ELSEVIER

Computers and Chemical Engineering 24 (2000) 2223–2245

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Optimal scheduling of batch plants satisfying multiple product orders with different due-dates

C.A. Méndez, G.P. Henning, J. Cerdá *

INTEC (UNL-CONICET), Güemes 3450-3000, Santa Fe, Argentina

Abstract

In most multiproduct batch plants, the short-term planning activity starts by considering the set of product orders to be filled during the scheduling period. Each order specifies the product and the amount to be manufactured as well as the promised due date and the release time. Several orders can be related to the same product, though featuring different quantities and due-dates. The initial task to be accomplished by the scheduler is the so-called batching process that transforms the product orders to fill into equivalent sets of batches to be scheduled and subsequently assigns a due date to each one. To execute the batching procedure for a particular product, the scheduler should not only account for the preferred unit sizes but also for all the orders related to such a product and their corresponding deadlines. Frequently, a batch is shared by several orders with the earliest one determining the batch due-date. In this paper, a new two-step systematic methodology for the scheduling of single-stage multiproduct batch plants is presented. In the first phase, the product batching process is accomplished to minimize the work-in-process inventory while meeting the orders' due-dates. The set of batches so attained is then optimally scheduled to meet the product orders as close to their due dates as possible. New MILP continuous-time models for both the batching and the scheduling problems were developed. In addition, widely known heuristic rules can be easily embedded in the scheduling problem formulation to get a faster convergence to near-optimal schedules for 'real-world' industrial problems. Three example problems involving up to 29 production orders have been successfully solved in low computational time.

Keywords: Short-term scheduling; Multiproduct batch plant; Batching problem; MILP formulation; Hybrid approach

1. Introduction

The short-term schedule of a multiproduct batch plant is to be periodically made to satisfy a given set of production orders for different products within a fixed time horizon. The orders may come directly from customers or be generated to meet inventory requirements. A non-regular production pattern is usually adopted since the list of products to be manufactured changes every week and, consequently, a cyclic production policy is inadequate to follow market demands. The product orders usually specify what products and how much of them are to be made as well as their deadlines and release times. The orders should be delivered either to customers at the promised due dates, if any, or to inventory throughout the scheduling horizon. The order release time representing the earliest time, at which the order can be started, is an important datum for products that can spoil. The total amount of a certain product to be manufactured is usually distributed among several orders each one featuring a different due date. Given the set of orders to be processed, the scheduling task is the temporal assignment of orders to resources (units) in such a way that a number of goals and conditions are fulfilled. Among the conditions, some operating constraints forbidding the consecutive processing of certain products are generally enforced to preserve product quality and reduce scrap material and equipment idle time. By excluding such prohibited order sub-sequences from the problem formulation, the number of sequencing variables may be significantly diminished. Moreover, transition times must be sequence dependent to resemble what is usually found in industry. Depending on its direct predecessor in the processing sequence, the required setup time for a certain product will generally assume a different value.

* Corresponding author.
E-mail address: jcerda@intec.unl.edu.ar (J. Cerdá).

0098-1354/00/$ - see front matter
PII: S0098-1354(00)00584-6

In the last decade, a large number of papers was published in the area of planning and scheduling of batch process operations. Extensive reviews have also been reported (Reklaitis, 1992; Zentner & Pekny, 1994; Applequist, Samikoglu, Pekny & Reklaitis, 1997; Pinto & Grossmann, 1998; Shah, 1998). Short-term scheduling approaches are generally classified into two classes according to the type of time domain representation being used. They are known as discrete-time and continuous-time scheduling methods. Discrete-time formulations divide the time horizon into a number of intervals of equal duration (Kondili, Pantelides & Sargent, 1993; Shah, Pantelides & Sargent, 1993; Gooding, Pekny & McCroskey, 1994). Though they are able to account for many scheduling features (storage modes, changeovers, resource constraints, batch mixing and splitting) and different plant layouts within the same framework, discrete-time formulations present two major drawbacks. They are: (a) the approximate time domain representation; and (b) the very large number of binary variables and constraints that results when real industrial problems are tackled. In fact, the problem size shows a combinatorial increase with the number of product orders and the extent of the time horizon. In addition, finding a uniform discretization of the time horizon where the events just take place at the boundaries of each interval is generally an unworkable task. Recent trends are then aimed at developing computationally efficient continuous time scheduling methodologies. Pinto and Grossmann (1995, 1996) developed an MILP optimization model and a solution method to the short-term scheduling of multistage batch plants that may include units in parallel at every processing stage. The problem has been modeled through a continuous time representation that relies on the time-slot notion and the use of parallel time axis for units and tasks. The major assumptions were sequence-independent set-ups and no resource constraints except equipment. Since the branch-and-bound solution method showed limitations for industrial-size problems, alternative solution schemes consisting on either the use of preordering constraints or the application of a two-step decomposition scheme were presented. Zhang and Sargent (1994) introduced another type of continuous time representation for a general resource-constrained batch/continuous plant scheduling problem based on the resource-task network (RTN) concept and the partitioning of the scheduling horizon into intervals of unknown duration. However, it assumed fixed task-unit assignments. The resultant large-scale highly complex MINLP was solved through an iterative sequence of MILPs and applied to a simple example. Using a similar type of approach, Schilling and Pantelides (1996) proposed an MILP continuous time formulation for the optimal scheduling of processes represented as resource-task networks (RTN). Though it still exhibits a large integrality gap, a novel branch-and-bound algorithm that branches on both discrete and continuous variables was proposed to address such computational issues. Mockus and Reklaitis (1997) proposed an alternative MINLP non-uniform time discretization method that relies on the state-task network (STN) and regards the timing of all events, batch sizes and task-unit allocations as problem variables to be determined. Nonetheless, the treatment of sequence-dependent changeovers requires a sharp increase in the number of variables and constraints. Cerdá, Henning and Grossmann (1997) introduced an MILP mathematical formulation for the scheduling of a given set of production orders in a multiproduct batch plant based on the linear precedence structure of the processing sequence at every unit. Order-unit assignments and order sequencing were jointly handled by defining a single set of binary variables and the mean tardiness was chosen as the problem objective to minimize. The model becomes quite attractive to deal with sequence-dependent changeovers and forbidden sequences caused by plant operational constraints. Karimi and McDonald (1997) introduced slot-based MILP formulations for the detailed short-term scheduling of a single-stage multiproduct facility involving multiple parallel semicontinuous units and sequence-dependent changeovers. The proposed representation seeks a production schedule fulfilling a given set of product orders (sizes and due dates) at minimum inventory, changeover and shortage costs. Ierapetritou and Floudas (1998) developed an MILP multiproduct/multipurpose batch plant scheduling model based on the state-task network (STN) representation. Task events and unit events have been decoupled to reduce the number of binary/continuous variables and thus improving the computational efficiency of the solution method. Instead of pre-defining time slots for each unit, the approach requires to guess the necessary number of event points each one corresponding to the start of either a task or a unit utilization. A simple iterative procedure is used for determining the number of event points needed. Though resource constraints can be handled, changeover times are not regarded as sequence-dependent. The authors claim that their formulation systematically requires fewer binary variables and exhibits smaller integrality gaps.

