Computers and Chemical Engineering 42 (2012) 206–216

Contents lists available at SciVerse ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Efficient mathematical frameworks for detailed production scheduling in food processing industries

Georgios M. Kopanos[a], Luis Puigjaner[a], Michael C. Georgiadis[b,*]

[a] Department of Chemical Engineering, Universitat Politècnica de Catalunya, ETSEIB, Av. Diagonal 647, 08028 Barcelona, Spain
[b] Department of Chemical Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 10 September 2011
Received in revised form
12 December 2011
Accepted 13 December 2011
Available online 29 December 2011

Keywords:
Production scheduling
Mixed integer programming
Food industry
Ice-cream production

ABSTRACT

The production scheduling of a real-world multistage food process is considered in this work. An efficient mixed integer programming (MIP) continuous-time model is proposed to address the production problem under study. The overall mathematical framework relies on an efficient modeling approach of the sequencing decisions, the integrated modeling of all production stages, and the inclusion of a set of strong tightening constraints. The simultaneous optimization of all processing stages aims at facilitating the interaction among the different departments of the production facility. Moreover, an alternative MIP-based solution strategy is proposed for dealing with large-scale food processing scheduling problems. Although this method may no guarantee global optimality, it favors low computational requirements and solutions of very good quality. Several problem instances are solved to reveal the salient computational performance and the practical benefits of the proposed MIP formulation and solution strategy.

1. Introduction

Process scheduling is generally defined as the allocation of the available resources over time to perform a collection of specific processing tasks optimizing a given objective. This problem has been studied extensively by several research disciplines and a plethora of solution approaches, most of them involving mathematical programming frameworks, can be found in the literature of the Process Systems Engineering (PSE) community (Kallrath, 2002; Maravelias & Sung, 2009; Méndez, Cerdá, Grossmann, Harjunkoski, & Fahl, 2006). Nevertheless, the food processing industry sector, which is part of the process industries, has received relatively little attention so far in the scheduling research area.

The main purpose of the food processing industries is to add value to the basic agricultural or farm raw materials by mixing, separating, forming or biochemical reactions to obtain final food products. In general, a food production process can be divided into three main stages: (i) processing of raw materials into intermediate products, (ii) storage of intermediate products and usually further processing (e.g., fermentation, aging) in buffer tanks, and (iii) packing of final products. As a natural consequence, most food production facilities operate in semicontinuous production mode, since batch and continuous processes are present in the overall production route.

Production scheduling in the food processing industry is specifically induced by the characteristics of the processes, thus industry-specific approaches are usually preferred to generic frameworks. For instance, in dairy industries a natural sequence of products often exists (e.g., from the lower taste to the stronger, from low to high concentrations of a certain additive, or from the brighter color to the darker).

Although numerous Mixed Integer Programming (MIP) models addressing production scheduling problems can be found in the literature of the operational research and PSE communities, the use of optimization-based techniques for scheduling food processing plants is still in its infancy.

Entrup, Günther, Van Beek, Grunow, and Seiler (2005) presented three discrete/continuous-time MIP models, based on the definition of products groups, for scheduling and planning problems in yogurt packing lines. Shelf life issues and fermentation capacity limitations were considered, however production costs as well as sequence-depended changeover times were ignored. The latter makes the proposed models more appropriate to cope with planning rather than scheduling problems, where products changeovers details are crucial. Marinelli, Nenni, and Sforza (2007) studied the planning problem in yogurt packing lines that share resources, having as optimization goal the minimization of inventory, production and machines setup cost. A discrete mathematical model which failed to obtain the optimal solution of the real application in an acceptable computation time was proposed. As a result, a two-stage heuristic based on the decomposition of the problem into a lot-sizing problem and a scheduling problem was

* Corresponding author.
E-mail address: mgeorg@otenet.gr (M.C. Georgiadis).

0098-1354/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2011.12.015

Nomenclature

Indices/sets

$b, b' \in B$	product batches (batches)
$i, i' \in I$	product orders (products)
$j, j' \in J$	processing units (units)
$s \in S$	processing stages (stages)

Subsets

I_j	products i that can be processed in unit j
I_i^{suc}	successor of product i
I_i^{SP}	products that share the same dedicated packing line with product i
J_i	available units j to process product i
J_s	available units j to process stage s
J_{ij}^{last}	highest indexed aging vessel $j \in J_{s_2}$ where product i can be stored

Parameters

α_j^{min}	minimum number of products assigned to packing line $j \in J_{s_3}$
β_i^{min}	minimum number of batches for product i
$\gamma_{ii'j}$	sequence-dependent changeover time between product i and i' in unit $j \in (J_i \cap J_{i'})$
γ_j^{min}	minimum sequence-dependent changeover time between two different products in packing line $j \in J_{s_3}$
ε_i^{life}	shelf life for product i in aging vessels
ζ_i	demand for product i
θ_i	position of product i in its dedicated packing line
λ_i	total number of alternative storage vessels for product i
μ_j^{max}	maximum capacity of aging vessel $j \in J_{s_2}$
ρ_{ij}	processing rate for every product i in the process line $j \in (J_i \cap J_{s_1})$ and the packing lines $j \in (J_i \cap J_{s_3})$
τ_i^{ag}	minimum aging time for product i
τ_i^{empt}	emptying time of aging vessel from product i
τ_i^{fill}	filling time of aging vessel with product i
ϕ_j^{min}	minimum wait time to begin using packing line j
ω	available production horizon

Continuous variables

C_{ibs}	completion time for stage s of batch b of product i
C_{max}	makespan
L_{ibs}	starting time for stage s of batch b of product i
W_{ibs}	standing time for stage s of batch b of product i

Binary variables

$X_{ii'}$	= 1 if product i is processed before product i' (for the aging vessels and the packing lines)
$\overline{X}_{ibi'b'}$	= 1 if batch b of product i is processed before batch b' of product i' (for the process line)
Y_{ibsj}	= 1 if stage s of batch b of product i is assigned to unit j

developed. Sequence-dependent changeover costs and times were not considered.

Kopanos, Puigjaner, and Georgiadis (2010) presented a mixed discrete/continuous-time MIP formulation, based on the definition of families of product, for the simultaneous lot-sizing and scheduling in real-life yogurt packing lines. Timing and capacity constraints for the fermentation stage were taken into account despite the fact that the problem was focused on the packing stage. In addition, sequence-dependent changeover times and costs, production overtime and typical daily production shutdown and setup times were considered. More recently, this mathematical formulation was extended by Kopanos, Puigjaner, and Georgiadis (2011b) so as to deal with renewable resource constraints (e.g., manpower). Furthermore, the definition of product families was more generalized allowing to products that belong to the same family to have different: (i) packing rates, (ii) setup times/costs, (iii) minimum/maximum production runs, (iv) operating costs, and (v) inventory costs.

Amorim, Antunes, and Almada-Lobo (2011) proposed two different MIP models for the multi-objective lot-sizing and scheduling problem considering perishability issues. Multiple objectives included production related costs (i.e., sequence-dependent changeover costs, setup costs and machines costs), and the mean freshness of final products. A hybrid genetic algorithm was developed to solve the proposed models, and it was tested in various problem instances based on the real-life dairy plant described by Kopanos, Puigjaner, et al. (2010). Recently, Kopanos, Puigjaner, and Georgiadis (2011a) developed a MIP formulation and proposed a simple intuitive solution strategy for the production scheduling in real-life multistage ice-cream processes. The main features of this approach rely on the integrated modeling of all production stages, and the inclusion of strong valid integer cuts favoring shorter computational times. Recently, Gellert, Höhn, and Moehring (2011) presented a scheduling framework for packing lines in dairy industry. Various algorithms were developed to solve this problem in an efficient manner. However, the proposed modeling framework does not explicitly consider other production stages of the overall plant.

