$ Group all patterns in P in by GetCanonicalRepresentation(p)
- **foreach** pattern group $g \in G$ **do**
  - $P' \leftarrow P' \cup \text{argmin}_p = \{c_p\}$
- **return** P';

GetCanonicalRepresentation (p)
- **inputs :** A pattern p
- **output:** A string value
- $f \leftarrow$ beginning family of pattern p
- $g \leftarrow$ ending family of pattern p
- $M \leftarrow$ ordered distinct list of families in pattern p
- $s \leftarrow \textit{concatenate}(f, f' \in M, g)$
- **return** s;

3.6. Formulation Variations

In both formulations PTBM and FTBM, infeasible changeovers between families over period boundaries are prohibited explicitly with Eq. (18) and Eq. (28) in PTBM and FTBM, respectively. From another point of view, this is equivalent to the condition that over period boundaries, only feasible family setups should be allowed. Hence, this can be achieved with Eq. (39) for PTBM:

$$\sum_{\substack{p,r \in P \\ f=f_p^T \\ g=f_r^H \\ (f,g)\notin \Gamma(f,g)}} \theta_{prt} = 1 \quad \forall\, t \in T,\ t \geq 1 \tag{39}$$

and with Eq. (40) for FTBM:

$$\sum_{\substack{f,g \in F \\ (f,g)\notin \Gamma(f,g)}} \theta_{fgt} = 1 \quad \forall\, t \in T,\ t \geq 1 \tag{40}$$

Notice that Eqs. (39)–(40) may decrease the number of constraints significantly depending on the number of patterns and families. In PTBM and FTBM, Eq. (18) and Eq. (28) are written explicitly for each period transition and for each pair of infeasible pattern and family pairs respectively. On the other hand, in variant models PTBMV and FTBMV, a single equation exists as Eq. (39) and Eq. (40) for each period transition. Model 3 in Appendix C and Model 4 in Appendix D represent the complete formulation for PTBMV and FTBMV respectively.

We argue that the variant formulations are tighter than primary formulations. The following proposition shows that PTBMV is tighter than PTBM.

Proposition 2. Let S^{PTB} and S^{PTBV} be the feasible regions of linear programming relaxations of PTBM and PTBMV respectively. Then, $S^{PTBV} \subset S^{PTB}$.

Proof. Let I be the set of pattern pairs (p', r') such that $\theta_{p'r'(t+1)} > 0$ in a feasible solution to PTBMV. Then, for each (p', r') we have

$$\delta_{p't} \geq \theta_{p'r'(t+1)}$$
$$\delta_{r'(t+1)} \geq \theta_{p'r'(t+1)}$$

from Eqs. (11)–(12) and since $\sum_{(p',r' \in I)} \theta_{p'r'(t+1)} = 1$ by Eq. (39), then we have

$$\sum_{(p',r')\in I} \delta_{p't} = \sum_{(p',r')\in I} \delta_{r'(t+1)} = 1$$

Hence,

$$\sum_{(p'',r'')\notin I} \delta_{p''t} = \sum_{(p'',r'')\notin I} \delta_{r''(t+1)} = 0$$

Note that such pattern couples include both feasible and infeasible pattern pairs and such feasible pairs Eq. (18) is not relevant. Moreover, for pairs $(p', r') \in I$ such that (p', r') setup is infeasible, since $\sum_{(p',r' \in I)} \theta_{p'r'(t+1)} = 1$ by assumption, we have $\delta_{p't} + \delta_{r'(t+1)} \leq 1$. Hence, each fractional solution of PTBMV is also feasible with respect to PTBM.

On the other hand, let f_1, f_2, f_3, f_4, f_5 and f_6 be families with no feasible transition between any couple except within same family. Let us note patterns including single families also as f_1, f_2 etc. Let δ values in a solution of PTBM be $\delta_{f_1t} = 0.4$, $\delta_{f_2t} = 0.5$, $\delta_{f_3t} = 0.1$, $\delta_{f_1(t+1)} = 0.4$, $\delta_{f_4(t+1)} = 0.5$ and $\delta_{f_5(t+1)} = 0.1$. Note that since there is no feasible transition between any couples other than f_1t to $f_1(t+1)$, for any combination Eq. (18) is satisfied. However, since the only feasible transition (f_1t to $f_1(t+1)$) implies that $\theta_{f_1f_1(t+1)} \leq 0.4$ then Eq. (39) is violated and hence there exists a fractional solution of PTBM, which is not a feasible of PTBMV. □

Note that by similar approach, we can also prove the following proposition.