So far, few contributions have been focused on the study of the short-term scheduling problem weekly solved in the manufacturing industry where a given set of product orders requested by customers at specified due dates and/or made for inventory are to be accomplished within a fixed time horizon. Since the production requirement pattern in such facilities changes every week, a non-cyclic scheduling policy is commonly adopted. Rather than considering them as specified data as they usually are, most authors regarded the size of the production orders as problem variables that can be chosen during the scheduling process so as to maximize the net profit. Market demands are just given as bounds for such order sizes. This paper presents a new mathematical formulation for the non-cycling scheduling of multiproduct batch plants involving a unique processing stage with several units in parallel. The major features of the proposed approach are:

1. Use of a continuous time domain representation that does not rely on the definition of time slots or time events.
2. Separate handling of assignment and sequencing decisions through different sets of binary variables to get a reduction in the number of 0–1 variables.
3. Definition of 0–1 sequencing variables based on the linear precedence structure of the processing sequence at each unit. Each order after the first has exactly one direct predecessor and each order before the last has exactly one direct successor in the sequence. Such a type of sequencing variable makes much easier to derive a scheduling problem formulation accounting for sequence-dependent setup times and forbidden sub-sequences.
4. Release times for orders and ready-times for units may assume positive values without any further change in the problem formulation.

In prior work, relaxing the sequence-independent transition time assumption or excluding forbidden job sub-sequences to reduce the number of 0–1 variables would have considerably complicated the problem model (Pinto & Grossmann, 1995). Another advantage of the proposed formulation is that the computational effort does not depend at all on the number of time slots for each unit or the number of event points defined beforehand as long as it is not based on such basic concepts. As a result, the potential source of model degeneracy due to the equivalence of time slots at each unit is avoided. In addition, relaxed pre-ordering conditions can be easily embedded into the mathematical model to further reduce both the problem size and the computing time and still find a very good schedule.

This paper has been organized as follows. First, a new MILP mathematical model aimed at scheduling a given set of single-batch production orders (or simply batches) in a multiproduct batch plant while fulfilling the promised due dates is presented. Afterwards, the assumption of single-batch order is relaxed and the production requirements are given in mass units, as it is usually the case in the chemical and food industry. Moreover, a production order may be equivalent to either a real number of batches or even a fraction of a single batch. A MILP formulation is then derived to optimally: (i) convert production requirements into a set of batches; and (ii) allocate every batch to one or several orders and choose the due date for each one, so that all the orders are timely satisfied at minimum work-in-process inventory. Next, the set of batches so attained is scheduled by using the MILP batch scheduling approach first introduced in the paper. Two medium-scale industrial examples dealing with the scheduling of single-batch orders are subsequently solved. Finally, a third example involving first the batching of non-single batch orders and then, the scheduling of the resultant set of batches is tackled.

2. Nomenclature

Before introducing the proposed mathematical models for both the scheduling of batches in a single-stage multiproduct batch plant with units in parallel and the batching of non-single batch orders, the following sets and parameters are first to be defined.