In this work, the MIP formulation of Kopanos et al. (2011a) is further enhanced by introducing new sets of tightening constraints in order to improve the computational efficiency in industrial-size scheduling problems in multiproduct multistage food industries, such as the ice-cream production facility considered here. Moreover, a new MIP-based solution strategy is proposed to efficiently cope with large-scale industrial problems. Although the proposed MIP and the MIP-based solution frameworks are well-suited to a real-life ice-cream production process, they could be also used in scheduling problems that arise in other food industries with similar processing characteristics.

The paper is organized as follows. Section 2 describes the process under consideration and discusses the main drawbacks of typical current scheduling policies in food industries. In Section 3, we formally define the production scheduling problem considered in this work. The pre-optimization step for solving the batching problem is described in Section 4. Then, Section 5 presents an MIP formulation including strong tightening constraints. Section 6 describes an efficient decomposition-based solution approach for solving large scale problem instances as illustrated in Section 7. Finally, concluding remarks are drawn in Section 8.

2. The ice-cream production process

The food production process considered is derived from a real-world ice-cream production facility, originally introduced and studied by Bongers and Bakker (2006). A schematic diagram of the ice-cream production process is shown in Fig. 1.

A total number of eight ice-cream products (A–H) is produced in the multistage semicontinuous production plant. First, the basic mixes are produced in the main process line (PROC), by mixing the raw materials according to the recipes. Afterwards, these intermediate products are stored into aging vessels (V1–V6), where they are cooled down for a minimum aging time. Finally, the frozen

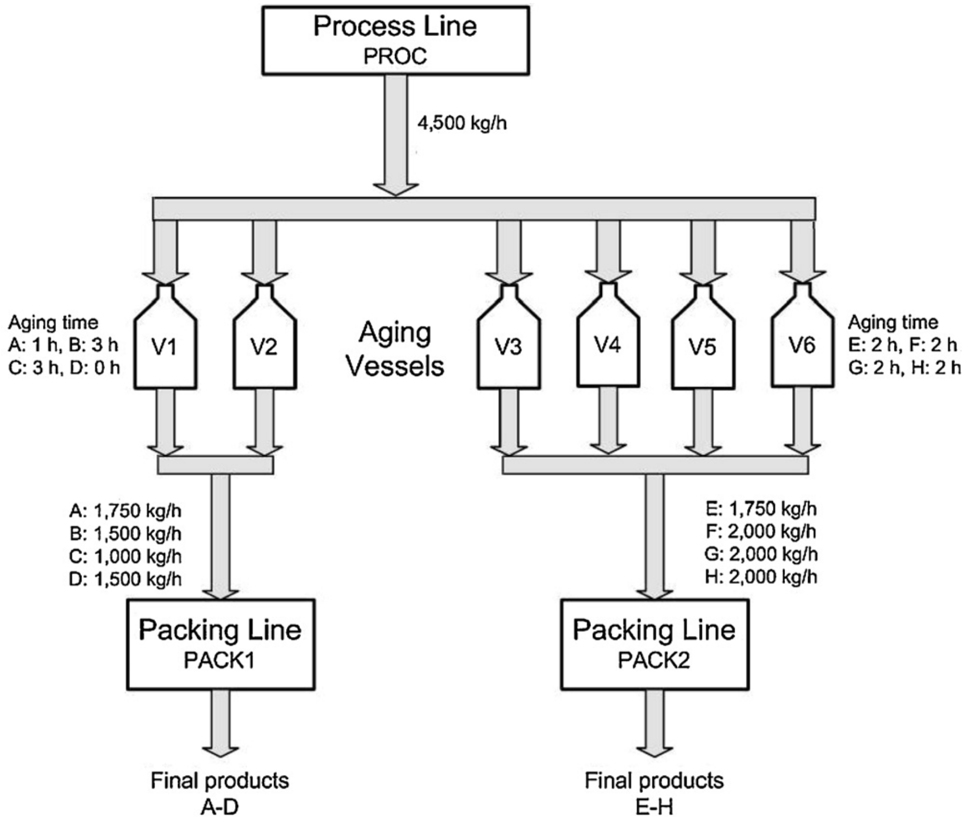

Fig. 1. Ice-cream production facility.

intermediate products are sent for packing in two packing lines (PACK1, PACK2), thus producing the final products.

The main process line PROC has a processing rate of 4500 kg/h, and can feed all aging vessels; one at a time. Packing line PACK1 is supplied by two aging vessels (V1 and V2) of 8000 kg capacity each and can accommodate products A–D, whereas packing line PACK2 can pack products E–H and is supplied by four aging vessels (V3–V6) of 4000 kg capacity each. Minimum aging times and packing rates for all products are given in Fig. 1. Also, there is a maximum shelf-life for all intermediate mixes in the aging stage (72 h). Sequence-dependent changeover, or simply changeover, operations (mainly cleaning and sterilizing tasks) are performed both in the process and the packing lines whenever the production is switched from a product to another. Table 1 gives these changeover times. Finally, a cleaning time of 2 h is needed before shutting down the process line and the packing units.

It should be pointed out that the typical industrial scheduling practice in food processing plants is focused on just scheduling the packing lines. The packing schedule is then sent to the process department, which should define a new schedule to meet packing product demands. The derived process schedule is then sent to the materials department, wherein a new schedule should be made to define orders for raw materials. It should be noted that this industrial practice is valid when the only bottleneck of the entire process are the packing lines, featuring very low packing rates compared to the flow rates of the previous stages. However, there are food processes, such as the ice-cream process studied here, where the packing rates are comparable to the processing rates at previous stages. That means that both process and packing lines could be the bottlenecks. Apparently, the current way that the majority of the food industries are being scheduled is posing two major problems: (i) less available production capacity due to bad coordination among the different departments, and (ii) high risk of infeasible schedules in the upstream departments in any packing schedule modification. Since the above problems are frequently met in relevant industrial environments, the challenge is to appropriately tackle them in an integrated manner so as to improve the plant production capacity and simultaneously reduce the total production costs for the final products. This industrial reality sets the simultaneous scheduling of all processing stages (i.e., the process line, the aging vessels, and the packing lines) as a challenging problem in the food processing industries.

3. Problem statement

The multistage semicontinuous process considered in this work has the following characteristics:

(i) A set of products $i \in I$ should be processed through a predefined sequence of processing stages $s \in S$ (i.e., process (s_1), storage/aging (s_2), and packing (s_3) stage), with processing units $j \in J$ working in parallel.
(ii) Product i can be processed in a specific subset of units $j \in J_i$. Likewise, processing stage s can be performed in a specific subset of units $j \in J_s$.
(iii) The total demand ζ_i for each product i is divided into a number of batches $b \in B$.
(iv) In aging vessels, a product batch should remain for a minimum aging time (τ_i^{ag}), and no longer than its corresponding shelf-life (ε_i^{life}). Aging vessels $j \in J_{s_2}$ have a maximum storage capacity μ_j^{max}.
(v) Process and packing lines are characterized by specific processing and packing rates (ρ_{ij}) depending on product i.

Table 1
Changeover times in the process and packing lines (in min).

Product	Process line								Packing lines							
	A	B	C	D	E	F	G	H	A	B	C	D	E	F	G	H
A	0	30	30	30	30	30	30	30	0	60	60	60	–	–	–	–
B	30	0	30	30	30	30	30	30	30	0	60	60	–	–	–	–
C	30	30	0	30	30	30	30	30	30	30	0	60	–	–	–	–
D	30	30	30	0	30	30	30	30	30	30	30	0	–	–	–	–
E	30	30	30	30	0	15	15	15	–	–	–	–	0	60	60	60
F	30	30	30	30	5	0	15	15	–	–	–	–	30	0	60	60
G	30	30	30	30	5	5	0	15	–	–	–	–	30	30	0	60
H	30	30	30	30	5	5	5	0	–	–	–	–	30	30	30	0

(–) Impossible subsequence.