Proposition 3. Let S^{FTB} and S^{FTBV} be the feasible regions of linear programming relaxations of FTBM and FTBMV respectively. Then, $S^{FTBV} \subset S^{FTB}$.

4. Computational Experiments

In this section, we give details about numerical results from running the proposed formulations. We implemented formulations with C# language of the .NET Framework and used commercial solvers CPLEX (12.8) and Gurobi (8.1) for computational experiments. We executed all experiments on a PC with Intel Core i7-8750H CPU 2.20 GHz and 16 GB RAM.

4.1. Data Set

The data used in the numerical experiments is based on real life data provided by a major float glass manufacturer in Turkey. Hence, the data is realistic in terms of production, setup and cost perspective. The data set contains 153 unique products of different color, size, quality, coating, thickness and packaging type attributes.

Color is the primary attribute affecting the duration and the cost of a changeover. Hence, we include color in the family structure. In addition, coating is another attribute that requires setup between products of the same color. Hence, color and coating will be considered as attributes that form a family. Moreover, in order to investigate the significance of adding or removing an attribute in family structure, we will work with three different structures. We can enumerate them as follows:

- **Color:** The simplest structure. Only color forms a family, and all coating types are considered in the same family
- **Color & C/NC:** In addition to color, coating is incorporated into family structure in a binary form: **C** = Coated, **NC** = Not Coated
- **Color & Coating:** Both color and coating attributes are considered in families.

There are three colors, namely fume (FM), bronze (BR) and blue (MV), and three coating types, namely without coating (Z), pyrolitic (P) and titanium (T). For each different family structure explained above, we have 3, 6 and 8 families respectively aggregating 153 unique products.

4.2. Formulation Analysis

In order to compare the performances of the four models proposed with the data set explained in Section 4.1 we designed a set of run instances. We can list the main attributes for the instances as follows:

- **Number of Periods:** 4, 6 and 8 periods
- **Formulation:** PTBM, FTBM, PTBMV and FTMBV
- **Family Structure:** Color, Color & C/NC and Color & Coating

Table **??** shows values for the number of patterns, the number of continuous and binary variables and the number of constraints.

We note that the number of patterns depends on the family structure. Similarly, the number of variables in each formulation depends on the formulation and the number of periods in addition to the family structure. The number of binary variables, on the other hand, depends on the number of patterns and periods (δ_{pt}).

We can observe that the number of variables and constraints increase in all formulations with respect to the family structure. However, the increase rate is much higher in Pattern Transition Based (PTB) models (PTBM and PTBMV). The number of variables and constraints are expected to be much higher in Pattern Based models than Family Based models, which is the case for family structure Color & Coating and eight periods instance. However, we observe that when coating is not selected as a family-forming attribute the results are somewhat surprising. For instance when we compare PTBM and PTBMV in Color family structure and four periods instance, we see that the number of variables remains constant and that the number of constraints increases in Variant version. We observe that the reason behind such a case is the following: once the coating attribute is removed from the family structure, the family sequence setup restrictions disappear as it is possible to change colors in any sequence (with different setup durations). Hence, PTBM contains no constraints (18) and its variant version contains constraints (39). A similar situation is also observed in Family Transition Based models.

In order to analyse the efficiency of the pattern preprocessing, let us share the details about the number of patterns per family structure. In Color structure, Algorithm 1 generates 42 patterns and Algorithm 2 eliminates 18 of them resulting in 43% decrease. Similarly, respective numbers for Color & C/NC are 165, 115 and 30%, and for Color & Coating are 171, 135 and 21%. Note that the number of patterns decreases by 31% on average, which is important in terms of performance since the number of binary variables depends on the number of patterns.

Regarding the solution performance, let us first observe the Linear Programming (LP) relaxation objective values of the formulations. Table **??** shows the objective values of LP relaxation of the proposed formulations. We observe that Family Transition Based (FTB) formulations generate significantly tighter LP relaxation objectives compared to PTB models.

Moreover, for both PTB and FTB models, variant formulations produce higher LP relaxation objectives in all run instances compared to their respective original formulations, which is in alignment with Propositions 2 and 3.