Subindices

i, i'	manufacturing orders
j	processing unit
p	product
d, d'	due date
s	unit group

Sets

I	set of manufacturing orders to be scheduled
I_j	set of orders which can be processed in unit j ($I_j \subseteq I$)
I_p	set of orders involving product p ($I_p \subseteq I$)
FS	set of forbidden processing sequences (orders i, i' belong to the set FS only if order i' can never be processed right after order i in a particular unit)
A_i	set of orders to be completed before starting order i because of technological precedence constraints ($A_i \subseteq I$)
J	set of units
J_i	set of units which can process order i ($J_i \subseteq J$)
$J_{ii'}$	set of units which can process both orders i and i' ($J_{ii'} \subseteq \mathrm{J}$)
J_p	set of units which can process product p ($\mathrm{J_p} \subseteq \mathrm{J}$)
J_s	set of units which belong to group s ($J_s \subseteq J$)
P	set of products to be manufactured
P_j	set of products which can be processed in unit j ($P_j \subseteq P$)
S	set of different unit groups
S_p	set of unit groups which can process product p ($S_p \subseteq S$)
D	set of order due dates
D_p	set of due dates for orders involving product p ($D_p \subseteq D$)

Order-related parameters

N	number of orders to be scheduled
p_i	priority of order i/product related to order i
$(\mathrm{ro})_i$	release time of order i
d_i	due date of order i
t_{ij}	processing time of order i in unit j
$\tau_{ii'j}$	sequence-dependent setup time between orders i and i' in unit j

Unit-related parameters

$(\mathrm{ru})_j$	ready time of unit j
$(\mathrm{su})_{ij}$	unit-dependent setup time of order i in unit j

Product-related parameters

p_{pd}	priority of the order involving product p and due at time d
θ_{pj}	processing time of a batch of product p in unit j
Q_{pd}	accumulated requirement of product p at due date d
B_{pj}	specified batch size for product p in unit j
B_{ps}	batch size for product p in each unit of the group $s \in S_p$

3. Model assumptions

The following assumptions have been used to derive the short-term scheduling problem formulation to be introduced in the next two sections:

(i) The model parameters are all deterministic.
(ii) The time horizon is a given problem datum.
(iii) Transition times are expressed as the sum of two terms: a unit-dependent and a sequence-dependent term (this one also varying with the unit).
(iv) Once the processing of an order starts, it should be carried out until completion without interruption (non-preemption processing mode).
(v) Every order involves a single batch and a specified due date.
(vi) Every batch is completely allocated to a single order. Two or more orders involving the same product and featuring different due-dates can never share a particular batch.
(vii) Each unit runs at maximum effective capacity, unless a lower product batch size has been specified for operating reasons. In other words, product batch sizes are fixed parameters.
(viii) No resource constraints except equipment are taken into account.

Because of assumption (vii), the product batch size at every unit is known beforehand. Then, the problem formulation may implicitly consider the variation of the processing time with the batch size, if any, without losing linearity. In a real-world batch industrial plant, assumptions (iv) and (vii) are frequently satisfied. This is not the case, however, for assumptions (v) and (vi) which should be relaxed before the scheduling problem model really resembles what is normally found in industry and, by so doing, better production schedules can be generated. In fact, batches related to a non-single batch order are often produced in different units and some of them are even processed at the same time. Moreover, if the orders do not comprise an integer number of batches, a normal industrial practice is then to allocate some batches to two or more orders (batch splitting) so as to run units at full effective capacity. To determine the set of batches to schedule, therefore, the scheduler should select the batch sizes and the real number of batches of each size to be allocated to every order. In a later section, the so-called batching problem formulation is introduced to choose the set of batches to schedule and their due dates and to additionally provide tools for refining the specified set of production orders and their promised due dates. Even assumption (vii) can be relaxed through a proper formulation of the batching problem.

Current continuous-time scheduling approaches like those proposed by Schilling and Pantelides (1996), Mockus and Reklaitis (1997) and Ierapetritou and Floudas (1998) were derived without assuming fixed product batch sizes and single-batch orders (assumptions v and vii). Moreover, the set of batches to be scheduled, including the size for each one, and the optimal batch schedule itself are found by solving a unique problem formulation. In other words, such scheduling methodologies can be regarded as single-level approaches. However, they usually assume a single order per product (destined to inventory) to be completed before ending the time horizon, i.e. no due date is specified. Consequently, every batch of product is allocated to just a single order (assumption vi). Though a single-level scheduling approach is a more rigorous one, it generally leads to a much larger problem formulation especially if real-world scheduling problems are tackled. Additional pairs of binary/continuous variables each one denoting the decision of producing a batch and its optimal size, should be considered. Since the number of batches to be processed to meet a production order is unknown before solving the problem, then it should be overestimated at the time of guessing the number of event points or intervals of unknown duration to use in the problem formulation. Moreover, the approach must also include allocation variables standing for the assignment of batches to orders rather than products when multiple orders for a particular product are considered.

4. Model variables

Three sets of variables, as many as the types of scheduling decisions to be taken, have been defined.

- Order-unit assignment
  WF_{ij} binary variable denoting that order i is the earliest processed in unit j
  W_{ij} binary variable denoting that order i has been scheduled in unit j but not in first place (it can be defined as a continuous variable restricted to the range [0, 1]).
- Order sequencing
  $X_{ii'}$ binary variable denoting that order i' has been scheduled right after order i
- Order time allocation
  C_i completion time of order i
- Order performance measures
  NT_i binary variable denoting that order i is tardy. Only used if the minimum number of tardy orders is sought.
  E_i earliness of order i
  T_i tardiness of order i
- Aggregate performance measures
  MK makespan

Fig. 1 shows a Gantt diagram where eight manufacturing orders have been scheduled in a pair of units U1 and U2. To specify the processing sequences at units U1 and U2, the following assignment and sequencing variables should be made equal to one: (a) the assignment variable $WF_{1,U1}$ to denote that order 1 is the first processed in unit U1; (b) the assignment variables $W_{3,U1}$, $W_{2,U1}$ and $W_{7,U1}$ to indicate that orders {3, 2, 7} are all processed in unit U1 but not in the first place; (c) the sequencing variables $X_{1,3}$, $X_{3,2}$ and $X_{2,7}$ to denote that orders {1, 3, 2} are succeeded by orders {3, 2, 7}, respectively, in unit U1; and (d) the assignment variables $WF_{5,U2}$, $W_{8,U2}$, $W_{6,U2}$ and $W_{4,U2}$ as well as the sequencing variables $X_{5,8}$, $X_{8,6}$ and $X_{6,4}$ to indicate that orders {5, 8, 6, 4} have been scheduled in unit U2 and the direct successors to the first three orders are {8, 6, 4}, respectively. The remaining assignment and sequencing variables are all equal to zero.

