Computers and Chemical Engineering 94 (2016) 141–156

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Multi-week MILP scheduling for an ice cream processing facility

Ezra Wari*, Weihang Zhu

Department of Industrial Engineering, Lamar University, Beaumont, TX 77710-0032, United States

ARTICLE INFO

Article history:
Received 1 May 2016
Received in revised form 6 July 2016
Accepted 28 July 2016
Available online 5 August 2016

Keywords:
MILP
Production scheduling
Food processing

ABSTRACT

This paper presents a multi-week mixed integer linear programming (MILP) scheduling model for an ice cream processing facility. The ice cream processing is a typical complex food manufacturing process and a simplified version of this processing has been adapted to investigate scheduling problems in the literature. Most of these models only considered the production scheduling for a week. In this paper, multi-week production scheduling is considered. The problem has been implemented as an MILP model. The model has been tested on a set of cases from the literature, and its results were compared to the results of problems solved using hybrid MILP-heuristics methods in the literature. The inclusion of clean-up session, weekend break and semi-processed product from previous week were also assessed with two additional sets of experiments. The experiments result show that the proposed MILP is able to handle multi-week scheduling efficiently and effectively within a reasonable time limit.

1. Introduction

This paper presents a model and its solution to a multi-week scheduling problem for an ice cream facility in the food processing industry. Optimization of scheduling problems in the food processing industry can be done using heuristics methods, exact/mathematical methods or the combination of the two. Exact/Mathematical methods can lead to optimal solutions, but they are slow and in the worst cases, unable to solve a complex optimization problem. As a solution-search approach, heuristics/meta-heuristics can substitute or supplement the exact/mathematical methods. However heuristic/meta-heuristics cannot guarantee finding the optimal solution. In a highly competitive environment, optimal scheduling results can provide several benefits including insights on the full potential of a production system or inputs for decision making on expansion projects.

The single week ice cream production scheduling problem has been modeled and solved with hybrid exact-heuristic solution approaches in the literature (Bongers and Bakker, 2006; Kopanos et al., 2012). They adopted Mixed Integer Linear Programming (MILP) and problem-specific heuristics or manual adjustments to optimize a makespan objective. This paper considers multi-week scheduling for this problem with a pure MILP model. More realistic constraints for production such as clean-up sessions, weekend breaks and semi-processed products are incorporated into the model.

Meta-heuristics approaches used in solving scheduling problems include Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Tabu Search (TS) and Artificial Bee Colony (ABC). Hecker et al. (2013) combined ACO and PSO algorithm to solve a no-wait hybrid flow shop scheduling problem. This approach searched different scheduling horizon for jobs using the algorithm to fit the objective – minimizing the makespan and idle time for a bakery. GA is one of the most frequently adopted meta-heuristic approaches. Earlier publications such as those by Shaw et al. and Heinonen and Pettersson demonstrated that GA could attain better optimization result when compared to mathematical programmming (Shaw et al., 2000; Heinonen and Pettersson, 2003). Karray et al. (2011) compared two GA approaches, i.e., the conventional GA approach and parallel GA approach to solve a single machine scheduling problem. The problem model optimized the makespan and cost functions for freshness (perishability) and distribution discount. Processing orders of jobs were coded as genetic codes and the better solution were searched by combining different permutation of the genes (sequence of product). Gellert, Höhn and Möhring adopted GA for sequencing and filling problem in dairy processing (Gellert et al., 2011). The sequencing problem with specific attributes such as clean-up/sanitization, equipment capacity, and processing time were formulated and optimized using GA. Zhou and Gu (2009) proposed a combined GA and Game evolution approach to a no-wait flow shop scheduling problem. Their approach evaluated

* Corresponding author.
E-mail addresses: humorstar@gmail.com, humorstar@yahoo.com (E. Wari).

http://dx.doi.org/10.1016/j.compchemeng.2016.07.025
0098-1354/© 2016 Elsevier Ltd. All rights reserved.

Nomenclature

Sets

$b, b', b'' \in B$ batches of products
$i, i' \in I$ product types
$j \in J$ processing units
$s \in S$ processing stage
$Itransfer \in I$ product types transferred from preceding week

Subsets

I_j product i processed in unit j
$ISuc_i$ immediate successor of product i
$ISucc_i$ successors of product i
$IPred_i$ predecessor of product i
I_i^{SP} products that share the same packaging line with product i
J_i units j that process product i
$J2_i$ units j that process product i in the second stage
$Mach_s$ units j that process in stage s
J_{ij}^{last} last unit j that process product i

Parameters

$\alpha_j^{\min}$ minimum number of products assigned to packaging line j
$\beta_i^{\min}$ minimum number of batches for product i
$\beta Itransfer_i^{\min}$ minimum number of batches transferred from preceding week for product i
$\gamma_{ii'j}$ sequence-dependent change-over time between product i and i' in units j
$\gamma_j^{\min}$ minimum sequence-dependent change-over time in packaging units j
$IdleGamma_i$ sequence-dependent change over time from product to idle state
ε_i^{life} shelf life for product i in processing
ζ_i demand for product i
θ_i priority of product i in the packaging unit j
ω long production horizon (big M value)
λ_i total number of aging vessels for product i
μ_j^{max} maximum capacity of aging vessel j
ρ_{ij} processing rate of product i in process line or packaging line j
τ_i^{ag} minimum aging time for product i
τ_i^{empty} emptying time in aging vessel for product i
τ_i^{fill} filling time in aging vessel for product i
ϕ_j^{min} minimum wait time to begin packaging in line j
$Workweeklength$ available production horizon
$WeekNumber$ processing week

Parameter function

$\beta_i^{min} = \frac{\zeta_i}{\mu_j^{max}}$ where $i \in I, j \in J2_i$

$\beta Itransfer_i^{min} = \frac{\zeta_i}{\mu_j^{max}}$ where $i \in Itransfer, j \in J2_i$

$\tau_i^{empty} = \frac{\mu_j^{max}}{\rho_{ij}}$ where $i \in I, j \in Mach[s] : s = 3$

$\tau_i^{fill} = \frac{\mu_j^{max}}{\rho_{ij}}$ where $i \in I, j \in Mach[s] : s = 1$

$\omega = 1.2\left(\phi_j^{\min} + \left(\alpha_j^{\min} - 1\right)\gamma_j + \min_j\left(\sum_{i \in I_j}\tau_i^{empty}\beta_i^{\min}\right)\right)$ where $j \in Mach[s] : s = 3$

Decision variables

C_{ibs} completion time for stage s of batch b of product i
C_{0ibs} completion time for stage s of transferred batch b of product i
C_{max} makespan
$CWeek_{ibs}$ processing completion week for stage s batch b of product i
L_{ibs} starting time for stage s of batch b of product i
L_{0ibs} starting time for stage s of transferred batch b of product i
W_{ibs} waiting (standing) time for stage s of batch b of product i
W_{0ibs} waiting (standing) time for stage s of transferred batch b of product i
$\bar{X}_{ibi'b'}$ (binary) 1 if batch b of product i processed before batch b' of product i'
Y_{ibsj} (binary) 1 if batch b of product i in stage s is processed in unit j

the solutions based on completion time and deviation from the due dates. The problem model included sequence, no-wait, resource and due date as constraints and the summation of weighted penalized time as an objective. Samarghandi and ElMekkawy proposed an approach where coded schedules were searched for local optima using PSO, followed by TS to search for global optima (Samarghandi and ElMekkawy, 2012). Samarghandi developed a PSO optimization model for a makespan objective with due date constraint. These PSO approaches modeled a no-wait flow shop scheduling problem (Samarghandi, 2015). Banerjee et al. proposed ABC for the optimization of a scheduling problem with multiple objectives (cost and robustness) in a milk processing industry (Banerjee et al., 2008). This approach attained solutions for different cost values and risk levels due to the uncertainty factors of the optimization.

MILP models have frequently been used to solve optimization problems in the food processing industry. Doganis and Sarimveis (2008a) developed an MILP model for a yogurt processing facility, with five cost objective functions (setup, the storage, machine utilization, labor overtime, and the freshness of products) and constraints for daily production limit, machine/product assignment, process time window, and sequence dependent processing. The authors have also demonstrated that a similar approach can be applied to optimize combined makespan and costs function objectives (changeover, inventory, machine utilization, and overtime) (Doganis and Sarimveis, 2008b; Doganis and Sarimveis, 2007). Sadi-Nezhad and Darian (2010) proposed an MILP model for optimization of production cost (inventory, setup and unmet demand) for a juice processing facility. The model considered several constraints such as production capacity, machine-job assignment, inventory, and completion time. To optimize the profit as a fucntion of revenue and cost, Liu et al. (2010) adopted a similar MILP model for scheduling a processing stage in edible oil manufacturing (deoderizing stage). The model was formulated for two cases (objectives with backlog and without a backlog) with assignment and sequencing, batch size, inventory, and demand constraints.