We implemented a general purpose optimization layer in our implementation that enables us to use both CPLEX and Gurobi solvers. Table **??** illustrates Central Processing Unit (CPU) time in seconds, relative MILP gap and incumbent solution objective value per solver and per run instance. All instances are solved with a time limit of 8 hours (28800 seconds).

We note that for each family structure and number of period combination, at least one of the formulations was able to find an optimal solution. Moreover, some of the solution runs, such as PTBM in eight periods and Color & C/NC family structure, were able to find an optimal objective value but were not able to prove the optimality. Regarding the formulations, we note that in all instances FTB models outperform

PTB models. We investigate the performances of CPLEX for the sake of simplicity in summary. Considering FTBM and its variant, FTBMV, the variant performs better than the original formulation regarding computational time except a single instance, 6 periods and Color as family structure. We observe that FTBM finds an optimal solution in the root node, whereas FTBMV also finds an optimal solution at the root node but couldn't prove the optimality without exploring 383 nodes resulting in 1 second of difference.

On the other hand, PTBMV consistently performs worse than PTBM regarding computational time. To further investigate, we checked the solver logs and observed that root node solution time is consistently taking much longer in variant formulations. For example, in 8 periods and Color & Coating family structure, root node processing takes 1708 seconds in PTBMV while 201 seconds in PTBM. A potential reason for such a difference is related to PTBM having many more constraints than its variant except for one case explained above. PTBM has more and sparser constraints as in Eq. (18) whereas the variant PTBMV has less and denser set of constraints with Eq. (39). Considering the solvers' working mechanism of working with sparse algebra, we can explain the difference in computational performance.

A solver outperforms the other if it obtains a solution with lower optimality gap. If both obtain an optimal solution within the time limit, then whichever proves optimality earlier is noted as the winner. Let us summarize the number of "wins" per solver as follows:

- **4 Periods:** Gurobi wins 5 times while CPLEX wins remaining 7
- **6 Periods:** Gurobi wins 8 times while CPLEX wins remaining 4
- **8 Periods:** Gurobi wins 7 times while CPLEX wins remaining 5

We observe that, in more cases Gurobi outperforms CPLEX and especially in FTB models, Gurobi obtains provably optimal solutions faster than CPLEX. As the problem instance becomes more complex, Gurobi tends to outperform CPLEX. However, in the most complex instance, which is 8 periods with Color & Coating family structure, CPLEX finds a provably optimal solution in 8873 seconds whereas Gurobi is able to solve the instance in 17534, which is almost twice the time. Moreover, in smaller instances, those with 4 periods, CPLEX outperforms Gurobi. Hence, we can conclude that there is no clear superiority of one solver to the other. Nevertheless, we will use FTBMV and Gurobi for further experiments, being the combination most frequently performing better than the others.

4.3. Business Insights

Analysis presented in Section 4.2 discusses the problem and formulations in detail from a mathematical point of view. Set of experiments up to now measure the performance of different formulations proposed. However, since the problem has some unique challenges it is also valuable to elaborate the analysis on some business insights perspective. Our main goal is to observe the characteristics of the generated campaign plans with respect to different business scenarios.

As stated in Section 4.2, we will use FTBMV in a set of experiments for testing further scenarios. Our main goal in the next is to analyse the changes in number campaigns and average duration per campaign overall. Total setup duration driven by campaign plan is also another metric to be observed. We expect to gather further insights from other business indicators such as average total ending inventory per month and total backlogged or unsatisfied demand.

Costs associated with inventory holding and demand backlog/unsatisfaction are subject to some business requirements and assumptions. Moreover, setup costs have a crucial role in campaign decisions being a significant expense item and having physical counterpart. Since all these costs mentioned are in the objective function to be minimized, we decided to design a new set of run instances that will enable us to observe the marginal effect of each cost component to the resulting campaign plan.

We adapt an approach similar to Fiorotto et al. (2017) in order to evaluate effects of cost components. We first assume a baseline run instance with family structure Color & Coating and 8 periods. Then, for each cost component, we solve the campaign planning problem having corresponding coefficients multiplied with 0.1, 0.2, 0.5, 2, 5 and 10. In each case, we observe the changes in various measures such as the number of campaigns, total setup duration and average ending inventory. Figure **??** shows an optimal campaign plan for our baseline instance.