5. The short-term scheduling problem formulation

In the proposed mathematical formulation, the problem constraints have been grouped according to the type of decision (assignment, sequencing, timing) upon which they are imposed.

5.1. Allocation of manufacturing orders to units

5.1.1. Every manufacturing order must be assigned to a single processing unit j

This implies that order i cannot be split and processed in two or more units. When order i is the first processed in unit j, then $WF_{ij} = 1$ and W_{ij} drops to 0. If i is processed in unit j but following another order in the processing sequence, then W_{ij} rather than WF_{ij} becomes equal to one. If order i is not processed in unit j, both WF_{ij} and W_{ij} are equal to zero.

$$\sum_{j\in J_i} WF_{ij} + \sum_{j\in J_i} W_{ij} = 1 \qquad \forall i\in I \tag{1}$$

5.1.2. A production order and its direct successor in the processing sequence are both manufactured in the same unit

Whenever order i is directly succeeded by order i' ($X_{ii'} = 1$) in the processing queue and manufactured in unit $j\in J_{ii'}$ (W_{ij} or $WF_{ij} = 1$), then order i' must also be assigned to unit j and the variable $W_{i'j}$ should become equal to one. This condition is imposed upon the value of $W_{i'j}$ through inequality (2) that relates assignment and sequencing decision variables among themselves. For any unit $j\in J_i$ at which order i' cannot be processed ($j\notin J_{ii'}$), the variable $W_{i'j}$ is never defined and inequality (2) reduces to (3).

$$WF_{ij} + W_{ij} \le W_{i'j} + 1 - X_{ii'} \qquad \forall j\in J_{ii'}, \quad i, i'\in I, \quad (i, i')\notin \mathrm{FS} \tag{2}$$

$$WF_{ij} + W_{ij} \le 1 - X_{ii'} \qquad \forall j\in \{J_i - J_{ii'}\}, \quad i, i'\in I, \quad (i, i')\notin \mathrm{FS} \tag{3}$$

After defining the variables WF_{ij} as binary, constraints (1) and (2) (or its equivalent (3)) permit to relax the integer condition on the variables W_{ij}. Despite that, they still take 0–1 values by constraining such continu-

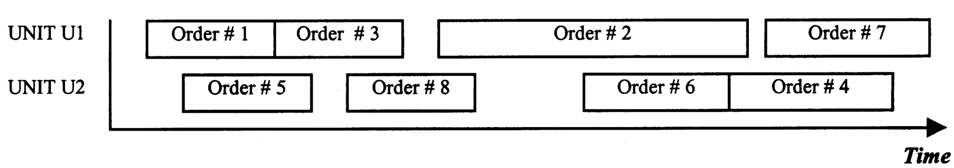

Fig. 1. A simple schedule illustrating the meaning of assignment and sequencing variables.

ous variables to the range [0–1]. In this way, an important saving of binary variables is achieved. It should also be noted that variables $X_{ii'}$ and constraints (2) and (3) related to prohibited order sub-sequences $(i, i')\in$FS are never defined. Thus, the problem formulation ignores from the start every schedule involving forbidden sub-sequences and, consequently, the computational performance of the solution algorithm is significantly improved.

5.2. Order sequencing

5.2.1. A unique order can at most be first processed at every unit

In Fig. 1, order #1 leads the processing sequence at unit U1 ($WF_{1,U1}=1$), and order #5 is the earliest one processed in unit U2 ($WF_{5,U2}=1$).

$$\sum_{i\in I_j} WF_{ij} \le 1 \qquad \forall j\in J \tag{4}$$

5.2.2. Every manufacturing order should either be the first processed or directly preceded by another order in the processing sequence

This sequencing constraint states that every time order i is the first processed in unit j ($WF_{ij}=1$), it cannot feature a direct predecessor i' in the processing sequence ($X_{i'i}=0$ for any $i'\in I_j$ and $(i', i)\notin$FS). On the contrary, there must exist a single order i' directly preceding order i ($X_{i'i}=1$) whenever order i is not the earliest manufactured at the assigned unit j ($WF_{ij}=0$ for any $j\in J_i$). Therefore,

$$\sum_{j\in J_i} WF_{ij} + \sum_{\substack{i'\in I \\ (i',i)\notin \mathrm{FS}}} X_{i'i} = 1 \qquad \forall i\in I \tag{5}$$

5.2.3. Every manufacturing order should at most be directly succeeded by another one in the processing sequence

This constraint states that at most a single order i' can be scheduled immediately after order i ($X_{ii'}=1$), unless order i is last in the sequence. In Fig. 1, order #5 is directly succeeded by order #8 ($X_{58}=1$) while orders #7 and #4 the last ones processed in units U1 and U2, respectively, are not succeeded by anyone.

$$\sum_{\substack{i'\in I \\ (i,i')\notin \mathrm{FS}}} X_{ii'} \le 1 \qquad \forall i\in I \tag{6}$$

5.3. Order timing

5.3.1. The processing start time for order i can never be lower than the completion time of its direct predecessor in the processing sequence