MILP scheduling problems can also be formulated with the integration of other production planning features such as lot-sizes. To minimize the different costs (inventory, operating, fermentation recipe preparation, machine utilization and changeover), Kopanos et al. (2009) formulated an MILP model for scheduling and determining inventory level and packaging machine/line capacity for different products for a yogurt manufacturing facility. The model included constraints for processing time, sequencing, machine allocation, mass balance (demand and supply), and processing (fermentation). Upper/lower bound and tightening constraints were used to accelerate the optimization run. Marinelli et al. (2007) proposed mixed mathematical and heurstic methods to optimize similar scheduling problem for a yogurt processing facility. The problem was first modeled by defining the objective as a function of setup, storage and process cost, and constraints as functions of tank (machine) capacity, setup/clean-up requirement, and balancing demands. Then MILP was used to optimize lot-sizes and schedules as separate sub-processes, and a heuristic method was used to search the best feasible solution for the two sub-processes. For a production planning and scheduling problem with product shelf life, Lütke Entrup et al. developed an MILP model for yogurt processing with an objective of maximizing profit (Lütke Entrup et al., 2005). The model considered constraints including batch and machine (fermentation) capacity size, line utilization, sequencing, line utilization, setup and output-demand. Bilgen and Dogan presented a MILP model for the production planning of multi-stage dairy processing (Bilgen and Dogan, 2015). The model optimized profit which was defined as the difference of total sales and cost. Total cost included the cost of production, inventory, lost sales, and setup. Model constraints comprised of functions for limits of production equipment, and supply, lot-size, demand, and flow balance. Bilgen and Celebi proposed a hybrid approach by using MILP to formulate a combined scheduling-distribution problem and solved it using iterative simulation for a dairy processing problem (Bilgen and Çelebi, 2013). The objective for the model was set to maximize the benefit attained from increased shelf life and reduced cost such as inventory, processing, setup, overtime, backlogging, and transportation.

The rest of the paper is organized as follows. Section 2 discusses the background and description of the problem model. The proposed solution is described in Section 3. Section 4 presents the results attained for the optimization of the problem instances and the discussion on these results. The paper is concluded with remarks in Section 5.

2. Problem background and formulation

The scheduling case study considered in this paper is a medium-size ice cream processing enterprise first presented by Bongers and Bakker (2006). In the facility, production starts by transferring ice cream ingredients from a local warehouse to a mixing department. Following mixing of ingredients in accordance to receipts, product mixes are pasteurized (two equipment) and moved to the aging vessels. These vessels age the product mixes for the minimum amount of time in each receipt. Product mixes are then packaged into different sizes and shapes, and moved to the hardening tunnel for further freezing.

Bongers and Bakker (2006) could only develop an MILP scheduling model with manual adjustment for part of the real ice cream production process due to the resulting highly complex scheduling problem which could not be optimized by the software used – INFOR. They also reduced the number of product mixes for the same reason. The formulated scheduling model consisted of a makespan objective for a three-stage processing of the ice cream product mixes. The first stage had one unit for the pasteurization process, followed by six aging vessels (also called tanks) and finally two packaging lines. The pasteurizer fed aging vessels at a constant rate of 4500 kg/h. Product mixes are aged in the vessels/tanks according to their recipes and were moved to the next stage for packaging. These rates for processing/packaging and the minimum aging period for eight product mixes were given in Fig. 1. Bongers and Bakker reported a makespan value of 120 h for single week production horizon (70 batches of products), and the model improved the existing schedule by 10–30%.

Kopanos, Puigjaner and Georgiadis published two papers on the ice cream processing scheduling problem discussed above. These papers optimized a makespan objective with multiple constraints to allocate resources, sequence processes and define the start and completion of processing time. In the first paper, the authors presented a combined MILP-heuristic model to schedule 8 products mixes for a single week production horizon (Kopanos et al., 2011). The MILP part of the model formulated the different functions of the constraints and objective whereas the heuristics part determined which constraint to be included in the cases of feasible and infeasible optimization run. Overall, the model consisted of 18 constraints and was solved using GAMS modeler and CPLEX solver. Ten different product mixes size combinations for a total demand batch size ranging between 62 and 71 were used to test the model. In the second paper, the authors modified the model in the first paper to optimize a larger number of product mixes and batches, for multi-week production horizon (Kopanos et al., 2012). The assignment of product mixes to vessels, arrangements of processing orders, and calculation of completion time need to be computed simultaneously in their first model. This made the application of the first model to large-size problem instance impractical as it took a long time to solve. Therefore, they broke down the assignment of a product mix to vessels as a separate step in their constraint formulation and

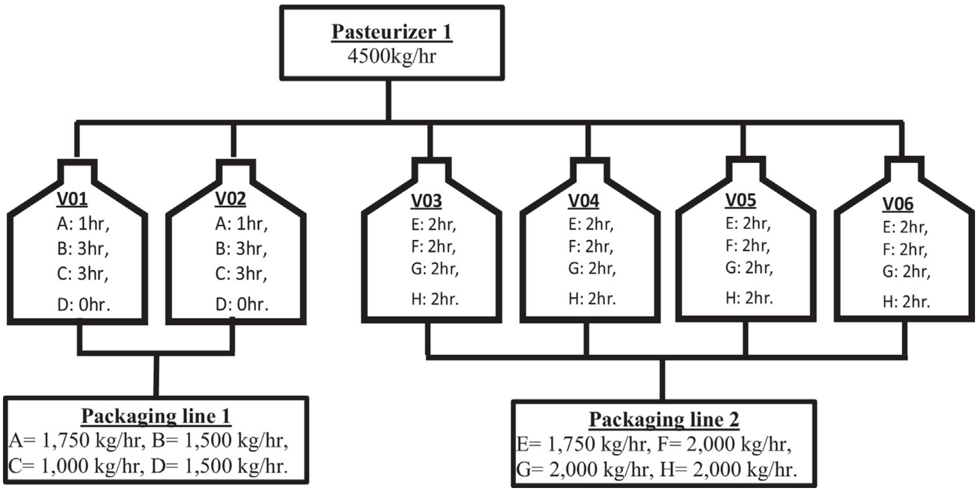


Fig. 1. Three-stage ice cream processing scheduling problem model.

Source: Kopanos et al. (2012).

speeded up the optimization calculation. A different heuristics was integrated into the MILP to set product mix priorities for processing and speeding up optimization in this second model. Product priorities were defined at the beginning of the run and optimization priorities limited a maximum of 25 batches to be optimized at a time. The model was tested using different demand size combinations where the variation was both in product mix and batch size (3 sets of product mix sizes with 70–180 total batch sizes per problem instances, as detailed Appendix 3). The optimization run time were restricted to 10 min (600 s CPU time) in both models. Furthermore, transferred products (two products mixes with a total of four batches) from previous week were also considered in part of their optimization runs. Processing of these products batches started in the preceding week and only need to complete the last stage. In other words, these products need to be scheduled for the packaging stage for the current week. The two publications reported an optimal makespan value of 118.55 h for the same product mixes and batch sizes, comparable to that of Bongers and Bakker (2006).

At the end of each production week, the down-sized ice cream processing schedules need to add clean-up sessions since the production breaks over the weekends. Kopanos, Puigjaner and Georgiadis incorporated these sessions by simply adding them up at the end of the optimization run. As it will be shown later in this paper, simple addition of extra hours at the end of the makespan value may not necessarily lead to the same optimization results as to scheduling the clean-up sessions because of the 120 h per week constraint. The impact of these sessions gets more significant as the number of scheduled weeks increase, which is the result from an increase in the product mix numbers and batch sizes. The MILP models in Bongers and Bakker (2006), Kopanos et al. (2011) and Kopanos et al. (2012) were combined with manual manipulation or heuristics to speed up the optimization. The proposed model in this paper, which is based on the second model in Kopanos et al. (2012), employs only MILP method to attain optimal solutions.