(vi) A sequence-dependent changeover time ($\gamma_{ii'j}$) between successive product batches is required in the process and the packing stage mainly for cleaning and sterilization operations.
(vii) All model parameters are deterministic, and a non-preemptive production mode applies.

The key decisions to be made are:

(i) the allocation of product batches i, b to units $j \in J_i$ per stage, Y_{ibsj};
(ii) the relative sequence for any pair of product batches i, b and i', b' in the process line (s_1), $\overline{X}_{ibi'b'}$;
(iii) the relative sequence for any pair of products i and i' in aging vessels (s_2) and packing lines (s_3) for $j \in (J_i \cap J_{i'} \cap J_s)$, $X_{ii'}$; and
(iv) the starting and completion time of product batch i, b in every stage s; L_{ibs} and C_{ibs}, respectively.

So as to minimize the makespan (C_{max}) under full demand satisfaction.

4. Pre-optimization step: solving the batching problem

In most food industries, such as the ice-cream factory considered, the intermediate storage units (e.g., aging vessels, fermentation tanks) operate in their *maximum capacity*; so as to favor higher production levels and less production changeovers. As a result, the batching problem (i.e., the number and the size of batches per product) can be solved a priori. In addition to that, in the ice-cream production processes studied, each product i can be stored into a number of *equal-capacity* aging vessels, which are supplied by a *single* process line.

All the above allow us to calculate:

(i) the minimum number of batches to fully satisfy the demand for each product i: $\beta_i^{min} = \zeta_i / \mu_j^{max}$, where $j \in (J_i \cap J_{s_2})$;
(ii) the filling time of product i for each aging vessel: $\tau_i^{fill} = \mu_j^{max} / \rho_{ij'}$, where $j \in (J_i \cap J_{s_2})$ and $j' \in (J_i \cap J_{s_1})$; and
(iii) the emptying time of product i for each aging vessel: $\tau_i^{empt} = \mu_j^{max} / \rho_{ij'}$, where $j \in (J_i \cap J_{s_2})$ and $j' \in (J_i \cap J_{s_3})$.

Notice that unit $j' \in J_{s_1}$ corresponds to the process line, unit $j \in J_{s_2}$ to the aging vessels, and unit $j' \in J_{s_3}$ to the packing lines. Apparently, the aging vessels filling (emptying) time equals to the processing (packing) time in the process (packing) line due to the continuous process mode between these stages.

5. Mathematical formulation

To facilitate the presentation of the model, we use uppercase Latin letters for optimization variables and sets, and lowercase Greek letters for parameters.

The objective function to be optimized is the minimization of makespan (i.e., the time point at which all product demands are met):

$$\min \quad C_{max} \geq C_{ibs} \quad \forall i, \ b \leq \beta_i^{min}, \ s \geq 2 \tag{1}$$

subject to:

$$\sum_{j \in (J_i \cap J_s)} Y_{ibsj} = 1 \quad \forall i, \ b \leq \beta_i^{min}, \ s \tag{2}$$

$$L_{ibs} + \tau_i^{fill} = C_{ibs} \quad \forall i, \ b \leq \beta_i^{min}, \ s = 1 \tag{3}$$

$$L_{ibs} + \tau_i^{fill} + \tau_i^{ag} + W_{ibs} + \tau_i^{empt} = C_{ibs} \quad \forall i, \ b \leq \beta_i^{min}, \ s = 2 \tag{4}$$

$$W_{ibs} \leq \varepsilon_i^{life} - \tau_i^{ag} \quad \forall i, \ b \leq \beta_i^{min}, \ s = 2 \tag{5}$$

$$L_{ibs} + \tau_i^{empt} = C_{ibs} \qquad \forall i, b \leq \beta_i^{min}, s = 3 \tag{6}$$

$$L_{ibs} = L_{ibs-1} \qquad \forall i, b \leq \beta_i^{min}, s = 2 \tag{7}$$

$$C_{ibs} = C_{ibs-1} \qquad \forall i, b \leq \beta_i^{min}, s = 3 \tag{8}$$

$$C_{ibs} = L_{ib+1s} \qquad \forall i, b < \beta_i^{min}, s = 3 \tag{9}$$

$$\begin{gathered} L_{i'b's} \geq C_{ibs} + \gamma_{ii'j} - \omega(1 - \overline{X}_{ibi'b'}) - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s = 1 \end{gathered} \tag{10}$$

$$\begin{gathered} L_{ibs} \geq C_{i'b's} + \gamma_{i'ij} - \omega \overline{X}_{ibi'b'} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s = 1 \end{gathered} \tag{11}$$

$$\begin{gathered} L_{i'b's} \geq C_{ibs} + \gamma_{ii'j} - \omega(1 - X_{ii'}) - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s > 1 \end{gathered} \tag{12}$$

$$\begin{gathered} L_{ibs} \geq C_{i'b's} + \gamma_{i'ij} - \omega X_{ii'} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s > 1 \end{gathered} \tag{13}$$

$$\begin{gathered} L_{ib's} \geq C_{ibs} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \\ \forall i, b \leq \beta_i^{min}, b' \leq \beta_i^{min}, s, j \in (J_i \cap J_s) : b < b' \end{gathered} \tag{14}$$

$$\begin{aligned} Y_{ibsj} &\in \{0, 1\} \quad \forall i, b \leq \beta_i^{min}, s, j \in (J_i \cap J_s) \\ \overline{X}_{ibi'b'} &\in \{0, 1\} \quad \forall i, b \leq \beta_i^{min}, i', b' \leq \beta_{i'}^{min}, \\ &\qquad s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s = 1 \\ X_{ii'} &\in \{0, 1\} \quad \forall i, i', s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s > 2 \\ L_{ibs}, C_{ibs} &\geq 0 \quad \forall i, b \leq \beta_i^{min}, s \\ W_{ibs} &\geq 0 \quad \forall i, b \leq \beta_i^{min}, s = 2 \end{aligned} \tag{15}$$

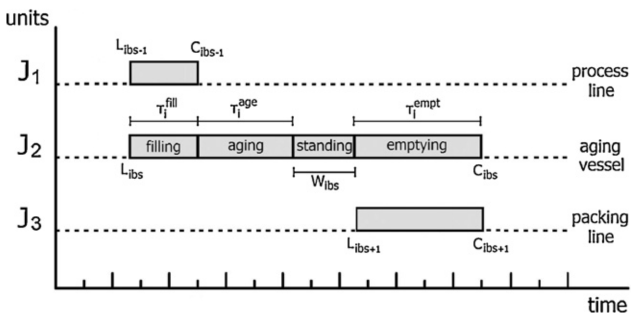

Fig. 2. Timing decisions for a product batch i, b for every processing stage.

Constraints (2) are typical unit allocation constraints for any product batch. Constraints (3)–(6) define the timing of any product batch in each stage (see Fig. 2). Note that variable W_{ibs} corresponds to the standing (waiting) time for any product batch in the aging stage. Constraints (7) and (8) define the timing of every product batch between two successive processing stages, as illustrated in Fig. 2, and constraints (9) provide the timing of two batches of the same product in the packing stage under single production campaign. In addition, constraints (10)–(14) define the relative sequencing between two product batches in all processing stages. Constraints (10)–(13) are big-M constraints, where the available production horizon ω plays the role of the M parameter. Notice that the proposed MIP model uses global sequencing variables: (i) for any pair of product batches i, b and i', b' ($i < i'$) in the process stage $\overline{X}_{ibi'b'}$, and (ii) for any pair of different products i and i' ($i < i'$) both in the aging and the packing stage $X_{ii'}$. Also, observe that sequencing decisions are the same for the aging and packing stage and they are thus modeled for both stages through a single binary variable $X_{ii'}$ for a given pair of products. Finally, constraint (15) define the domains of all decision variables. For a more detailed description of the above constraints refer to Kopanos et al. (2011a).