Constraint (7) states that order i' directly succeeding order i ($X_{ii'}=1$) in the jth-unit processing sequence should start after the completion of order i. The time interval between such events will just include the setup time for order i', which depends on both the preceding order i and the assigned unit j. The parameter M in (7) is a big positive number that makes such an inequality redundant whenever order i' does not directly succeed order i ($X_{ii'}=0$).

$$C_{i'} \ge C_i + \sum_{j\in J_{ii'}} (\tau_{ii'j} + su_{i'j} + t_{i'j}) W_{i'j} + MX_{ii'} - M$$
$$\forall i, i'\in I, \quad (i,i')\notin \mathrm{FS} \tag{7}$$

Though assignment and sequencing variables are separately treated, the sequence-dependent transition time $\tau_{ii'j}$ not only depends on the order i being processed right before order i' but also on the unit j where i' is manufactured.

5.3.2. The processing of order i in unit j can be started only if they both are ready

Inequality (8) states that the processing of order i in unit j can start earlier than neither its release-time (ro_i) nor the ready-time of the assigned unit j (ru_j). Very often, some units are still processing orders from the prior schedule at the beginning of the new production horizon and their ready-times are then positive.

$$C_i \ge \sum_{j\in J_i} (\max[\mathrm{ru}_j, \mathrm{ro}_i] + \mathrm{su}_{ij} + t_{ij})(WF_{ij} + W_{ij}) \quad \forall i\in I \tag{8}$$

5.4. Technological precedence constraints

5.4.1. Technological restrictions can impose precedence constraints upon the order processing sequence

Such precedence constraints usually specify the orders to be completed before starting a particular one. Whenever there exists a precedence relation between orders i and i', order i' should then be started after ending the processing of order i. This usually happens when order i involves the production of an intermediate required for the manufacture of i'.

$$C_i \le C_{i'} - \sum_{j\in J_{i'}} t_{i'j}(WF_{i'j} + W_{i'j}) \qquad \forall i'\in I, \quad i\in A_{i'} \tag{9}$$

5.5. Selected performance measures

5.5.1. Makespan

The makespan MK represents how much time is required to complete the processing of the whole set of manufacturing orders. Inequality (10) defining the variable MK is included in the problem formulation only if the makespan has been chosen to guide the scheduling decisions.

$$\mathrm{MK} \ge C_i \qquad \forall i\in I \tag{10}$$

5.5.2. Order earliness

The earliness of order i (E_i) is a non-negative continuous variable that takes a positive value only if it is completed earlier than its specified due-date ($C_i < d_i$). The definition of E_i should be included in the problem formulation whenever the earliness of order i takes part in the objective function.

$$E_i \geq d_i - C_i \qquad \forall i \in I \tag{11}$$

5.5.3. Order tardiness

The definition of the tardiness of order i (T_i) should be included as another constraint in the problem formulation only if the tardiness of order i participates in the objective function.

$$T_i \geq C_i - d_i \qquad \forall i \in I \tag{12}$$

5.5.4. Number of tardy orders

The binary variable NT_i indicates whether order i is a tardy order, i.e. $C_i > d_i$. Inequality (13) should be included as another constraint only if the number of tardy orders has been chosen as the schedule performance measure.

$$T_i - M\,NT_i \leq 0 \qquad \forall i \in I \tag{13}$$

5.6. Problem objective function

Different aggregate performance measures have been considered to guide the scheduling decisions simply to show that the proposed model is perfectly suited to a wide range of objective functions. Since the capacity of a multiproduct batch plant changes with the mix of products to be manufactured, then its value cannot be known beforehand. As a result, it is a difficult task to predict whether the specified set of orders can entirely be manufactured before ending the production horizon. It is even a more difficult assignment to establish in advance if every order will be completed on time, i.e. without tardiness. Because of that, a prior adjustment of the specified set of product orders and their due dates through efficient tools is often necessary to avoid either conflicting order tardiness or costly equipment idle time. Generally, the scheduler seeks a production schedule that minimizes the weighted mean tardiness while keeping order earliness and product inventory at reasonable values, i.e. the minimum weighted lateness. This performance measure has been adopted to solve the examples included in this paper. Since the set of production orders to be processed is one of the problem data, then both the total amount of each product to be processed and the raw material requirements do not practically change with the schedule being chosen. This usually happens in real-world manufacturing plants at the scheduling stage. Therefore, the value of products and the cost of feedstock were not included in the performance measure.

5.6.1. Minimum makespan

If all the orders are manufactured for inventory and they must be completed at the end of the time horizon, then the makespan arises as an attractive problem objective.

$$\min \mathrm{MK} \tag{14}$$

5.6.2. Minimum weighted mean earliness

The parameter p_i representing the priority of order i has been chosen as the weighting coefficient for order i. If the weighted mean earliness is to be minimized, then the objective function is given by

$$\min \sum_{i \in I} p_i E_i \tag{15}$$

5.6.3. Minimum weighted mean tardiness

$$\min \sum_{i \in I} p_i T_i \tag{16}$$

5.6.4. Minimum number of tardy orders

$$\min \sum_{i \in I} NT_i \tag{17}$$

5.6.5. Minimum weighted lateness

Often, there are several alternative schedules featuring the least weighted tardiness, but some of them show a rather large average earliness and a costly work-in-process inventory. The objective function (18) favors 'just-in-time' schedules with minimum lateness.

$$\min \sum_{i \in I} p_i \left(T_i + \frac{E_i}{N+1} \right) \tag{18}$$

where the earliness penalty includes a weight factor always lower than one, equal to $(N+1)^{-1}$, while the weighting coefficient for the tardiness penalty is just equal to one. This is so because customer satisfaction is the most important cost term. By completing the production orders as close to their due dates as possible, one is implicitly minimizing changeover times, inventory costs and penalties for missed demands at the same time.