3. Proposed multi-week scheduling model and solution

The ice cream scheduling model by Kopanos et al. (2012) consisted of 30 constraints. These constraints first formulated a sequential start and completion processing time for all stages, and assigned aging vessel to each batches of products. Then they rearranged the batches in accordance to the different processing priorities of products. This paper modified this approach by integrating the products mixes from the preceding week into the optimization run, scheduling the clean-up sessions and determining proper breaking of product processing in consecutive weeks when scheduling extends over multiple weeks.

The improved constraints can be grouped into four general categories (refer Appendix 1 for the complete model). The first group of constraints, constraints (A1)–(A11), create the process intervals (the start and completion time) for each product batch at all stages and then arrange them serially based on the batch number (Fig. 2). Constraints for product mixes scheduled to be processed in all three stage

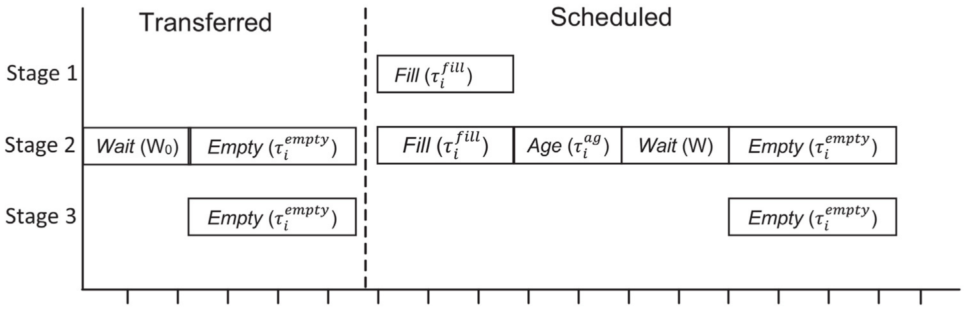


Fig. 2. Processing intervals for transferred and scheduled products.

Source: modified from Kopanos et al. (2012).

in the current week, i.e., constraints (A1)–(A7) (refer Appendix 1), are inherited into this paper from Kopanos et al. (2012). This paper adds constraints (A8)–(A11) (refer Appendix 1) to this group to create intervals for semi-processed product batches transferred from the preceding week to the current week. These transferred ice cream products go through only packaging stage but would be stored in the vessels until their respective packaging line is freed up. Therefore the interval would have only a waiting and emptying time and be arranged serially if more than one batches were transferred. Completion time for transferred products in the second and third stages are calculated in constraints (8) and (9), respectively.

$$C_{oibs} = W_{0ibs} + \tau_i^{empty} \quad \forall i, \quad b \leq \beta Itransfer_i^{\min}, \quad s = 2 : i \ \text{ in } \ Itransfer \tag{8}$$

$$C_{oibs} = L_{0ibs} + \tau_i^{empty} \quad \forall i, \quad b \leq \beta Itransfer_i^{\min}, \quad s = 3 : i \ \text{ in } \ Itransfer \tag{9}$$

The completion time for transferred products in the second and third stage are set to be equal since the third stage is a continuous process (constraint (10)).

$$C_{0ibs} = C_{0ibs-1} \quad \forall i, \quad b \leq \beta Itransfer_i^{\min}, \quad s = 3 : i \ \text{ in } \ Itransfer \tag{10}$$

Sequential arrangement of transferred products batches are made using constraint (11).

$$C_{0ibs} = L_{0ib+1s} \quad \forall i, \quad b \leq \beta Itransfer_i^{\min} - 1, \quad s = 3 : i \ \text{ in } \ Itransfer \tag{11}$$

The second group of constraints, allocate the vessels to every batch in each product (Kopanos et al., 2012). The allocation rule was to assign each batch of product to the group of vessels in a serial and cyclic (when reaching the last vessel) manner. Although the rule was not changed, the constraints from Kopanos, Puigjaner and Georgiadis are replaced with a new set to accommodate products transferred from preceding week in the model. Besides, a feature in CPLEX may be utilized to automatically incorporate the cyclic roles of the constraints in this group. Hence it becomes unnecessary to keep these constraints. With the same objective of allocating products to vessels, the new set of constraints, i.e., constraints (12)–(20), have been formulated and adopted for the multi-week scheduling in this paper. As shown below, it starts by allocating the one vessel in the group to all batches and then assign the first vessel in the respective group to the transferred product (constraints (12) and (13)).

$$\sum Y_{ibsj} = 1 \quad \forall i, \quad i \in I, \quad b \leq \beta_i^{\min}, \quad s = 2, \tag{12}$$

$$Y_{ibsj} = 1 \quad \forall i \quad i \in Itransfer, \quad b = 1, \quad s = 2, \quad j = first(J2_i) \tag{13}$$

Transferred ice cream products from the preceding week are rearranged serially in constraints (14) and (15).

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in Itransfer, \quad b \leq \beta Itransfer_i^{\min} - 2, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} \leq \beta_i^{\min} \tag{14}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in Itransfer, \quad b \leq \beta_i^{\min} - 1, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} > \beta_i^{\min} \tag{15}$$

The last batch of the transferred product and the first batch of the current week are arranged consecutively so that vessel's assignments are continuous (constraints (16)–(18)).

$$\begin{aligned} &Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, \quad i \in Itransfer, \quad b = \min_i(\beta Itransfer_i^{\min} - 1, \beta_i^{\min}), \\ &i' \in Last(IPred_i), \quad b' = \beta Itransfer_{i'}^{\min}, \quad s = 2, \quad j \in J2_i : card(IPred_i) > 0 \end{aligned} \tag{16}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in Itransfer, \quad b = \beta Itransfer_i^{\min} - 1, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} \leq \beta_i^{\min}, card(IPred_i) = 0 \tag{17}$$

$$\begin{aligned} &Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, \quad i \in Itransfer, \quad b = \beta Itransfer_i^{\min} - 1, \quad i' \in ISuc_i, \\ &b' = \beta Itransfer_{i'}^{\min}, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} - 1 > \beta_i^{\min}, card(IPred_i) = 0 \end{aligned} \tag{18}$$

The products scheduled in the current and succeeding weeks are rearranged serially and cyclically until all products are assigned (constraints (19) and (20)). Generally, this second group of constraints starts assigning vessels to products from the preceding week and moves on to the products processed at present and future weeks.

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, \quad i \in I, \quad b = \beta_i^{\min}, \quad i^{'} \in ISuc_i, \quad b' = \beta Itransfer_i^{\min}, \quad s = 2, \quad j \in J2_i \tag{19}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in I, \quad \beta Itransfer_i^{\min} \leq b \leq \beta_i^{\min}, \quad s = 2, \quad j \in J2_i \tag{20}$$

The third group of constraints arranges products that are assigned in the same equipment in all the processing stages. The start time and completion time of processing each batch of products are organized using product priority and the changeover time between products in constraints (21)–(26) (Kopanos et al., 2012). The arrangement in this group also accounts for process completion time of the prior stages. For the multi-week scheduling, transferred products from the preceding week are required to be integrated into the schedule. Therefore, constraints for adding the waiting and packaging time of these products are appended to this third group as constraints (27) and (28).

$$\begin{aligned} &L_{i'b's} = C_{0ibs} + \gamma_{ii'} - \omega(2 - Y_{ibsj} - Y_{i'b'sj}) \quad \forall i, \quad i \in Itransfer, \quad b \leq \beta transfer_i^{\min} - 1, \quad i' \in 1, \beta transfer_{i'}^{\min} \leq b' \leq \beta_{i'}^{\min}, \quad s = 2, \\ &j \in J_i \cap Mach[s] : i \in Itransfer, \beta Itransfer_i^{\min} > 1 \end{aligned} \tag{27}$$

$$L_{i'b's} = C_{0ibs} + \gamma_{ii'} \quad \forall i, \quad i \in Itransfer, \quad b \le \beta Itransfer_i^{\min} - 1, \quad i' \in I, \beta Itransfer_{i'}^{\min} \le b' \le \beta_{i'}^{\min},$$
$$s = 3, \quad j \in J_i \cap Mach[s] : i \in Itransfer \tag{28}$$

The last group consists of miscellaneous constraints, which include the makespan, lower bound, and the domain set for decision variables constraints (constraints (A29)–(A34), refer Appendix 1). Except for the minor modification made to the lower bound (replace sum of emptying rate by the minimum of the sum of filling rate for each packaging line), the constraints in this group are inherited (Kopanos et al., 2012). The modification made here to constraint 30 dropped down the lower bound to ensure the inclusion of potential solution in the optimization.

$$C_{\max} = \phi_j^{\min} + \left(\alpha_j^{\min} - 1\right)\gamma_j + \min_j \sum\nolimits_{i \in I_j} \tau_i^{fill} \beta_i^{\min} \quad \forall j \in Mach[s] : s = 3 \tag{30}$$

Earlier attempts to solve this ice cream production scheduling problem aimed for scheduling within one week (120 h in 5 continuous working days). Hence the problem instances did not need to address the issue of scheduling transfers of product batches between consecutive production weeks. In Kopanos et al. (2012), problem instances required schedules extending over multiple production weeks. A clean-up time needs to be added in these schedules and proper break-up points have to be set so that processes would not be started unless they could be completed before the clean-up. Bongers and Bakker (2006) defined the changeover time from/to idle state in the weekends to be 2 h, and this time is taken as the clean-up time. Since clean-ups are assumed to be done at the end of each production week, no clean-up or idle changeover time was considered at the beginning of the production week in this paper's model. It is important to note that this break and clean-up sessions do not apply to vessels scheduled to age and store product mixes over the weekends.