We first analyze the effect of setup costs. Figure **??** shows some metrics that will help us interpret the behavior of the outcoming campaign plans compared to the expectations. In each one of the charts, term Mx corresponds to a run instance where M stands for the multiplier used. Note that 1x is the Baseline instance. With increasing the setup costs, we expect to have fewer setups, which is validated with Figure **??** (a). Considering the average campaign duration, although the trend is increasing as expected with fewer campaigns per family, in 5x instance we observe the measure against our expectation. The difference is driven by family BRP, which in 5x instance has a single campaign of 5.06 days whereas in 2x instance there are two BRP campaigns with average duration of 17.72 days. We further observe that the ending inventory at the end of the planning horizon for family BRP is 14247 in 2x instance whereas this figure is only 331 in 5x instance. The inventory to be held shifted to FMZ family in 5x instance, which did not have any ending inventory in 2x instance. We anticipate that with increased setup costs, model could decrease the overall costs with such a combination regarding inventory holding costs. With fewer number of campaigns, the total setup duration spent is expected to be less as well, which can be observed in Figure **??** (c). With longer campaign durations higher amount of inventory is expected to be carried, which we validate with Figure **??** (b) and we observe a similar behaviour for total backlogged and unsatisfied demand quantity.

Figure **??** shows the effects of the changes in backlog coefficients. With increasing backlog costs, in order to decrease the cost due to backlogging, we expect to have more campaigns in shorter duration. Figures **??** (a) and (b) illustrate the increase in both number of campaigns and total setup duration. However, average campaign duration fluctuates even though the trend is downwards. Clearly, with increasing backlog cost, models tend to have less and less backlogged demand and average ending inventory is also decreasing since there is a larger number of shorter campaigns.

Inventory holding cost is the expense item with the least effect on resulting campaign plans as observed in Figure **??**. With increasing inventory cost, we expect to have more campaigns having shorter duration to avoid holding more inventory longer. This is observed with Figure **??** (a). Also with more campaigns, we observe eventually longer total setup duration. The average ending inventory tends to decrease but only a significant change in inventory costs can drive this.

5. Conclusion

In this paper we studied the single machine campaign planning problem under sequence dependent family setups and co-production in the process industry. We proposed two formulations PTBM and FTBM and variations being stronger in terms of LP relaxation. With the runs using a realistic dataset, we are able to obtain an optimal solution for each problem instance within a given time limit. Regarding the different formulations, FTBM and its variation are shown to outperform PTB models. Moreover, FTBM and FTBMV are both more compact in terms of the number of variables and constraints. The sensitivity of some measures related to the business insights are also provided showing expected behavior in most cases. As a feature research direction, the problem can be studied in multiple machine environments with alternative selection considering production costs. Moreover, we can further extend the research by including multiple facilities and multiple BOM levels.

Acknowledgement(s)

Z. Caner Taşkın's research was partially supported by TÜBİTAK Grant No 116M555.

6. References

References

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

Allahverdi, A., Ng, C. T. D., Cheng, T. C. E., & Kovalyov, M. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, *18*, 985–132.

Almada-Lobo, B., Oliveira, J. F., & Carravilla, M. A. (2008). Production planning and scheduling in the glass container industry: A vns approach. *International Journal of Production Economics*, *114*, 363–375.

Araujo, S. A., & Clark, A. (2013). A priori reformulations for joint rolling-horizon scheduling of materials processing and lot-sizing problem. *Computers and Industrial Engineering*, *65*, 577–585.

Ağralı, S. (2012). A dynamic uncapacitated lot-sizing problem with co-production. *Optimization Letters*, *6*, 1051–1061.

Bektur, G., & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers and Operations Research*, *103*, 46–63.

Clark, A., Morabito, R., & Toso, E. A. (2010). Production setup-sequencing and lot-sizing at an animal nutrition plant through atsp subtour elimination and patching. *Journal of Scheduling*, *13*, 111–121.

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

Figueira, G., Amorim, P., Guimarães, L., Amorim-Lopes, M., Neves-Moreira, F., & Almada-Lobo, B. (2015). A decision support system for the operational production planning and scheduling of an integrated pulp and paper mill. *Computers and Chemical Engineering*, *77*, 85–104.