6. Embedding soft preordering rules in the model: a hybrid approach

Due dates and processing times specified for the product orders can be used to reduce the size of the problem formulation by excluding from the feasible space a significant number of non-optimal processing sequences. If an order i' has to be completed much earlier than another order i ($d_{i'} \ll d_i$), it is then very likely that order i' will never be a direct successor of order i in the optimal schedule. Let us suppose that the production horizon is 1-week long and schedules involving processing sub-sequences (i, i') with $d_i \geq d_{i'} + 24$ h are usually non-optimal. In other words, order i is hardly processed immediately before order i' in some unit if it is due 24 h later. Generally speaking, the condition has an adjustable parameter that was chosen, in this particular case, equal to 24. To ignore such schedules involving poor sub-sequences (i, i') in the problem formulation, the related sequencing variables $X_{ii'}$ must never be defined whenever the condition $d_i \geq d_{i'} + 24$ h is satisfied. In this way, a relaxed earliest-due-date rule, called REDD rule, can be embedded in the mathematical model. The resultant number of binary variables may sharply drop, thus enhancing the computational efficiency of the branch-and-bound solution algorithm. However, application of the REDD-rule gives rise to a larger feasible space compared with the one generated by the original EDD-rule since it still includes to some extent short-term schedules where the orders are not strictly arranged by increasing due dates. Similarly, a relaxed minimum-slack-time rule, called RMST, can be defined to account for due dates and processing times at the same time. In this case, schedules involving processing sub-sequences (i, i') such that $(st)_i \geq (st)_{i'} + 24$ h will be ignored by simply excluding the related variables $X_{ii'}$ from the problem formulation. The slack time for order i will be estimated by using: $(st)_i = d_i - \min_{j \in J_i}\{t_{ij}\}$. In contrast to Pinto and Grossmann (1995), the proposed pre-ordering conditions do not increase the problem size and, in addition, the user can decide upon how much is sliced-off the original solution space by properly tuning the rule adjustable parameter. Neither further constraints nor additional variables are necessary to exclude an important number of non-optimal schedules from the feasible space. Moreover, the use of soft rather than hard preordering conditions makes possible that the model not only decides on what jobs to assign to which unit but also on what sequencing to choose at every one.

7. The product batching problem

In manufacturing batch plants, it is not uncommon that several batches of different sizes rather than a single one are required to meet a certain product order. If so, they are usually processed in different units and even simultaneously. Because of assumption (v), neither the short-term scheduling approach given in Section 5 nor most of the prior work is able to generate such a type of real-world multiproduct batch plant schedule. Moreover, the amount of product required to meet a production order cannot always be exactly satisfied through an integer number of batches. In such cases, a common industrial practice is to assign some of the batches to two or more orders involving the same product. A portion of the shared batch will remain in inventory after fulfilling the earliest order to which it was assigned. In this way, the manufacture of products for later orders is partially anticipated to prevent from having some idle capacity. The ultimate goal is to run the processing units at full effective capacity. However, such an opportunistic production policy always generates a temporary inventory that has a finite cost. A schedule showing such special features (batches shared by several orders) can never be selected by current algorithmic approaches, including the one described before, simply because it does not belong to the defined feasible space. In fact, all of the aforementioned approaches apply the hypothesis (vi) to derive the problem mathematical formulation. Assumptions (v) and (vi) imply that orders and batches are absolutely equivalent. Such equivalence between orders and batches is indeed necessary since the input data correspond to product orders and the batch plant scheduling problem is usually formulated in terms of batches to schedule rather than product orders to fill.

Therefore, an efficient complementary tool is to be developed to optimally convert every product order into a proper set of product batches before applying available batch scheduling methodologies. Otherwise, real-world batch plant schedules involving the production of multiple batches to meet a product order, the processing of such batches in different units even at the same time and the allocation of a single batch to several orders, will never be generated. Selection of the set of batches to schedule not only pursues the most efficient use of the available units under specified operational constraints (order due dates, preferred unit and batch sizes for every product). It also intends to minimize the product inventory cost resulting from the opportunistic ahead-of-time production policy.

The so-called product batching problem introduced in this paper is precisely concerned with the optimal selection of both batch sizes and number of batches of each size for every product to be processed along the production horizon. Assignment of a batch to several orders is permitted. In addition to that, it is also selected the time at which every batch is to be completed (its due date) to meet product orders in a just-in-time manner. Sometimes, however, the specified production requirements may surpass the available plant capacity and there will be a backlog of orders still waiting for processing at the end of the horizon. Moreover, the strict satisfaction of some order due dates frequently becomes an impracticable task. Nonetheless, let us assume first that the short-term production plan and the promised due dates were wisely selected accounting for the available plant capacity all over the scheduling horizon.

7.1. Model variables

Two additional sets of variables are to be defined in order to derive the mathematical formulation of the product batching problem. They are:

NB_{pjd} integer variable denoting the number of batches of product p processed in unit j to meet a production order with deadline d.

Γ_{pd} positive variable denoting the remaining inventory of product p at time d. It results from batches previously produced and partially allocated to production orders due up to time d. Such a product inventory can be assigned to orders with due dates later than d.