Four new constraints are formulated to shift processing schedules that cannot be completed before the clean-up session for stages 1 and 3 of the ice cream production process. These constraints first check whether the processing starts in one week and completes in the next, and then shift the start of processing to the subsequent week. Product mixes that had been already aging or completed the aging process but could not finish packaging were kept in the vessel over the weekends. Constraints (35) and (36) shift the starting of the processing of product batch in stages 3 and 1, respectively. Constraints (37) and (38), on the other hand, restrained the processing of the products from extending over two successive weeks.

$$L_{ibs} \le (n * Workweeklength) \&\& C_{ibs} \ge ((n * Workweeklength) - IdleGamma_i)$$
$$\Rightarrow L_{ibs} \ge (n)) * Workweeklength \quad \forall i, \quad j \in Mach[s], \quad i \in I, \beta Itransfer_i^{\min} \le b \le \beta_{i'}^{\min}, \quad s = 3, \quad n \in WeekNumber \tag{35}$$

$$L_{ibs} \ge ((n * Workweeklength) - IdleGamma_i) \&\& L_{ibs} \le (n * Workweeklength) > L_{ibs}$$
$$\ge (n)) * Workweeklength \quad \forall i, j \in Mach[s], \quad i \in I, \beta Itransfer_i^{\min} \le b \le \beta_{i'}^{\min}, \quad s = 3, \quad n \in WeekNumber \tag{36}$$

$$CWeek_{ibs} \le (C_{ibs}/(Workweeklength - IdleGamma_i)) + 1 \quad \forall i, \quad i \in I, \quad j \in J, \quad s \in S \tag{37}$$

$$CWeek_{ibs} \ge C_{ibs}/(Workweeklength - IdleGamma_i) \quad \forall i, \quad i \in I, \quad j \in J, \quad s \in S \tag{38}$$

4. Experiment result and discussion

Three groups of optimization experiments were conducted using the proposed MILP model. These experiments started by testing the model's capacity to solve a problem with a similar parameter setup to those used by Kopanos et al. (2012). Then the main optimization experiments which integrated weekend break with clean-up session and products from preceding week were conducted. The model was formulated and optimized using IBM ILOG CPLEX 12.6 optimization software (academic version) with a run setup given in Appendix 2. Computation was carried out on an Alien-ware Area 51 Workstation with Intel® Core™ i7-5820K Processor CPU, 3.6TB Hard Drive and 32 GB RAM, on Windows 8.1 Professional operating systems.

4.1. The first set of experiment: test the model under parameter conditions from the literature

Experimental conditions by Kopanos et al. (2012) did not consist of product mix transferred from preceding week to the next week. It also did not schedule the clean-up sessions, but rather added 2 h of clean-up time to the final makespan values. To replicate this scheduling problem setup, the variable $\beta Itransfer_i^{\min}$ was set to 1 for all $i \in I$, constraints (35)–(38) were relaxed in the optimization run and 2 h were added to the makespan values attained at the end of each of problem instance run. All three problem instances sets (see Appendix 3 for these problem sets and Appendix 4 for related data) from Kopanos et al. (2012) were used to test the model, and the results attended are given in Table 1. The optimization run time limit of 10 min (600 s) is also observed in this set of experiment. The results published by Kopanos et al. (2012) using only the MILP method and combined MILP/heuristics approaches are also included in the table. A column presenting the difference in the makespan result attained between the MILP/heuristics and the proposed MILP model in this paper is added at the end (given as percentage of the deviation of the MILP results from the published results of the MILP/heuristics approach).

The proposed MILP solved all the problem instances in all problem sets. These results are compared to the results of the mixed MILP/heuristics (MILP/H column) model published by Kopanos et al. (2012). Overall, the proposed model attained result within −0.63% to 18.82% of the published results for all 50 problem instances. Positive deviation implies that the model attained a better result. As a mixed heuristics approach, the published results are not expected to give optimal results in all cases. However negative percent deviation indicate that the proposed MILP model did not attain the optimal value. A close examination of these variations indicate that all values are less than 0.63% exhibited in only 7 problem instances. The cause for these deviation may be attributed to rounding and decimal point

Table 1
The makespan (C_{max} in hours) results for problem conditions from literature.

(a) Problem set 1

Problem instance	Batch size	Published results		Proposed model results		Difference in C_{max}
		C_{max} MILP	C_{max} MILP/H	C_{max}	Optimization run time (in s)	
1	70	120.33	120.33	120.33	2.16	0.00%
2	75	118.17	118.17	118.18	2.80	−0.01%
3	80	131.48	131.48	131.21	2.68	0.21%
4	85	142.1	142.1	142.10	2.36	0.00%
5	90	149.66	149.66	149.66	2.76	0.00%
6	95	152.34	152.34	152.34	3.79	0.00%
7	100	161.47	161.47	161.19	3.65	0.17%
8	105	171.37	171.37	171.37	4.19	0.00%
9	110	175.82	175.82	175.82	3.98	0.00%
10	115	187.75	187.75	187.75	7.40	0.00%
11	120	191.25	191.25	191.25	4.00	0.00%
12	125	206.42	206.42	206.42	6.13	0.00%
13	130	201.76	201.76	201.87	6.18	−0.06%
14	135	223.56	223.56	223.56	6.43	0.00%
15	140	224.71	224.71	224.71	5.99	0.00%
16	145	222.06	222.06	222.06	6.37	0.00%
17	150	238.04	238.04	238.04	5.67	0.00%
18	160	251.49	251.49	251.52	9.58	−0.01%
19	170	260.52	260.52	260.54	11.47	−0.01%
20	180	291.75	291.75	291.75	14.79	0.00%
				Average	5.62	

(b) Problem set 2

Problem instance	Batch size	Published results		Proposed model results		Difference in C_{max}
		C_{max} MILP	C_{max} MILP/H	C_{max}	Optimization run time (in s)	
21	70	119.83	119.83	115.77	4.17	3.39%
22	75	121.62	121.62	121.62	2.73	0.00%
23	80	127.25	127.25	126.50	3.60	0.59%
24	85	141.14	141.14	141.29	3.19	−0.10%
25	90	147.02	147.02	145.81	4.58	0.82%
26	95	154.94	154.94	154.02	3.68	0.59%
27	100	162.94	162.94	161.02	7.41	1.18%
28	105	181.21	181.21	169.21	9.71	6.62%
29	110	181.23	181.23	180.80	7.14	0.24%
30	115	187.46	187.46	182.07	10.63	2.87%
31	120	190.95	190.95	189.58	6.60	0.72%
32	125	218.26	214.21	199.54	11.39	6.85%
33	130	216.94	210.76	202.99	29.76	3.69%
34	135	–	234.81	206.87	34.11	11.90%
35	140	226.31	226.31	222.89	9.42	1.51%
36	145	–	252.13	223.73	67.25	11.26%
37	150	250	265.3	234.96	10.50	11.44%
38	160	–	298.78	242.54	20.97	18.82%
39	170	–	292.34	267.90	65.74	8.36%
40	180	–	326.58	288.21	71.77	11.75%
				Average	19.22	

(c) Problem set 3

Problem instance	Batch size	Published results		Proposed model results		Difference in C_{max}
		C_{max} MILP	C_{max} MILP/H	C_{max}	Optimization run time (in s)	
41	70	118.98	118.98	119.74	2.07	−0.63%
42	80	136.43	136.43	136.95	1.97	−0.38%
43	90	146.78	146.78	145.96	4.75	0.56%
44	100	164.99	164.99	162.75	4.80	1.36%
45	110	177.05	177.05	175.24	5.76	1.02%
46	120	205.32	205.32	191.65	8.76	6.66%
47	140	222.63	221.66	219.44	7.57	1.00%
48	160	262.51	258.16	255.52	12.38	1.02%
49	180	294.32	294.32	282.15	82.86	4.13%
50	200	332.19	330.1	325.02	24.12	1.54%
				Average	15.50	

Note: Optimization run time are measured in CPU seconds (the measure of processing time a CPU takes to carry out the optimization computation).