5.1. Tightening constraints

The computational efficiency of the proposed mathematical framework can be further improved by incorporating the following tightening constraints.

5.1.1. Makespan lower bound

A lower bound on the makespan objective is given by:

$$C_{max} \geq \phi_j^{min} + (\alpha_j^{min} - 1)\,\gamma_j^{min} + \sum_{i \in I_j} \tau_i^{empt}\,\beta_i^{min} \qquad \forall j \in J_{s_3} \tag{16}$$

Parameter ϕ_j^{min} corresponds to the minimum waiting time to begin using packing line $j \in J_{s_3}$, which depends on the minimum filling time for aging vessels $j' \in J_{s_2}$ that are connected to packing line j. Moreover, parameter α_j^{min} represents the minimum number of products that must be assigned to packing line j to ensure full demand satisfaction, while parameter γ_j^{min} stands for the minimum changeover time between two products in packing line j.

5.1.2. Relative sequence correlation of the process and packing stage

Constraints (17) force the relative sequence between products i and $i' > i$ in the packing (and aging) stage to maintain the same for all product batches i, b and i', b' in the process stage; for products i and i' that share the same packing line (i.e., $i' \in I_i^{SP}$).

$$\overline{X}_{ibi'b'} = X_{ii'} \qquad \forall i, b \leq \beta_i^{min}, i' \in I_i^{SP}, b' \leq \beta_{i'}^{min} : i < i' \tag{17}$$

In other words, as Fig. 3 illustrates, if product i is assigned before product i' to packing unit j, constraint set (17) drives all the batches of product i to be allocated to the process line before any batch of product i'. It is worth noticing that by this rule the reduction of changeovers in the process line is favored, as it can be clearly seen in Fig. 3.

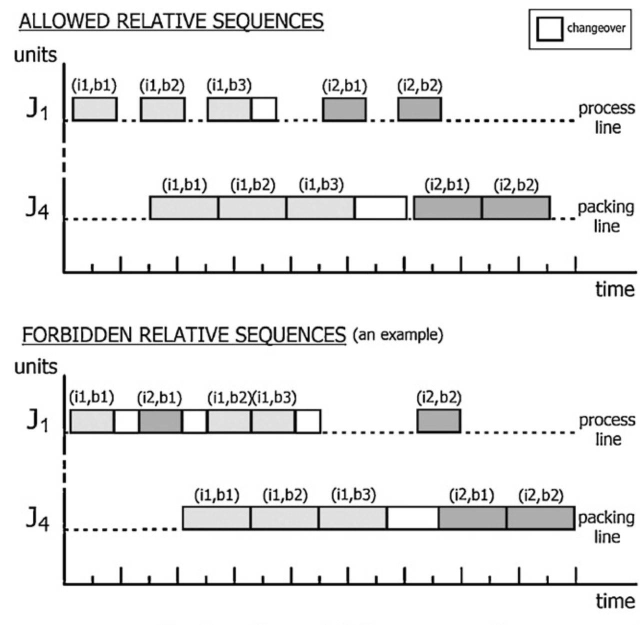

Fig. 3. Relative sequences for batches of different products according to constraints (17).

5.1.3. Symmetry-breaking of allocation decisions for batches of the same product

The unit allocation decision process for batches of the same product, in the aging stage, is a very potential source for degenerate solutions.

For instance, let consider the simple case according to which two batches b_1 and b_2, of the same product, are to be allocated to units J_1 and J_2. Note that the alternative allocation decisions, i.e., $\{b_1$ to J_1; b_2 to $J_2\}$ or $\{b_1$ to J_2; b_2 to $J_1\}$, are equivalent since batches b_1 and b_2 have the same batch size and processing time. In more details, the batch names "b_1" and "b_2" are irrelevant and the actual decision to be taken (by the model) is the allocation of two batches of product i: one to unit J_1 and the other to unit J_2. Apparently, a large number of equivalent batch-to-unit allocation decisions for each product are generated (and explored), when several batches for a specific product and multiple storage units are available.

In order to break the aforementioned symmetry of the batch-to-unit allocation decisions, and after a thorough examination of the problem under study, the following constraints are introduced to the MIP formulation:

$$Y_{ibsj} = Y_{ib+1sj+1} \qquad \forall i, b < \beta_i^{min}, s, j \in (J_i \cap J_s) : s = 2, j \notin J_{ij}^{last} \tag{18}$$

$$Y_{ibsj} = Y_{ib+1sj-(\lambda_i-1)} \qquad \forall i, b < \beta_i^{min}, s, j \in (J_i \cap J_s) : s = 2, j \in J_{ij}^{last} \tag{19}$$

$$Y_{ibsj} = Y_{ib+\lambda_i sj} \qquad \forall i, b \leq (\beta_i^{min} - \lambda_i), s, j \in (J_i \cap J_s) : s = 2 \tag{20}$$

Notice that λ_i denotes the total number of available alternative storage vessels for product i, and J_{ij}^{last} is the highest indexed aging vessel where product i could be stored. Constraints (18) and (19) set the unit allocation rule for the successive batches b and $b+1$ for product i. This rule states that when batch b is allocated to unit $j \notin J_{ij}^{last}$, batch

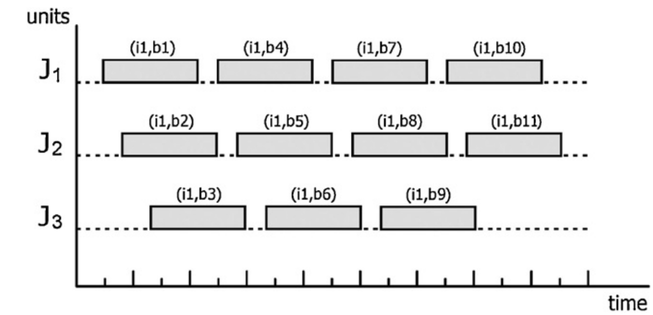

Fig. 4. Symmetry breaking of allocation decisions for batches of the same product.

$b+1$ should be assigned to unit $j+1$, while when batch b is stored in unit $j \in J_{ij}^{last}$, batch $b+1$ is allocated to unit $j-(\lambda_i-1)$. Furthermore, constraints (20) guarantee that, for a specific product i, if batch b is assigned to storage vessel j, then batch $b+\lambda_i$ is also assigned to the same storage vessel, and vice versa. Fig. 4 depicts an illustrative example of the batch-to-unit allocation decisions for 11 batches of product i_1 to three units (J_1–J_3).

5.1.4. Correlation of allocation decisions for successive products

A special feature of the ice-cream process considered, as well as in other food processes such as yogurt production, is that there exist a natural sequence for products in the packing stage. Moreover, every product can be packed in a specific packing line. These characteristics allow us to define correlated batch-to-unit allocation constraints between the last ($b=\beta_i^{min}$) and the first batch ($b'=1$) of two successive products i and i'; where i' is processed exactly after i (i.e., $i' \in I_i^{suc}$):

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall s, j \in J_s, i \in I_j, b, i' \in (I_i^{suc} \cap I_{j+1}), b' : s=2, b=\beta_i^{min}, b'=1, j \notin J_{ij}^{last} \tag{21}$$

$$Y_{ibsj} = Y_{i'b'sj-(\lambda_{i'}-1)} \quad \forall s, j \in J_s, i \in I_j, b, i' \in (I_i^{suc} \cap I_{j-(\lambda_{i'}-1)}), b' : s=2, b=\beta_i^{min}, b'=1, j \in J_{ij}^{last} \tag{22}$$

Let consider the case that three products i_1, i_2, and i_3 (with $\beta_{i_1}^{min}=5$, $\beta_{i_2}^{min}=4$, and $\beta_{i_3}^{min}=3$) are to be assigned into three units (J_1–J_3). Additionally, there is a natural sequence: $i_1 \to i_2 \to i_3$. The aging vessels schedule of that illustrative example is presented in Fig. 5. Note that: (i) $Y_{i_1 b_5 J_2} = Y_{i_2 b_1 J_3} = 1$ in accordance with constraints (21), and (ii) $Y_{i_2 b_4 J_3} = Y_{i_3 b_1 J_1} = 1$ in agreement with constraints (22).