The amount of product p to be allocated to a production order with deadline d is given by Q_{pd}. In other words, the parameter Q_{pd} is the size of the order requiring product p at due date d. The total pth-product requirement due up to time d can be determined by adding the sizes of all orders involving product p and having due dates no later than d.

7.2. Model constraints

7.2.1. A sufficient number of batches of product p must be scheduled to fill every pth-product order

Eq. (19) defines the inventory of product p available from batches partly allocated to orders due up to time $d(\Gamma_{pd})$. Γ_{pd} can be used to meet subsequent orders featuring deadlines later than d. The LHS of Eq. (19) provides the total amount of product p that is manufactured (but sometimes not completely allocated) to meet orders requiring p with deadlines no later than d. In turn, the first term on the RHS stands for the accumulated requirements of product p up to time d. By difference of such two terms, the value of Γ_{pd} is determined. Accomplishment of Eq. (19) does not imply that the orders are all satisfied in time but it just guarantees that a sufficient number of batches have been scheduled to meet each one.

$$\sum_{j\in J_p}\sum_{\substack{d'\leq d\\ d'\in D_p}} B_{pj}\mathrm{NB}_{pjd'} = \sum_{\substack{d'\leq d\\ d'\in D_p}} Q_{pd'} + \Gamma_{pd} \qquad \forall p\in P, \quad d\in D_p \tag{19}$$

where $B_{pj} = B_{ij}$ for any order i involving product p, i.e. $i\in I_p$. In turn, D_p is the set of due-dates related to production orders requiring product p.

7.2.2. Every product order should be satisfied in time

Inequality (20) states that the set of batches processed in unit j to meet product orders due up to time d must be completed no later than d. The LHS of (20) provides the time interval needed to produce such a set of batches in unit j while the RHS gives the period of time really available to accomplish the processing task. Setup times in the batching problem are assumed to be just unit-dependent and are included in the processing time θ_{pj}.

$$\mathrm{ru}_j + \sum_{p\in P_j}\sum_{\substack{d'\leq d\\ d'\in D_p}} \theta_{pj}\mathrm{NB}_{pjd'} \leq d \qquad \forall j\in J, \quad d\in D \tag{20}$$

where: $\theta_{pj} = su_{ij} + t_{ij}$, for any order $i\in I_p$. Release times for the earlier orders to be processed are assumed to be zero.

7.3. Model objective function

The problem goal is to optimally choose the set of batches to schedule so that all product orders are satisfied in time and the accumulated product inventory cost throughout the scheduling horizon is minimized.

$$\min \sum_{p\in P}\sum_{d\in D_p} \mathrm{IC}_p v_{pd}\frac{\Gamma_{p,d-1} + \Gamma_{pd}}{2} \tag{21}$$

where IC_p is the daily inventory cost per unit amount of product p. In turn, v_{pd} is the interval (in days) delimited by the due date d_p and its direct predecessor $(d-1)_p$ in the ordered set D_p during which the pth-product inventory is assumed to be given by $(\Gamma_{p,d-1} + \Gamma_{pd})/2$.

The optimal value of NB_{pjd} provides the number of batches of product p and size B_{pj} to be processed with a common deadline d. Such batches can indeed be processed at any unit $j\in J_p$ where the specified batch size for product p is similar to that of unit j. In this way, the set of pth-product batches to be scheduled as well as their sizes and due dates can be simply derived. Moreover, the set of alternative units where each batch can be processed becomes also defined. The next step is to solve the short-term scheduling problem.

To tackle a much smaller batching problem formulation, all the units $j\in J_s$ processing the same subset of products $P_s \subseteq P$ and featuring a similar batch size B_{ps} and processing time θ_{ps} for each product $p\in P_s$ are grouped into a single cluster $s\in S$. Let the parameter n_s represent the number of individual units in the cluster s. An effective reduction of the problem size can be attained by expressing constraints (19) and (20) in terms of the integer variables NB_{psd} instead of NB_{pjd}. Since the cardinality of the set S is generally much lower than $|J|$, a significant saving in integer variables and constraints is so attained. The new formulations for (19) and (20) in terms of variables NB_{psd} and parameters $\{n_s, B_{ps}, \theta_{ps}\}$ are the following:

$$\sum_{s\in S_p}\sum_{\substack{d'\leq d\\ d'\in D_p}} B_{ps}\mathrm{NB}_{psd'} = \sum_{\substack{d'\leq d\\ d'\in D_p}} Q_{pd'} + \Gamma_{pd} \qquad \forall p\in P, \quad d\in D_p \tag{19a}$$

$$\sum_{j\in J_s}\mathrm{ru}_j + \sum_{p\in P_s}\sum_{\substack{d'\leq d\\ d'\in D_p}} \theta_{ps}\mathrm{NB}_{psd'} \leq d\, n_s \qquad \forall s\in S, \quad d\in D \tag{20a}$$

where S_p is the subset of unit groups that can process product p. Since the cluster s comprises n_s units, then the period of time available for completing the set of batches allocated to orders due up to d in the cluster s is now equal to n_s times d. In contrast, the objective function (21) remains the same. The new MILP formulation consists of the objective function (21) subject to constraints (19a) – (20a).