Table 2
The comparison of makespan (C_{max} in hours) results for cases when clean-up session are exclude and included.

(a) Problem set 1

Problem instance	Clean-up session not included		Clean-up session included		Difference in C_{max}
	C_{max}	Optimization run time (in s)	C_{max}	Optimization run time (in s)	
1	118.33	2.05	124.57	5.81	6.24
2	116.18	2.76	116.18	7.38	0.00
3	129.21	2.33	133.71	8.58	4.51
4	140.10	2.29	144.12	10.76	4.02
5	147.66	2.41	150.21	13.14	2.56
6	150.34	3.82	154.02	15.45	3.68
7	159.19	3.50	163.36	20.54	4.17
8	169.37	3.79	174.60	23.93	5.22
9	173.82	3.65	177.14	23.74	3.33
10	185.75	6.82	188.81	27.46	3.06
11	189.25	3.65	190.86	29.54	1.61
12	204.42	6.03	210.14	28.48	5.72
13	199.87	6.12	204.43	35.89	4.56
14	221.56	6.32	224.62	38.21	3.06
15	222.71	5.89	228.43	43.91	5.72
16	220.06	6.21	223.00	51.89	2.94
17	236.04	5.49	244.57	58.89	8.53
18	249.52	9.44	256.00	69.09	6.48
19	258.54	11.43	267.43	63.24	8.89
20	289.75	14.98	297.14	79.61	7.40

(b) Problem set 2

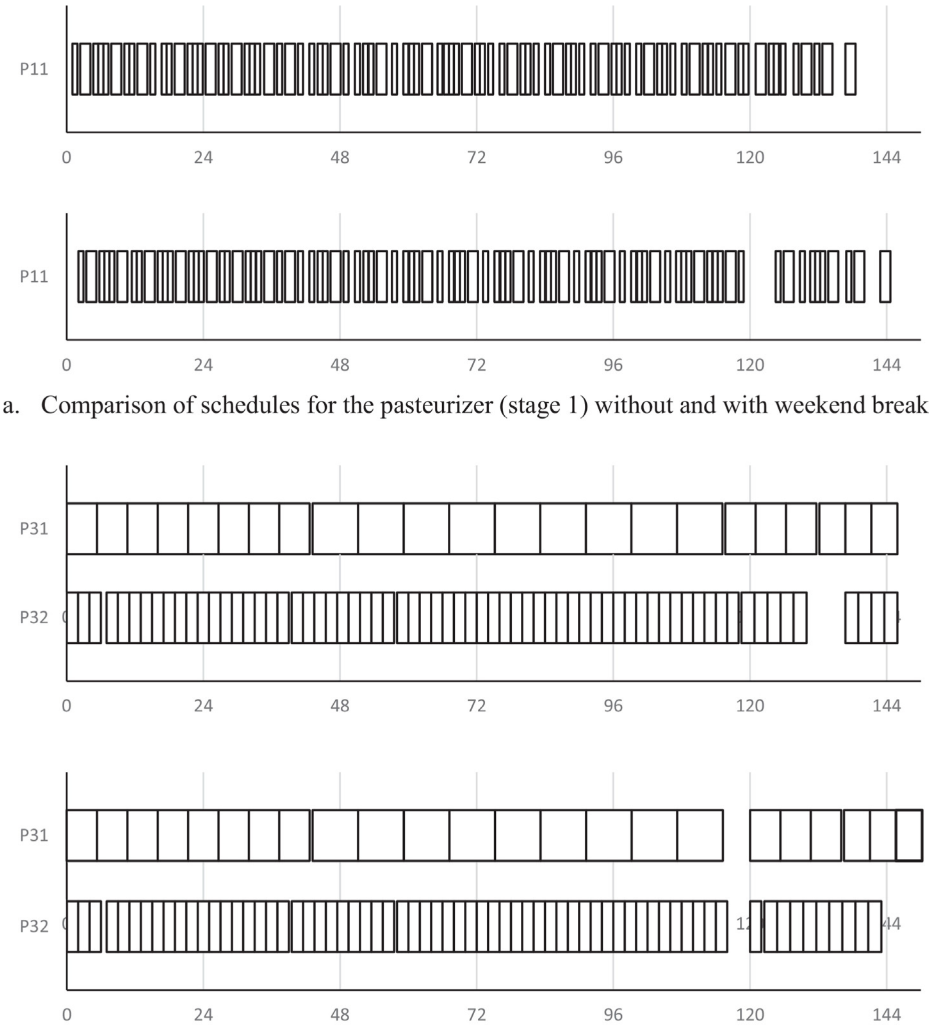
Problem instance	Clean-up session not included		Clean-up session included		Difference in C_{max}
	C_{max}	Optimization run time (in s)	C_{max}	Optimization run time (in s)	
21	113.77	4.39	113.77	6.81	0.00
22	119.62	2.17	124.57	7.98	4.95
23	124.50	3.78	129.14	10.53	4.64
24	139.29	3.02	142.86	12.24	3.57
25	143.81	4.66	148.69	22.74	4.88
26	152.02	3.47	155.55	17.41	3.52
27	159.02	7.23	162.41	31.56	3.39
28	167.21	9.44	170.14	60.11	2.94
29	178.80	5.05	184.64	28.06	5.85
30	180.07	9.86	182.33	51.14	2.26
31	187.58	6.60	191.86	36.64	4.27
32	197.54	11.01	200.93	53.22	3.39
33	201.00	14.79	205.69	60.43	4.69
34	204.87	39.33	207.81	75.66	2.94
35	220.89	9.95	224.14	85.53	3.25
36	221.52	600.00	224.55	600.00	3.03
37	232.96	10.37	235.60	62.53	2.65
38	240.54	19.03	249.07	107.76	8.53
39	265.90	118.06	272.24	156.94	6.33
40	286.21	600.00	296.24	493.13	10.02

(c) Problem set 3

Problem instance	Clean-up session not included		Clean-up session included		Difference in C_{max}
	C_{max}	Optimization run time (in s)	C_{max}	Optimization run time (in s)	
41	117.74	2.21	117.74	5.60	0.00
42	134.93	2.33	139.55	8.76	4.62
43	143.96	5.04	146.91	22.68	2.95
44	160.75	4.76	164.05	20.96	3.30
45	175.24	5.96	176.98	53.79	1.74
46	189.65	8.77	193.24	72.87	3.59
47	217.44	7.86	222.83	53.61	5.39
48	253.52	12.57	258.29	87.73	4.77
49	279.76	600.00	285.19	243.29	5.43
50	323.02	23.89	334.33	210.35	11.31

consideration, CPLEX model variation and/or optimization model variation. Optimization took longer run time as batch and product mix sizes increase with an average run time of 5.62, 19.22 and 15.5 s for each problem set. Run time also rose sharply when compared to the increase in the batch size because additional batches of product mixes increased the total number of variables and constraints the model had to solve.

The results from this first set of the experiments are used as the benchmark for the follow-up experiments considering multi-week constraints in the coming sub-sections.



a. Comparison of schedules for the pasteurizer (stage 1) without and with weekend break

b. Comparison of schedules for the packaging stage (stage 3) without and with weekend break

Fig. 3. Gantt chart for the schedule of stage 1 and 3 for the problem instance 5 (the horizontal axis: working hours).

4.2. The second set of experiment: test the model by including clean-up session

The second set of optimization experiments compared the makespan values of the same problem instances where the cases for products with clean-up sessions were included. Table 2 presents the result of the experiments. The first set of runs did not integrate the clean-up sessions similar to the problems by Kopanos et al. (2012). The week constraints (constraints (31)–(34)) discussed above were then included into the model to test the experimental condition. The makespan result of the first test run (given on the second column of Table 2) were compared against the result without the clean-up constraints (given on the fourth column of Table 2). CPLEX 12.6 optimization software was also used here on the same computer. However additional run set up modification (to global time limit) were made to the CPLEX default setting (refer to Appendix 2).