The overall MIP formulation consists of constraints (1)–(22).

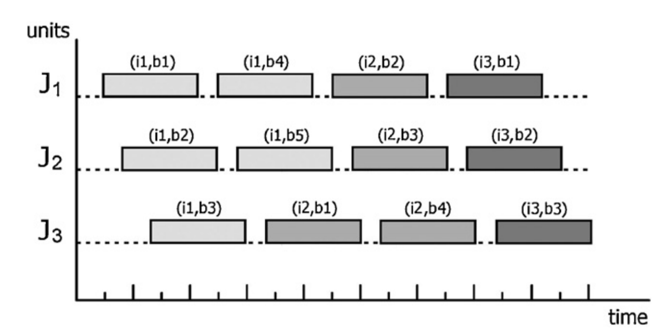

Fig. 5. Correlation of allocation decisions between the last batch of product i and the first batch of product $i' \in I_i^{suc}$.

5.2. Refinement of the mathematical formulation

The mathematical framework described in the previous sections can be further refined by reconsidering the problem-specific features of: (i) the existence of a single process line, (ii) the lack of allocation flexibility in the packing stage (i.e., every product can be packed in to one dedicated packing line), and (iii) the natural sequence of products in the packing lines. As a result, batch-to-unit decision variables, Y_{ibsj}, in the process and packing lines as well as the sequencing variables, $X_{ii'}$, in the packing stage are redundant. A description of the new sets of constraints for the problem-specific MIP model follows.

The refined mathematical formulation is mainly based on the exclusion of the aforementioned decision variables from the original MIP model, by simply defining: (i) products subsets I_i^{SP} which contain all products that share the same dedicated packing line with product i, and (ii) parameters θ_i that denote the a priori known position of product i in its dedicated packing line. That way, in the new mathematical formulation, the batch-to-unit allocation constraints (2) are replaced by constraints (23), and constraints (24)–(30) take the place of constraints (10)–(15). Also, note that constraints (17) are not needed due to the definition of parameters θ_i. The new sets of constraints are given below.

$$\sum_{j \in (J_i \cap J_s)} Y_{ibsj} = 1 \quad \forall i, \ b \le \beta_i^{min}, \ s=2 \tag{23}$$

$$L_{i'b's} \ge C_{ibs} + \gamma_{ii'j} - \omega(1 - \overline{X}_{ibi'b'}) \quad \forall i, b \le \beta_i^{min}, i' \notin I_i^{SP}, b' \le \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s=1 \tag{24}$$

$$L_{ibs} \ge C_{i'b's} + \gamma_{i'ij} - \omega \overline{X}_{ibi'b'} \quad \forall i, b \le \beta_i^{min}, i' \notin I_i^{SP}, b' \le \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s=1 \tag{25}$$

$$L_{i'b's} \ge C_{ibs} + \gamma_{ii'j} \quad \forall i, b \le \beta_i^{min}, i' \in I_i^{SP}, b' \le \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : \theta_i < \theta_{i'}, s \ne 2 \tag{26}$$

$$L_{i'b's} \ge C_{ibs} + \gamma_{ii'j} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \quad \forall i, b \le \beta_i^{min}, i' \in I_i^{SP}, b' \le \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : \theta_i < \theta_{i'}, s=2 \tag{27}$$

$$L_{ib's} \ge C_{ibs} \quad \forall i, b \le \beta_i^{min}, b' \le \beta_i^{min}, s, j \in (J_i \cap J_s) : b < b', s \ne 2 \tag{28}$$

$$L_{ib's} \ge C_{ibs} - \omega(2 - Y_{ibsj} - Y_{ib'sj}) \quad \forall i, b \le \beta_i^{min}, b' \le \beta_i^{min}, s, j \in (J_i \cap J_s) : b < b', s=2 \tag{29}$$

$$\begin{gathered} Y_{ibsj} \in \{0,1\} \quad \forall i, b \le \beta_i^{min}, s, j \in (J_i \cap J_s) : s=2 \\ \overline{X}_{ibi'b'} \in \{0,1\} \quad \forall i, b \le \beta_i^{min}, i' \notin I_i^{SP}, b' \le \beta_{i'}^{min}, s, j \in (J_i \cap J_{i'} \cap J_s) : i < i', s=1 \\ L_{ibs}, C_{ibs} \ge 0 \quad \forall i, b \le \beta_i^{min}, s \\ W_{ibs} \ge 0 \quad \forall i, b \le \beta_i^{min}, s=2 \end{gathered} \tag{30}$$

More specifically, big-M constraints (24) and (25) define the relative sequencing and timing of the process line between two batches of two different products i and i' that do not share the same dedicated packing line (i.e., $i' \notin I_i^{SP}$). Additionally, the timing between two batches of two different products i and i' that share the same packing line (i.e., $i' \in I_i^{SP}$): in the process and packing stage is given by constraints (26), and in the aging stage is defined by constraints (27). Besides, constraints (28) and (29) set the timing in all processing stages between two batches of the same product. Finally, constraint sets (30) define the domains of the reduced set of decision variables.

Therefore, the refined mathematical formulation, called **MIP-R**, consists of constraints sets (1), (3)–(9), (16) and (18)–(30).

5.3. Remarks

At this point, it should be emphasized that the proposed mathematical formulation has been developed to tackle a specific problem, wherein there is no allocation flexibility in the packing stage. Notice that constraints (9) in the original model may be valid only if the previous condition is satisfied. In case that more than one packing line is available to discharge a vessel, violation of constraints (9) might be convenient and generate better solutions. The same hold for other constraints of the original mathematical model.

Additionally, notice that the correlation of allocation decisions for successive products works only if all products assigned to a packing line can be allocated in exactly the same aging vessels. If only some of the vessels connected to a packing line could be used for the aging of a product, then this set of constraints does not necessarily apply. Similar issues are encountered in the symmetry breaking constraints, where such a set of constraints requires that all products are allocated in sequentially ordered equipments.

Finally, the tightening constraints and constraints (9) are very suitable for the makespan optimization objection. However, they may not be appropriate when other objective function such as tardiness or lateness is optimized. For those cases, it is possibly more convenient to break the packing of a product in more than one campaigns depending on demand due dates.

6. MIP-based solution strategy

The combinatorial complexity of the proposed MIP model increases significantly with the number of products and their demands, potentially hindering the solution of large-scale scheduling problems by exact solution methods. For this reason, an alternative solution approach is developed to improve the computational efficiency of the proposed MIP model for solving large-scale scheduling problems involving a large number of products and longer production horizons. This method has as a basic core the proposed MIP model, and it is based on the idea of a reduced search space, which usually results in manageable model sizes that often ensure a more stable and predictable optimization model behavior.

In the proposed solution strategy, the overall scheduling problem is broken down by product batches. First, all product batches are ranked depending on the dedicated packing line and their position (θ_i) there. We also rank lower (i.e., schedule first) products: (i) that are dedicated to the less-flexible packing lines, regarding the available number of storage vessels connected to them, and (ii) that are packed earlier (i.e., lower θ_i). Let consider a simple case assuming a single batch demand for every product. Note that the less-flexible packing line is PACK1 which is fed by two aging vessels in contrast to PACK2 that is fed by four aging vessels. According to Table 1, the natural sequence in PACK1 is: D → C → B → A, and in PACK2 is: H → G → F → E. Therefore, taking into consideration all the above issues the rank of batches for this example is: {1.D, 2.C, 3.B, 4.A, 5.H, 6.G, 7.F, and 8.E}.