Figueira, G., Santos, M. O., & Almada-Lobo, B. (2013). A hybrid vns approach for the short-term production planning and scheduling: A case study in the pulp and paper industry. *Computers and Operations Research*, *40*, 1804–1818.

Fiorotto, D. J., Jans, R., & de Araujo, S. A. (2017). An analysis of formulations for the capacitated lot sizing problem with setup crossover. *Computers and Industrial Engineering*, *106*, 338–350.

Fleischmann, B., & Meyr, H. (1997). The general lotsizing and scheduling problem. *OR Spectrum*, *19*, 11–21.

Florian, M., K. Lenstra, J., & H. G. Rinnooy Kan, A. (1980). Deterministic production planning: Algorithms and complexity. In *Management science* (p. 669-679).

Furlan, M., Almada-Lobo, B., Santos, M., & Morabito, R. (2015). Unequal individual genetic algorithm with intelligent diversification for the lot-scheduling problem in integrated mills using multiple-paper machines. *Computers and Operations Research*, *59*, 33–50.

Ghirardi, M., & Ameiro, A. (2019). Matheuristics for the lot sizing problem with back-ordering, setup carryovers, and non-identical machines. *Computers and Industrial Engineering*, *127*, 822–831.

Gören, H. G., & Tunalı, S. (2015). Solving the capacitated lot sizing problem with setup carryover using a new sequential hybrid approach. *Applied Intelligence*, *42*, 805–816.

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

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

Guo, Q., & Tang, L. (2015). An improved scatter search algorithm for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times. *Applied Soft Computing*, *29*, 184–195.

Haase, K., & Kimms, A. (2000). Lot sizing and scheduling with sequence-dependent setup costs and times and effcient rescheduling opportunities. *International Journal of Production Economics*, *66*, 159–169.

Herr, O., & Goel, A. (2016). Minimising total tardiness for a single machine scheduling problem with family setups and resource constraints. *European Journal of Operational Research*, *248*, 123–135.

Jin, F., Song, S., & Wu, C. (2009). A simulated annealing algorithm for single machine scheduling problems with family setups. *Computers and Operations Research*, *36*, 2133–2138.

Lime, R., Grossmann, I., & Jiao. (2011). Long-term scheduling of a single-unit multi-product continuous process to manufacture high performance glass. *Computers and Checmial Engineering*, *35*, 554–574.

Miegeville, N. (2005). *Supply chain optimization in the process industry. methods and and case-study of the glass industry.* (Unpublished doctoral dissertation). Ecole Centrale, Paris, Paris.

Öner, S., & Bilgiç, T. (2008). Economic lot scheduling with uncontrolled co-production. *European Journal of Operational Research*, *188*, 793–810.

Stefansdottir, B., Grunow, M., & Akkerman, R. (2017). Classifying and modeling setups and cleanings in lot sizing and scheduling. *European Journal of Operational Research*, *261*, 849–865.

Taşkın, Z. C., & Ünal, A. T. (2009). Tactical level planning in float glass manufacturing with co-production, random yields and substitutable products. *European Journal of Operational Research*, *199*(1), 252–261.

Toledo, C. F. M., da Silva Arantes, M., de Oliveira, R. R. R., & Almada-Lobo, B. (2013). Glass container production scheduling through hybrid multi-population based evolutionary algorithm. *Applied Soft Computing*, *13*, 1352–1364.

Toledo, C. F. M., da Silva Arantes, M., Hossomi, M. Y. B., & Almada-Lobo, B. (2016). Mathematical programming-based approaches for multi-facility glass container production planning. *Computers and Operations Research*, *74*, 92–107.

Toso, E. A., Morabito, R., & Clark, A. (2009). Lot sizing and sequencing optimisation at an animal-feed plant. *Computers and Industrial Engineering*, *57*, 813–821.

Uzsoy, R., & Velíasquez, J. D. (2008). Heuristics for minimizing maximum lateness on a single machine with family-dependent set-up times. *Computers and Operations Research*, *35*, 2018–2033.

Wittrock, R. (1990). Scheduling parallel machines with major and minor setup times. *The International Journal of Flexible Manufacturing Systems*, *2*, 329–341.