Inclusion of clean-up sessions made a significant impact on the makespan results for most problem instances as shown in the table above. Adding clean-up sessions at the end of the makespan value may not necessarily give the same result as scheduling these sessions in their respective processing weeks. When scheduling the clean-up, processes that could not be completed were pushed down to the succeeding week. For example, the makespan attained for problem instance 1 (without product from preceding week) was 118.33, which equals the result in first optimization experiment in Table 1 after adding 2 h for clean-up, i.e., 120.33 h to complete the processing with the clean-up. This violates the constraint of 120 working hours per week and thus is not a feasible solution. The proposed model which scheduled the clean-up sessions gave a result of 124.57. Any process beyond the 118th hour has to be scheduled in the coming week if a clean-up session is to be inserted at the end of the week. For problem instance 1, the last process (in this case the packaging process for last batch of product mix A) would have to start at hour 120 which happens to be the starting time for the second week. With processing time of 4.57 h per batch for product mix A, the makespan value for problem instance 1 becomes 124.57 (=120 + 4.57). Fig. 3 below shows the

Table 3
Comparison of makespan (C_{max} in hours) for cases where products from preceding week were excluded or included.

(a) Problem set 1

Problem instance	Total scheduled batch sizes	Transferred batch sizes	Transferred batches not included		Transferred batches included	
			C_{max}	Optimization run time (in s)	C_{max}	Optimization run time (in s)
1	70	4	124.57	5.81	116.55	4.67
2	75	4	116.18	7.39	113.12	4.47
3	80	4	133.71	8.64	129.14	5.10
4	85	4	144.12	10.05	140.57	5.79
5	90	4	150.21	12.97	150.21	6.58
6	95	4	154.02	15.39	154.02	8.45
7	100	4	163.36	20.78	161.14	9.42
8	105	4	174.60	21.97	174.60	11.27
9	110	4	177.14	24.11	173.83	17.40
10	115	4	188.81	27.24	188.81	14.21
11	120	4	190.86	29.05	190.86	15.91
12	125	4	210.14	28.35	210.14	18.59
13	130	4	204.43	35.37	199.74	19.83
14	135	4	224.62	38.58	224.62	21.20
15	140	4	228.43	44.37	228.43	20.83
16	145	4	223.00	49.79	221.07	24.59
17	150	4	244.57	59.56	244.57	25.08
18	160	4	256.00	68.76	253.71	33.22
19	170	4	267.43	66.33	267.43	30.61
20	180	4	297.14	78.94	294.86	33.07

(b) Problem set 2

Problem instance	Total scheduled batch sizes	Transferred batch sizes	Transferred batches not included		Transferred batches included	
			C_{max}	Optimization run time (in s)	C_{max}	Optimization run time (in s)
21	70	4	113.77	6.93	110.20	5.02
22	75	4	124.57	7.96	116.00	4.43
23	80	4	129.14	10.37	124.57	4.77
24	85	4	142.86	12.47	141.07	6.70
25	90	4	148.69	23.05	142.86	14.74
26	95	4	155.55	17.53	150.21	8.29
27	100	4	162.41	32.24	156.57	11.69
28	105	4	170.14	54.83	170.14	21.46
29	110	4	184.64	27.91	177.38	15.20
30	115	4	182.33	51.93	182.33	18.23
31	120	4	191.86	37.48	188.33	38.87
32	125	4	200.93	51.10	200.24	19.85
33	130	4	205.69	60.17	200.36	27.34
34	135	4	207.81	75.87	205.37	37.93
35	140	4	224.14	85.59	223.48	21.97
36	145	4	224.55	600.00	220.09	600.00
37	150	4	235.60	61.82	233.48	27.75
38	160	4	249.07	108.46	242.29	37.10
39	170	4	272.24	156.13	272.24	62.67
40	180	4	296.24	492.68	296.24	600.00

(c) Problem set 3

Problem instance	Total scheduled batch sizes	Transferred batch sizes	Transferred batches not included		Transferred batches included	
			C_{max}	Optimization run time (in s)	C_{max}	Optimization run time (in s)
41	70	4	117.74	5.44	115.66	3.94
42	75	4	139.55	8.88	139.55	5.95
43	80	4	146.91	22.31	146.91	7.89
44	85	4	164.05	20.50	164.05	10.41
45	90	4	176.98	54.01	173.31	12.27
46	95	4	193.24	73.28	191.98	15.74
47	100	4	222.83	53.37	217.50	20.05
48	105	4	258.29	87.68	258.29	56.07
49	110	4	285.19	242.65	285.19	117.82
50	115	4	334.33	210.69	334.33	61.87

schedule for the problem instance 5 without and with the weekend break. The first part, (a), compares the schedule for the pasteurization stage (stage 1) where production ceased around the end of the first week and the start of the second week, due to the clean-up and weekend break. Part (b) depicts the schedules for the two packaging lines (stage 3) where production ceased for weekend break leaving time for clean-up and resumed at the start of second week. In general, the increment to the makespan due to the clean-up could not be predicted accurately since processing time for different products varies and therefore, must be scheduled in the model.

4.3. The third set of experiment: test the model by including transferred products

The third set of experiments integrated the products transferred from previous week. Part of the demand size defined in the problem instances were assigned as the products transferred from the previous week. Therefore, the total product batch sizes defined in ζ_i (demand for product i) were not exceeded. This was set in the model by changing the values of the parameter $\beta Itransfer_i^{\min}$ to the first batch scheduled after the last transferred batch. For the ice cream manufacturing case study in this paper, two product mixes with 3 and 1 batches, respectively were transferred from the preceding week (parameter values $\beta Itransfer_G^{min} = 4$ and $\beta Itransfer_D^{min} = 2$). This experimental run adopted the complete model, i.e., including the week constraints, and Table 3 presents the result of this run. The makespan values in Table 3 column 4 give the optimization run result without the inclusion of transferred products whereas column 6 provides the result of product mixes with the transfers. The experiment adapted the same computational and CPLEX conditions as the ones discussed in the second set of experiment.

Overall the makespan values attained for the cases with transferred batches were either equal to or lower than their respective counter-parts. It is important to note that the transferred batches were considered as part of the total demand requirement of each problem. This would explain the reduction in the makespan value for the cases where product transfers were considered when compared to those without the transfers. The same explanation can also be extended to the reduced optimization run time.

5. Conclusion and future work

This paper proposed an MILP model (without heuristics) for an ice cream multi-week production scheduling problem. The model adopted the makespan objective and a number of constraints including start/stop interval formulation, and processed product arranging constraints. Multi-week constraints were formulated to address the unique arrangements required to transfer products from week to week. This included constraints for integrating products from preceding production week, proper break-up points of products processing for the weekend break, and end of the week clean-up session. The model is implemented in CPLEX and three sets of optimization experiments were conducted. The first set of experiments showed that proposed MILP model was capable of solving problem instances that necessitated the use of heuristics to an MILP model in the past. The second and third sets of experiments compared the run results for cases with transferred products from prior week as well as added weekly clean-up and process break-up points for weekends. Proper break-up of process affected the final makespan value and hence would have to be integrated into the model. In particular, these runs showed that adding the clean-up sessions at the end may not give the same result as scheduling the sessions with proper break-up points.

For future work, several possible extensions can be made to the proposed MILP model to address other scheduling or related problems in the food processing industry. One possible extension to this model would be to optimize scheduling problems with a Tardiness or Lateness objective. Minimizing the difference between the scheduled and due-dates of the product batches can be set as objective for these problems. The other extension would be the inclusion of other constraints in the optimization model. For example, preventive maintenance schedule, or machine breakdown can be incorporated either by changing the data source (by changing the working week) or formulating constraints to this effect. Working hours for a given week can be adjusted for holidays or overtime in a similar manner. Cost function, such as those discussed in the introduction section, can also be incorporated using additional constraints. The proposed model in this paper adopted deterministic optimization. As part of our future research, stochastic optimization will be pursued for this scheduling problem. Uncertainty factors such as processing time, setup time, and demand size can be integrated into the stochastic scheduling problem model.