The MIP-based solution method, called **MIP-BasB**, consists of the following two basic steps:

1. No more that a specific number of batches are scheduled (depending on their rank) in an iterative fashion, by fixing their batch-to-unit allocation and their relative sequencing decisions (i.e., Y_{ibsj}, $\overline{X}_{ibi'b'}$) at the end of each iteration. When all batches have been inserted, a feasible schedule for the complete scheduling problem is obtained.
2. Afterwards, a complete re-optimization (i.e., re-sequencing) of the process line is realized, by previously fixing binary batch-to-unit allocation decisions Y_{ibsj} for the aging stage. That way, the feasible schedule of the previous step is hopefully improved.

Note that the number of iterations in the first step depends on the total number of batches and the number of batches inserted in each iteration. For instance, if a total number of 95 batches are to be scheduled and no more that 25 batches are inserted per iteration, that means that the first step will need $95/25 \approx 4$ iterations.

At this point, it is essential to mention that the MIP-BasB can no longer assure global optimality, however favors lower computational times. From a practical point of view, guaranteeing global optimality may not be relevant in many industrial scenarios mainly due to the following features: (i) a very short time is just available to generate a solution and send it to the plant floor, (ii) optimality is easily lost because of the highly dynamic nature of industrial environments, (iii) implementing the schedule as such is limited by the real process, and (iv) only a part of the real scheduling goals are generally taken into account in the model since not all scheduling objectives can be quantified.

7. Case studies

The real-life ice-cream production facility described in Section 2 is considered in an attempt to demonstrate the efficiency of the proposed MIP-R formulation and the MIP-BasB solution strategy. A total set of 51 different problem instances, regarding the number of products and their demands, have been solved. All problem instances have been solved in a Dell Inspiron 1520 2.0 GHz with 2 GB RAM using CPLEX 11 via a GAMS 22.8 interface (Rosenthal, 2010), by imposing a time limit of 600 CPU s.

7.1. Bongers and Baker case study

Final products demands for this case study can be found in Bongers and Bakker (2006). In this problem some decisions at the beginning of the production week of interest have been imposed by the previous production week. More specifically, product batches D.b1, G.b1, G.b2, and G.b3 have already passed from the process line and assigned to aging vessels V1, V3, V4, and V5, respectively. Moreover, these product batches have already completed the aging process at the beginning of the time horizon, and as such they are ready for passing to the packing stage again at $t = 0$. For this reason, in this example parameter $\phi_j^{min} = 0$.

Bongers and Bakker (2006) made the first attempt to solve this scheduling problem by using an *advanced commercial scheduling software*. As they have reported, a feasible schedule on all stages could not be derived automatically by applying the available solvers. They finally generated a feasible schedule ($C_{max} = 120$ h) *by manual interventions* with heuristics. Recently, Subbiah, Schoppmeyer, and Engell (2011) studied the same ice-cream production plant by using the framework of timed automata. A heuristic methodology was implemented to reduce the model size and a feasible solution ($C_{max} = 119.21$ h) was reported in 13.13 CPU s. However it cannot be ruled out that the heuristics employed pruned the optimal solution.

Here, we have used the original MIP model proposed to solve this scheduling problem. The optimal solution ($C_{max} = 118.55$ h) was

Table 2
Minimum aging times (in h) and packing rates (in kg/h).

i	τ_i^{ag}	ρ_{iPACK1}	ρ_{iPACK2}	i	τ_i^{ag}	ρ_{iPACK1}	ρ_{iPACK2}
A	1	1750	–	M	3	–	2250
B	3	1500	–	N	2	–	2000
C	3	1000	–	O	3	–	1750
D	0	1500	–	P	2	–	2250
E	2	–	1750	Q	4	2500	–
F	2	–	2000	R	2	1250	–
G	2	–	2000	S	3	1500	–
H	2	–	2000	T	1	2250	–
I	2	1750	–	U	1	–	1500
J	3	1500	–	V	2	–	2000
K	2	2000	–	W	2	–	1750
L	1	2000	–	X	2	–	2750

(–) Unavailable packing line.

Table 3
Natural sequences for products in the packing lines for all problem instances.

Problem	Packing line	Natural sequence of products
01–20	PACK1	D ⇨ C ⇨ B ⇨ A
	PACK2	H ⇨ G ⇨ F ⇨ E
21–40	PACK1	L ⇨ K ⇨ J ⇨ I ⇨ D ⇨ C ⇨ B ⇨ A
	PACK2	P ⇨ O ⇨ N ⇨ M ⇨ H ⇨ G ⇨ F ⇨ E
41–50	PACK1	T ⇨ S ⇨ R ⇨ Q ⇨ L ⇨ K ⇨ J ⇨ I ⇨ D ⇨ C ⇨ B ⇨ A
	PACK2	X ⇨ W ⇨ V ⇨ U ⇨ P ⇨ O ⇨ N ⇨ M ⇨ H ⇨ G ⇨ F ⇨ E

reached in 0.92 CPU s, despite the fact of the challenging (very high) total demand for final products. A comparison of the optimal production schedule derived from our MIP formulation with those in Bongers and Bakker (2006) and Subbiah et al. (2011) shows that: (i) our production schedule features a shorter makespan, (ii) the process line utilization is significantly reduced (by more than 10% comparing with the schedule reported by Bongers and Bakker (2006)), (iii) total waiting time in the aging stage is significantly reduced, and (iv) the proposed MIP formulation achieved the optimal production schedule in very low computational time (being 14 times faster that the time automata framework of Subbiah et al. (2011)).

Table 4
Problem instances 01–20: product demands (in 1000 kg).

i	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
A	80	88	48	32	24	40	96	48	56	16	32	16	120	32	48	120	104	80	120	96
B	48	16	16	8	24	56	16	64	72	40	80	72	72	80	96	72	56	16	136	64
C	32	24	64	80	72	32	72	64	16	120	32	96	56	120	64	8	104	80	32	128
D	8	40	32	40	64	80	24	48	104	40	112	72	32	40	96	136	48	160	96	88
E	112	20	48	80	36	88	68	32	160	16	124	100	48	104	124	112	76	120	76	252
F	12	48	32	52	120	60	40	124	24	144	16	180	48	132	176	40	216	68	144	104
G	48	64	80	16	48	92	76	84	12	144	144	32	200	48	92	72	68	120	180	52
H	24	84	80	112	64	36	112	68	120	48	68	60	84	120	16	188	84	164	88	124
$\sum_i \beta_i^{min}$	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150	160	170	180

Table 5
Problem instances 21–40: product demands (in 1000 kg).

i	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
A	8	32	56	8	40	24	64	16	24	56	48	8	40	64	16	8	24	32	8	64
B	16	48	16	56	16	48	8	24	16	32	48	40	40	24	80	24	64	88	16	16
C	24	8	8	16	40	24	16	56	80	8	32	56	24	16	56	8	56	40	80	48
D	8	32	24	24	8	24	16	8	16	16	8	8	40	56	56	56	64	16	64	64
E	8	64	12	44	36	52	68	60	16	100	40	20	52	16	100	24	4	8	20	152
F	36	12	20	4	52	8	24	12	72	32	88	12	44	24	20	44	28	36	100	20
G	8	16	44	12	12	40	20	40	12	28	12	40	48	44	136	56	68	28	52	60
H	12	24	32	24	64	40	8	12	48	20	12	96	44	16	20	20	56	84	20	60
I	16	24	16	24	8	16	16	32	16	8	64	32	40	56	16	80	24	24	72	16
J	16	8	8	40	16	40	24	16	24	80	24	56	40	16	8	24	64	112	8	48
K	48	8	24	24	40	16	32	64	16	24	16	24	40	24	56	72	8	8	72	128
L	32	16	40	8	40	24	72	16	32	56	32	56	40	64	24	88	8	16	64	64
M	16	8	20	16	8	32	4	16	16	8	16	72	52	116	20	24	68	176	20	28
N	44	16	8	24	36	12	8	60	8	40	16	20	44	32	68	56	80	12	76	20
O	44	24	60	100	24	76	104	20	128	20	104	88	40	120	20	96	116	68	148	140
P	28	48	28	16	24	12	40	84	28	72	56	12	44	12	20	80	24	60	52	16
$\sum_i \beta_i^{min}$	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150	160	170	180