Appendix A. Pattern Transition Based Model

Model 1. *Pattern Transition Based Model (PTBM)*

$$\text{Minimize} \sum_{\substack{j\in J\\ t\in T}} \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k\in T\\ k\leq t}} (b_j \; (t-k) \; S_{jkt}) \right] + \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ f=f_p^T\\ g=f_r^H\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{prt}$$

$$\begin{aligned}
\text{subject to } & (1)\text{–}(9)\\
& (11)\text{–}(18)\\
& I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall (j \in J, t)\\
& S_{jtk} \geq 0 \quad \forall (j \in J, t, k \geq t)\\
& \delta_{pt} \in \{0, 1\} \quad \forall (p, t)\\
& F_{pt}, B_{pt}, \theta_{pt} \geq 0 \quad \forall (p, t)
\end{aligned}$$

Appendix B. Family Transition Based Model

Model 2. *Family Transition Based Model (FTBM)*

$$
\begin{aligned}
\text{Minimize} \; & \sum_{\substack{j\in J\\ t\in T}} \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k\in T\\ k\leq t}} (b_j \; (t-k) \; S_{jkt}) \right] \\
& + \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{fgt} \\
\text{subject to} \; & (1)\text{–}(2) \\
& (4)\text{–}(9) \\
& (19)\text{–}(28) \\
& I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall (j\in J, c, t) \\
& S_{jtk} \geq 0 \quad \forall (j \in J, t, k \geq t) \\
& \delta_{pt} \in \{0,1\} \quad \forall (p,t) \\
& 0 \leq \theta_{fgt} \leq 1 \quad \forall (f,g,t) \\
& \gamma^S_{ft}, \gamma^E_{ft} \geq 0 \quad \forall (f,t) \\
& F_t, B_t \geq 0 \quad \forall (t) \\
& n^P_{fgt}, n^S_{fgt} \geq 0 \quad \forall (f,g,t)
\end{aligned}
$$

Appendix C. Pattern Transition Based Model Variant

Model 3. *Pattern Transition Based Model Variant (PTBMV)*

$$
\begin{aligned}
\text{Minimize} \; & \sum_{\substack{j\in J\\ t\in T}} \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k\in T\\ k\leq t}} (b_j \; (t-k) \; S_{jkt}) \right] \\
& + \sum_{p\in P} c_p \delta_p + \sum_{\substack{t\in T\\ f=f^T_p\\ g=f^H_r\\ (f,g)\notin\Gamma(f,g)}} c_{fg}\theta_{prt} \\
\text{subject to} \; & (1)\text{–}(9) \\
& (11)\text{–}(17) \\
& (39) \\
& I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall (j\in J, t) \\
& S_{jtk} \geq 0 \quad \forall (j \in J, t, k \geq t) \\
& \delta_{pt} \in \{0,1\} \quad \forall (p,t) \\
& F_{pt}, B_{pt}, \theta_{pt} \geq 0 \quad \forall (p,t)
\end{aligned}
$$

Appendix D. Family Transition Based Model Variant

Model 4. *Family Transition Based Model Variant (FTBMV)*

$$
\begin{aligned}
\text{Minimize } & \sum_{\substack{j \in J \\ t \in T}} \left[h_j \; I_{jt} + b_j \; (|T| - t + 1) \; U_{jt} + \sum_{\substack{k \in T \\ k \leq t}} (b_j \; (t-k) \; S_{jkt}) \right] \\
& + \sum_{p \in P} c_p \delta_p + \sum_{\substack{t \in T \\ (f,g) \notin \Gamma(f,g)}} c_{fg} \theta_{fgt} \\
\text{subject to } & (1)\text{–}(2) \\
& (4)\text{–}(9) \\
& (19)\text{–}(27) \\
& (40) \\
& I_{jt}, X_{jt}, U_{jt} \geq 0 \quad \forall (j \in J, t) \\
& S_{jtk} \geq 0 \quad \forall (j \in J, t, k \geq t) \\
& \delta_{pt} \in \{0, 1\} \quad \forall (p, t) \\
& 0 \leq \theta_{fgt} \leq 1, n^P_{fgt}, n^S_{fgt} \geq 0 \quad \forall (f, g, t) \\
& \gamma^S_{ft}, \gamma^E_{ft} \geq 0 \quad \forall (f, t) \\
& F_t, B_t \geq 0 \quad \forall (t)
\end{aligned}
$$