The model formulated and solved using CPLEX (or other solvers) can be integrated with other software such as Excel to have a more user friendly interface. With such integration, optimal schedules can be attained within a short lead-time and be readily available for decision-making processes. Through optimal production schedules, companies can improve the utilization of different production resources and overall planning functions for their manufacturing processes.

Acknowledgements

We would like to extend our deepest gratitude for all the information provided by Dr. Giorgos Kopanos and Dr. Michalis Georgiadis. Part of this research is supported by National Science Foundation grants #1140457 and #1457880. Their support is greatly appreciated.

Appendix.

A.1. Appendix 1: The complete modified model

Objective: *Min Cmax*
Constraints:

$$L_{ibs} + \tau_i^{fill} = C_{ibs} \quad \forall i, \quad b \le \beta_i^{\min}, \quad s = 1 \tag{A1}$$

$$L_{ibs} + \tau_i^{fill} + \tau_i^{ag} + W_{ibs} + \tau_i^{empty} = C_{ibs} \quad \forall i, \quad b \le \beta_i^{\min}, \quad s = 2 \tag{A2}$$

$$W_{ibs} \le \varepsilon_i^{life} - \tau_i^{ag} \quad \forall i, \quad b \le \beta_i^{\min}, \quad s = 2 \tag{A3}$$

$$L_{ibs} + \tau_i^{empty} = C_{ibs} \quad \forall i, \quad b \le \beta_i^{min}, \quad s = 3 \tag{A4}$$

$$L_{ibs} = L_{ibs-1} \quad \forall i, \quad b \le \beta_i^{min}, \quad s = 2 \tag{A5}$$

$$C_{ibs} = C_{ibs-1} \quad \forall i, \quad b \le \beta_i^{\min}, \quad s = 3 \tag{A6}$$

$$C_{ibs} \le L_{ib+1s} \quad \forall i, \quad b \le \beta_i^{\min}, \quad s = 3 \tag{A7}$$

$$C_{oibs} = W_{0ibs} + \tau_i^{empty} \quad \forall i, \quad b \le \beta Itransfer_i^{\min}, \quad s = 2 : i \ \text{ in } \ Itransfer \tag{A8}$$

$$C_{oibs} = L_{0ibs} + \tau_i^{empty} \quad \forall i, \quad b \le \beta Itransfer_i^{\min}, \quad s = 3 : i \ \text{ in } \ Itransfer \tag{A9}$$

$$C_{0ibs} = C_{0ibs-1} \quad \forall i, \quad b \le \beta Itransfer_i^{\min}, \quad s = 3 : i \ \text{ in } \ Itransfer \tag{A10}$$

$$C_{0ibs} = C_{0ibs+1s} \quad \forall i, \quad b \le \beta Itransfer_i^{\min} - 1, \quad s = 3 : i \ \text{ in } \ Itransfer \tag{A11}$$

$$\sum Y_{ibsj} = 1 \quad \forall i, \quad i \in I, \quad b \le \beta_i^{\min}, \quad s = 2, \tag{A12}$$

$$Y_{ibsj} = 1 \quad \forall i, \quad i \in Itransfer, \quad b = 1, \quad s = 2, \quad j = first(J2_i) \tag{A13}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in Itransfer, \quad b \le \beta Itransfer_i^{\min} - 2, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} \le \beta_i^{\min} \tag{A14}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in Itransfer, \quad b \le \beta_i^{\min} - 1, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} > \beta_i^{\min} \tag{A15}$$

$$\begin{aligned} &Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, \quad i \in Itransfer, \quad b = \min_i(\beta Itransfer_i^{\min} - 1,)i' \in Last(IPred_i), \\ &b' = \beta Itransfer_{i'}^{\min}, \quad s = 2, \quad j \in J2_i : card(IPred_i) > 0 \end{aligned} \tag{A16}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in Itransfer, \quad b = \beta Itransfer_i^{\min} - 1, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} \le \beta_i^{\min}, card(IPred_i) = 0 \tag{A17}$$

$$\begin{aligned} &Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, \quad i \in Itransfer, \quad b = \beta Itransfer_i^{\min} - 1, \quad i' \in ISuc_i, \\ &b' = \beta Itransfer_{i'}^{\min}, \quad s = 2, \quad j \in J2_i : \beta Itransfer_i^{\min} - 1 > \beta_i^{\min}, card(IPred_i) = 0 \end{aligned} \tag{A18}$$

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, \quad i \in I, \quad b = \beta_i^{\min}, \quad i' \in ISuc_i, \quad b' = \beta Itransfer_i^{\min}, \quad s = 2, \quad j \in J2_i \tag{A19}$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, \quad i \in I, \quad \beta Itransfer_i^{\min} \le b \le \beta_i^{\min}, \quad s = 2, \quad j \in J2_i \tag{A20}$$

$$L_{i'b's} \ge C_{ibs} + \gamma_{ii'} - \omega\left(1 - \bar{X}_{ibi'b'}\right) \quad \forall i, \quad i \in I, \quad b \le \beta_i^{\min}, \quad i' \in I, \quad b' \le \beta_{i'}^{\min}, \quad s = 1, \quad j \in J_i \cap J_{i'} \cap Mach\,[s] : i < i' \tag{A21}$$

$$L_{i'b's} \ge C_{i'b's} + \gamma_{i'i} - \omega\bar{X}_{ibi'b'} \quad \forall i, \quad i \in I, \quad b \le \beta_i^{min}, \quad i' \notin I, \quad b' \le \beta_{i'}^{min}, \quad s = 1, \quad j \in J_i \cap J_{i'} \cap Mach\,[s] : i < i' \tag{A22}$$

$$L_{i'b's} \ge C_{ibs} + \gamma_{ii'} \quad \forall i, \quad i \in I, \quad b \le \beta_i^{min}, \quad i' \in I_i^{SP}, \quad b' \le \beta_{i'}^{min}, \quad s \ne 2, \quad j \in J_i \cap J_{i'} \cap Mach[s] : \theta_i < \theta_{i'} \tag{A23}$$

$$L_{i'b's} \geq C_{ibs} + \gamma_{ii'} - \omega\left(2 - Y_{ibsj} - Y_{i'b'sj}\right) \quad \forall i, \quad i \in I, \quad b \leq \beta_i^{min}, \quad i' \in I_i^{SP}, \quad b' \leq \beta_{i'}^{min}, \quad s = 2, \quad j \in J_i \cap J_{i'} \cap Mach[s] : \theta_i < \theta_{i'} \tag{A24}$$

$$L_{ib's} \geq C_{ibs} \quad \forall i, \quad i \in I, \quad b \leq \beta_i^{min}, \quad b' \leq \beta_{i'}^{min}, \quad s = 2, \quad j \in J_i \cap Mach[s] : b < b' \tag{A25}$$

$$L_{ib's} \geq C_{ibs} - \omega\left(2 - Y_{ibsj} - Y_{i'b'sj}\right) \quad \forall i, \quad i \in I, \quad b \leq \beta_i^{min}, \quad b' \leq \beta_{i'}^{min}, \quad s = 2, \quad j \in J_i \cap Mach[s] : b < b' \tag{A26}$$

$$L_{i'b's} = C_{oibs} + \gamma_{ii'} - \omega(2 - Y_{ibsj} - Y_{i'b'sj}) \quad \forall i, \quad i \in Itransfer, \quad b \leq \beta Itransfer_i^{\min} - 1, \quad i' \in I, \beta Itransfer_{i'}^{\min} \leq b' \leq \beta_{i'}^{\min}, \quad s = 2, j \tag{A27}$$

$$L_{i'b's} = C_{0ibs} + \gamma_{ii'} \quad \forall i, \quad i \in Itransfer, \quad b \leq \beta Itransfer_i^{\min} - 1, \quad i' \in I, \quad \beta Itransfer_{i'}^{\min} \leq b' \leq \beta_{i'}^{\min},$$
$$s = 3, \quad j \in J_i \cap Mach[s] : i \in Itransfer \tag{A28}$$

$$C_{Max} \geq C_{ibs} \quad \forall i, \quad i \in I, \quad b \leq \beta_i^{min}, \quad s \geq 2 \tag{A29}$$

$$C_{\max} = \phi_j^{\min} + \left(\alpha_j^{\min} - 1\right)\gamma_j + \min_j \sum\nolimits_{i \in I_j} \tau_i^{fill} \beta_i^{\min} \quad \forall j \in Mach[s] : s = 3 \tag{A30}$$