Table 6
Problem instances 41–50: product demands (in 1000 kg).

i	41	42	43	44	45	46	47	48	49	50
A	16	24	8	24	16	16	56	32	40	56
B	8	16	8	8	8	40	40	32	8	56
C	16	24	24	40	16	8	16	64	16	56
D	8	8	32	16	56	48	48	32	64	80
E	12	20	28	16	48	36	40	60	88	136
F	4	4	4	36	24	28	40	8	4	8
G	20	12	12	12	20	20	40	44	8	64
H	8	20	40	36	4	32	20	4	84	8
I	24	24	8	8	24	8	24	56	24	8
J	8	24	16	8	8	8	16	16	64	16
K	24	16	16	16	48	24	16	48	40	64
L	24	8	32	32	16	8	16	8	72	8
M	12	40	12	12	28	24	40	8	4	4
N	20	12	20	12	56	12	40	148	136	80
O	20	20	20	20	24	4	20	36	8	40
P	4	8	40	8	44	4	40	12	20	28
Q	16	8	16	8	16	16	16	16	8	16
R	8	16	16	8	16	56	40	8	16	40
S	16	16	8	40	16	16	16	16	16	16
T	8	8	24	16	16	24	16	40	80	64
U	48	24	4	36	20	52	20	12	68	80
V	20	4	40	12	20	4	40	8	16	8
W	4	52	4	36	8	40	40	80	16	80
X	20	8	32	52	16	88	20	36	44	24
$\sum_i \beta_i^{min}$	70	80	90	100	110	120	140	160	180	200

7.2. Experimental studies

In this part, 50 problem instances have been solved in order to validate the performance of the MIP-R formulation and the MIP-BasB method. Problems instances 01–20 deal with eight products, as in the original case study. Problem instances 21–40 consider 16 products, while problem instances 41–50 cope with 24 products. Note that the total number of batches per problem instance is maintained high, variating from 70 to 200.

Minimum products aging times and packing rates for all problem instances, are provided in Table 2. The natural sequence of products in the packing lines for all problems is given in Table 3. Moreover, Table 1 gives the changeover times in the process and the packing lines for problems instances 01–20. Changeover times for the remaining problems instances 21–50 follow the same pattern. More specifically, in the packing lines, the changeover time for the successors (in the natural sequence given in Table 3) of a given product is equal to 30 min, otherwise a 60-min changeover is required. The complete changeover times data for problem instances 21–50 can be found in the e-supplement. In addition, Tables 4, 5 and 6 provide the product demands for problems involving 8, 16 and 24 products, respectively. Finally, the available production horizon ω for each problem instance is given by the expression: $\omega = 1.2(\phi_j^{min} + (\alpha_j^{min} - 1)\gamma_j^{min} + \sum_{i \in I_j} \tau_i^{empt}\, \beta_i^{min})$.

Table 7 presents the computational results for MIP-R and MIP-BasB for problems (01–20) considering eight products. Notice that the number of nodes reported for the MIP-BasB contains the aggregated number of nodes for all iterations of the two steps of the method. Δ(MIP-R) gives the deviation of the MIP-BasB solution from the solution of the MIP-R. It should be noted that both MIP-R

Table 7
Results for problem instances involving eight products.

Problem	MIP-R			MIP-BasB			
	C_{max}	CPU s	Nodes	C_{max}	CPU s	Nodes	Δ(MIP-R)
01	120.33	1.05	38	120.33	1.84	38	0.00%
02	118.17	9.40	4912	118.17	2.52	635	0.00%
03	131.48	3.11	460	131.48	2.13	367	0.00%
04	142.10	2.78	110	142.1	2.28	135	0.00%
05	149.66	3.41	163	149.66	2.76	0	0.00%
06	152.34	4.90	201	152.34	3.34	0	0.00%
07	161.47	11.03	484	161.47	3.81	459	0.00%
08	171.37	8.25	338	171.37	3.74	0	0.00%
09	175.82	4.51	130	175.82	5.60	20	0.00%
10	187.75	15.04	502	187.75	5.53	1060	0.00%
11	191.25	7.00	179	191.25	8.05	271	0.00%
12	206.42	8.09	181	206.42	6.31	5	0.00%
13	201.76	31.44	1722	201.76	10.57	2026	0.00%
14	223.56	25.73	828	223.56	7.16	203	0.00%
15	224.71	17.93	380	224.71	7.42	0	0.00%
16	222.06	296.64	26,008	222.06	18.68	471	0.00%
17	238.04	29.82	591	238.04	9.46	0	0.00%
18	251.49	42.26	540	251.49	13.06	61	0.00%
19	260.52	58.23	498	260.52	13.53	0	0.00%
20	291.75	59.30	534	291.75	20.18	12	0.00%
Average		32.00	1940		7.40	288	0.00%

Table 8
Results for problem instances involving 16 products.

Problem	MIP-R				MIP-BasB			
	C_{max}	CPU s	Nodes	Rel. gap	C_{max}	CPU s	Nodes	Δ(MIP-R)
21	119.83	5.76	2515	0.00%	119.83	6.41	2534	0.00%
22	121.62	1.58	21	0.00%	121.62	2.41	21	0.00%
23	127.25	36.15	16,951	0.00%	127.25	4.44	2087	0.00%
24	141.14	5.88	288	0.00%	141.14	3.23	107	0.00%
25	147.02	6.25	4789	0.00%	147.02	7.72	4885	0.00%
26	154.94	43.12	12,820	0.00%	154.94	5.53	1243	0.00%
27	162.94	600.00	82,794	1.63%	162.94	49.69	15,372	0.00%
28	181.21	600.00	49,573	5.61%	181.21	12.39	998	0.00%
29	181.23	2.51	533	0.00%	181.23	4.25	539	0.00%
30	187.46	105.15	24,713	0.00%	187.46	104.06	6073	0.00%
31	190.95	14.52	3542	0.00%	190.95	19.29	4090	0.00%
32	218.26	600.00	26,398	9.74%	214.21	154.48	69,173	−1.86%
33	216.94	600.00	36,061	8.04%	210.76	87.35	10,801	−2.85%
34	–	600.00	–	–	234.81	263.61	29,568	–
35	226.31	9.67	1049	0.00%	226.31	17.78	1797	0.00%
36	–	600.00	–	–	252.13	600	71,115	–
37	250.00	265.23	10,936	0.00%	265.30	241.44	32,170	6.12%
38	–	600.00	–	–	298.78	600	27,302	–
39	–	600.00	–	–	292.34	185.79	17,997	–
40	–	600.00	–	–	326.58	600	62,698	–
Average		294.79	18,199			148.49	18,029	

Table 9
Results for problem instances involving 24 products.