$$Y_{ibsj} \varepsilon \{0, 1\} \quad \forall i, \quad i \in I \quad b \leq \beta_i^{\min}, \quad s = 2, \quad j \in J_i \cap Mach[s] \tag{A31}$$

$$\bar{X}_{ibi'b'} \varepsilon \left\{0, 1\right\} \quad \forall i, \quad i \in I \quad b \leq \beta_i^{\min}, \quad i' \in I_i^{SP}, \quad b \leq \beta_i^{\min} : i < i' \tag{A32}$$

$$L_{ibs}, C_{ibs}, C_{0ibs} \geq 0 \quad \forall i, \quad i \in I \quad b \leq \beta_i^{\min}, \quad s \in S \tag{A33}$$

$$W_{ibs}, W_{0ibs} \geq 0 \quad \forall i, \quad i \in I \quad b \leq \beta_i^{\min}, \quad s = 2 \tag{A34}$$

$$L_{ibs} \leq (n * Workweeklength) \&\& C_{ibs} \geq (n * Workweeklength) - IdleGamma_i)$$
$$= > L_{ibs} \geq (n) * Workweeklength \quad \forall i, \quad j \in J_s, \quad i \in I, \quad \beta Itransfer_i^{\min} \leq b \leq \beta_{i'}^{\min}, \quad s = 3, \quad n \in WeekNumber \tag{A35}$$

$$L_{ibs} \leq (n * Workweeklength) - IdleGamma_i) \&\& L_{ibs} \leq (n * Workweeklength)$$
$$\Rightarrow L_{ibs} \geq (n) * Workweeklength \quad \forall i, \quad j \in J_s, \quad i \in I, \quad \beta Itransfer_i^{\min} \leq b \leq \beta_{i'}^{\min}, \quad s = 1, \quad n \in WeekNumber \tag{A36}$$

$$CWeek_{ibs} \leq (C_{ibs} / (Workweeklength - IdleGamma_i)) + 1 \quad \forall i, \quad i \in I, \quad j \in J, \quad s \in S \tag{A37}$$

$$CWeek_{ibs} \geq C_{ibs} / (Workweeklength - IdleGamma_i) \quad \forall i, \quad i \in I, \quad j \in J, \quad s \in S \tag{A38}$$

A.2. Appendix 2: CPLEX setup for experimental runs

First experimental run setup

- computational time report set to CPU time,
- a global time limit equal to 600.00 s (10 min).

Second and third experimental run setup

- computational time report set to Wall Clock time,
- a global time limit equal to 600.00 s (10 min),
- the limit on nodes explored when a subMIP is being solved set to 75,
- the absolute gap for solution pool set to 10.0,
- the relative gap for solution pool set to 10.0,
- limit on the number of solutions kept in the pool set to 21.

A.3. Appendix 3: Problem instances sets (the numbers are in 1000 kg)

Set 1	Problem instance																			
Product mix	1	2	3	4	5	6	7	8	9	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

Set 2	Problem instance																			
Product mix	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

Set 3	Problem instances									
Product mix	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

A.4. Appendix 4: Setup time and other data

	τ_i^{ag} (h)	τ_i^{empty} (kg/h)
A	1	1750
B	3	1500
C	3	1000
D	0	1500
E	2	1750
F	2	2000
G	2	2000
H	2	2000
I	2	1750
J	3	1500
K	2	2000
L	1	2000
M	3	2250
N	2	2000
O	3	1750
P	2	2250
Q	4	2500
R	2	1250
S	3	1500
T	1	2250
U	1	1500
V	2	2000
W	2	1750
X	2	2750

Processing changeover time from one product to another (h)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
A	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
B	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
C	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
D	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
E	0.50	0.50	0.50	0.50	0.00	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
F	0.50	0.50	0.50	0.50	0.08	0.00	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
G	0.50	0.50	0.50	0.50	0.08	0.08	0.00	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
H	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.00	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
I	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
J	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
K	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
L	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
M	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.00	0.25	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
N	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.00	0.25	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
O	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.00	0.25	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
P	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.00	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.25
Q	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50
R	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50
S	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50
T	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.50	0.50	0.50	0.50
U	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.00	0.25	0.25	0.25
V	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.00	0.25	0.25
W	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.00	0.25
X	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.08	0.50	0.50	0.50	0.50	0.08	0.08	0.08	0.00

References

Banerjee, S., Dangayach, G.S., Mukherjee, S.K., Mohanti, P.K., 2008. Modelling process and supply chain scheduling using hybrid meta-heuristics. In: Metaheuristics for Scheduling in Industrial and Manufacturing Applications., pp. 277–300.

Bilgen, B., Dogan, K., 2015. Multistage production planning in the dairy industry: a mixed-integer programming approach. Ind. Eng. Chem. Res., 11709–11719.

Bilgen, B., Çelebi, Y., 2013. Integrated production scheduling and distribution planning in dairy supply chain by hybrid modelling. Ann. Oper. Res., 55–82.

Bongers, P.M.M., Bakker, B.H.,2006. Application of multi-stage scheduling. In: 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering. Elsevier, Boston, pp. 1917–1922.

Doganis, P., Sarimveis, H., 2008a. Mixed integer linear programming scheduling in the food industry. In: Sun, D.-W., Erdogdu, F. (Eds.), Optimization in Food Engineering, vol. 305. Taylor & Francis Group, LLC, New York.

Doganis, P., Sarimveis, H., 2008b. Optimal production scheduling for the dairy industry. Ann. Oper. Res., 315–331.

Doganis, P., Sarimveis, H., 2007. Optimal scheduling in a yogurt production line based on mixed integer linear programming. J. Food Eng., 445–453.

Gellert, T., Höhn, W., Möhring, R.H., 2011. Sequencing and scheduling for filling lines in dairy production. Optim. Lett., 491–504.

Hecker, F.T., Hussein, W.B., Paquet-Durand, O., Hussein, M.A., Becker, T., 2013. A case study on using evolutionary algorithms to optimize bakery production planning. Expert Syst. Appl., 6837–6847.

Heinonen, J., Pettersson, F., 2003. Scheduling a specific type of batch process with evolutionary computation. In: Evolutionary Computation. IEEE, pp. 966–970.

Karray, A., Benrejeb, M., Borne, P., 2011. New parallel genetic algorithms for the single-machine scheduling problems in agro-food industry. In: Communications, Computing and Control Applications (CCCA). IEEE, Hammamet, pp. 1–7.

Kopanos, G.M., Puigjaner, L., Georgiadis, M.C., 2012. Efficient mathematical frameworks for detailed production scheduling in food processing industries. Comput. Chem. Eng., 206–216.

Kopanos, G.M., Puigjaner, L., Georgiadis, M.C., 2009. Optimal production scheduling and lot-sizing in dairy plants: the yogurt production line. Ind. Eng. Chem. Res., 701–718.

Kopanos, G.M., Puigjaner, L., Georgiadis, M.C., 2011. Production scheduling in multiproduct multistage semicontinuous food processes. Ind. Eng. Chem. Res., 6316–6324.

Liu, S., Pinto, J.M., Papageorgiou, L.G., 2010. Single-stage scheduling of multiproduct batch plants: an edible-oil deodorizer case study. Ind. Eng. Chem. Res., 8657–8669.

Lütke Entrup, M., Günther, H.-O., Beek, P.V., Grunow, M., Seiler, T., 2005. Mixed-integer linear programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. Int. J. Prod. Res., 5071–5100.

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. Ann. Oper. Res., 177–192.

Sadi-Nezhad, S., Darian, S.B., 2010. Production scheduling for products on different machines with setup costs and times. Int. J. Eng. Technol., 410–418.

Samarghandi, H., ElMekkawy, T.Y., 2012. A meta-heuristic approach for solving the no-wait flow-shop problem. Int. J. Prod. Res., 7313–7326.

Samarghandi, H., 2015. A particle swarm optimisation for the no-wait flow shop problem with due date constraints. Int. J. Prod. Res., 2853–2870.

Shaw, K.J., Lee, P.L., Nott, H.P., Thompson, M., 2000. Genetic algorithms for multiobjective scheduling of combined batch/continuous process plants. In: Evolutionary Computation. IEEE, La Jolla, CA, pp. 293–300.

Zhou, Y., Gu, X., 2009. Research on no-wait flow shop scheduling problem with fuzzy due date based on evolution games. In: Computer Science and Information Technology. IEEE, Beijing, pp. 495–499.