Problem	MIP-R				MIP-BasB			
	C_{max}	CPU s	Nodes	Rel. gap	C_{max}	CPU s	Nodes	Δ(MIP-R)
41	118.98	1.11	93	0.00%	118.98	2.04	127	0.00%
42	136.43	1.75	189	0.00%	136.43	3.05	214	0.00%
43	146.78	63.31	20,171	0.00%	146.78	11.19	7270	0.00%
44	164.99	118.28	15,831	0.00%	164.99	6.27	1671	0.00%
45	177.05	33.43	16,689	0.00%	177.05	36.32	16,699	0.00%
46	205.32	28.65	3344	0.00%	205.32	30.67	3397	0.00%
47	222.63	600.00	19,154	1.79%	221.66	38.42	624	−0.44%
48	262.51	600.00	6902	3.84%	258.16	98.45	32,796	−1.66%
49	294.32	600.00	66,915	0.32%	294.32	600.00	72,197	0.00%
50	332.19	600.00	4898	3.32%	330.10	95.45	6209	−0.63%
Average		264.65	15,419			92.19	14,120	−0.27%

and MIP-BasB are able to solve to optimality all 8-product problem instances. MIP-R features an average time of 32.00 CPU s and an average number of nodes in the branch-and-bound search tree equal to 1940. On the other hand, MIP-BasB is 4 times faster than MIP-R with an average time of 7.40 CPU s, while on average explores just 288 nodes.

The computational results for the 16-product problem instances (21–40) can be found in Table 8. Note that rel. gap column presents the relative gap of the solution found by the MIP-R formulation within the predefined time limit. Observe that MIP-R managed to find the optimal schedule in 11 out of 20 problems. Moreover, in 4 other problems a feasible integer solution have been found. High relative gaps (i.e., >5%) are reported for 3 of the 4 integer solutions. The MIP-R formulation did not manage to find an integer solution within the predefined time limit in 5 problems of the 16-product problem set. On the contrary, MIP-BasB reports an integer solution to all problem instances, featuring an average computational time of 148.49 CPU s. Taking into account the complexity of the problems solved, the average computational time of MIP-BasB can be considered short. Also, note that MIP-BasB gives a better solution (than that of the MIP-R) in 7 out of 20 problems. MIP-R reports a better makespan than MIP-BasB only in problem instance 37.

Furthermore, Table 9 presents the computational results for MIP-R and MIP-BasB for problem instances (41–50) considering 24 products. Despite the complexity of the problems considered, both approaches manage to find an integer solution within the predefined time limit for all these problems. Specifically, MIP-R gives the optimal solution in 6 of 10 problems, while in the remaining problems integer solutions with acceptable relative gaps (i.e., <5%) are observed. MIP-BasB reports equal or better solutions in all problem instances of the 24-product problem set. More specifically, better solutions than those of the MIP-R have been reported in 3 problems. Furthermore, MIP-BasB method, with an average computational time equal to 92.19 CPU s, is approximately 3 times faster than the MIP-R model in this problem set.

In total, MIP-R reaches an integer solution in 90% of the 50 problems solved, while the MIP-BasB method finds a feasible solution (and in most of cases the optimal) for all problem instances with a low average computational time equal to 80.79 CPU s. MIP-R obtains the optimal solution in 74% of the total set of problems solved. In general, the MIP-BasB method performance compares favorably with the results of the MIP-R formulation. Nevertheless, both approaches have been proven efficient considering the high complexity of the scheduling problems addressed.

Finally, as a general but interesting observation, it seems that the MIP-BasB method reports such good quality solutions (and in some cases gives better solutions than those of the MIP model within the predefined time limit) due to the fact that the problem considered is highly constraint and features a low degree of flexibility regarding allocation and sequencing decisions.

8. Concluding remarks

In this work, the production scheduling problem in a real-world multistage food processing industry is considered. First, an efficient MIP continuous-time model (MIP-R) is presented relying on the integrated modeling of all production stages by introduction several sets of strong valid integer cuts regarding allocation and sequencing decisions. A MIP-based decomposition method (MIP-BasB) is also proposed for coping with large-scale production scheduling problems. Both solution approaches feature a salient computational performance (especially the MIP-BasB method) for 50 industrial-scale problem instances solved.

It should be mentioned that for even more complex scheduling problems, the MIP-R performance will likely be rendered due to big model sizes and high memory requirements, while the performance of the MIP-BasB method is expected to be more stable, since the user can define the degree of decomposition and thus controlling the computational performance of the method. Specifically, in the MIP-BasB method, the user could maintain a low number of inserted batches per iteration in an attempt to favor lower computational times. In addition, the second stage of the MIP-BasB method (i.e., re-optimization of the process line) could be slightly modified (relaxed) by applying partial re-sequencing (similar to those of Kopanos, Méndez, and Puigjaner (2010), Méndez and Cerdá (2003), Röslof, Harjunkoski, Björkqvist, Karlsson, and Westerlund (2001)) among batches rather than the complete re-sequencing tactic followed in this work. Finally, the comparison with other approaches, such as the timed automata framework, will be of particular interest.

Acknowledgements

Financial support from the Spanish Ministry of Education (FPU Grant to GMK) and projects DPI-2009-09386, D/030927/10, and 2009 SGR 861 is gratefully acknowledged. The authors would also like to thank Prof. PMM Bongers from the Unilever R&D Vlaardingen Research Center for his valuable comments and suggestions.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2011.12.015.

References

Amorim, P., Antunes, C. H., & Almada-Lobo, B. (2011). Multi-objective lot-sizing and scheduling dealing with perishability issues. *Industrial & Engineering Chemistry Research*, *50*(6), 3371–3381.

Bongers, P. M. M., & Bakker, B. H. (2006). Application of multi-stage scheduling. In W. Marquardt, & C. C. Pantelides (Eds.), *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering. Vol. 21 of Computer Aided Chemical Engineering* (pp. 1917–1922). Elsevier.

Entrup, M. L., Günther, H.-O., Van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-integer linear programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, *43*(23), 5071–5100.

Gellert, T., Höhn, W., & Moehring, R. H. (2011). Sequencing and scheduling for filling lines in dairy production. *Optimization Letters*, *5*(3), 491–504.

Kallrath, J. (2002). Planning and scheduling in the process industry. *OR Spectrum*, *24*(3), 219–250.

Kopanos, G. M., Méndez, C. A., & Puigjaner, L. (2010). MIP-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry. *European Journal of Operational Research*, *207*(2), 644–655.

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2010). Optimal production scheduling and lot-sizing in dairy plants: The yogurt production line. *Industrial Engineering & Chemistry Research*, *49*(2), 701–718.

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2011a). Production scheduling in multiproduct multistage semicontinuous food processes. *Industrial Engineering & Chemistry Research*, *50*(10), 6316–6324.

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2011b). Resource-constrained production planning in semicontinuous food industries. *Computers & Chemical Engineering*, *35*(12), 2929–2944.

Maravelias, C. T., & Sung, C. (2009). Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering*, *33*(12), 1919–1930.

Marinelli, F., Nenni, M. E., & Sforza, A. (2007). Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company. *Annals of Operations Research*, *150*(1), 177–192.

Méndez, C. A., & Cerdá, J. (2003). Dynamic scheduling in multiproduct batch plants. *Computers & Chemical Engineering*, *27*(8-9), 1247–1259.

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). Review: State-of-the-art of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, *30*(6–7), 913–946.

Rosenthal, R. E. (2010). *GAMS – A User's Guide*. Washington, DC, USA: GAMS Development Corporation.

Röslof, J., Harjunkoski, I., Björkqvist, J., Karlsson, S., & Westerlund, T. (2001). An MILP-based reordering algorithm for complex industrial scheduling and rescheduling. *Computers & Chemical Engineering*, *25*(4–6), 821–828.

Subbiah, S., Schoppmeyer, C., & Engell, S. (2011). An intuitive and efficient approach to process scheduling with sequence-dependent changeovers using timed automata models. *Industrial & Engineering Chemistry Research*, *50*(9), 5131–5152.