7.4. Optimizing the product batch sizes

A proper reformulation of constraint (19) permits to optimize the product batch sizes B_{ij} that are no longer fixed parameters but problem variables. It requires to define a new set of continuous variables F_{pjd} denoting the total amount of product p that is manufactured in unit j (but generally not completely allocated) to meet orders with deadline d. The new expression for (19) is:

$$\sum_{j\in J_p}\sum_{\substack{d'\leq d\\ d'\in D_p}} F_{pjd} = \sum_{\substack{d'\leq d\\ d'\in D_p}} Q_{pd'} + \Gamma_{pd} \qquad \forall p\in P, \quad d\in D_p \tag{19b}$$

In addition,

$$B_{pj,\,\min}\mathrm{NB}_{pjd} \leq F_{pjd} \leq B_{pj,\,\max}\mathrm{NB}_{pjd} \qquad \forall p\in P, \quad j\in J_p, \quad d\in D_p \tag{22}$$

The parameters $(B_{pj})_{\min}$ and $(B_{pj})_{\max}$ define the feasible range for the batch size of product p at unit j. Therefore, the set of batches to be processed and their optimal batch sizes can be simultaneously found by minimizing the linear objective function (21) subject to constraints (19b), (20) and (22).

7.5. Adjusting production targets to get problem feasibility

Let us now suppose that a rough-cut capacity requirement analysis to meet the specified product orders was really made. If so, it may happen that a backlog of orders is still waiting for processing at the end of the horizon due to insufficient plant capacity. Moreover, some promised due dates may represent unattainable targets in the sense that no workable schedule can meet all of them. In such a case, a modified version of the product batching problem can better be proposed to first align the production targets with the available plant capacity before accomplishing the batching process. The adjustment of production targets must then be followed by a fine-tuning of the order due dates. Let us first revise the production targets so that every specified order can be processed within the production horizon. Therefore, the length of the scheduling horizon (H) will act as a hard constraint for the makespan of any feasible schedule. In contrast, due date constraints can be ignored. To prevent from having a backlog of orders at the end of the horizon, low priority orders may be dropped, if necessary, from the original production requirements. To do that, the product requirements arising in constraint (19) as known problem data now become continuous variables q_{pd} having the original values Q_{pd} as their upper bounds ($q_{pd} \leq Q_{pd}$). The problem goal is to maximize the total value of products manufactured throughout the horizon, with order priorities used as weighting coefficients. Order priority can change with both the product and the due date. Therefore, it results the following MILP problem formulation:

$$\sum_{j\in J_p}\sum_{\substack{d'\leq d\\ d'\in D_p}} B_{pj}\mathrm{NB}_{pjd'} = \sum_{\substack{d'\leq d\\ d'\in D_p}} q_{pd'} + \Gamma_{pd} \qquad \forall p\in P, \quad d\in D_p \tag{23}$$

$$\mathrm{ru}_j + \sum_{p\in P_j}\sum_{d\in D_p} \theta_{pj}\mathrm{NB}_{pjd} \leq H \qquad \forall j\in J \tag{24}$$

$$q_{pd} \leq Q_{pd} \qquad \forall p\in P, \quad d\in D_p \tag{25}$$

where H is the length of the scheduling horizon. The objective function is now given by,

$$\max \sum_{p\in P}\sum_{d\in D_p} p_{pd}\, q_{pd} \tag{26}$$

The optimal value of q_{pd} provides the new size of the order involving product p and due date d. In this way, it is attained a new set of product orders with modified requirements and similar due dates to which the batching problem is subsequently applied.

7.6. Tuning-up order due dates

Although the production requirements may perfectly fit the available plant capacity, the resultant product orders are not necessarily satisfied in time. To remedy this situation before accomplishing the batching process, the scheduler has two options:

(i) To impose the intermediate due dates as hard constraints on the completion times of the related orders and at the same time allowing further modifications of the order requirements q_{pd};
(ii) To tune-up the order due dates by taking them as soft constraints that should be fulfilled but may be relaxed.

If option (i) is adopted, an additional refining of the order sizes by solving the MILP model comprising restrictions (20), (23), (25) and the problem goal (26) should be made. The new order sizes will next be used as fixed data in the batching problem formulation given by Eqs. (19)–(21). A subsequent solution of the product batching problem permits to find the set of batches of each product to be processed so as to meet the newly defined set of product orders at the promised due dates. In case (ii), the production targets q_{pd} that result from aligning production requirements with the plant capacity are taken as fixed parameters and no further change in the order sizes is made. However, order due dates can be adjusted. Then, the MILP mathematical formulation for the due date tuning-up problem will include Eq. (19) with the newly defined production targets q_{pd} and a modified form of the inequality (20) now written as follows,

$$ru_j + \sum_{p \in P_j} \sum_{\substack{d' \le d \\ d' \in D_p}} \theta_{pj} NB_{pjd'} \le d + \Delta_d \qquad \forall j \in J, \quad d \in D \tag{27}$$

where Δ_d is the due date shifting for every production order required at time d so as to meet every product requirement without tardiness. The new due date for those orders will be $(d + \Delta_d)$. The problem objective is to keep such due date changes as small as possible by minimizing the overall constraint violation of the original deadlines measured through the sum of the variables Δ_d.

$$\min \sum_{d \in D} \Delta_d \tag{28}$$

8. Results and discussion

Three large-scale short-term scheduling problems involving from 20 to 29 production orders were solved through the proposed MILP algorithmic approach. In each case, a due date has been specified for every order, transition times are sequence-dependent and the product batch size at each unit is also a given datum. The problem goal is to schedule a given set of production orders in a single-stage multiproduct plant following a just-in-time policy. In the first two examples, either a single batch or a single production campaign satisfies every product order. In other words, there is a one-to-one correspondence between orders and units. In the third example, however, most products feature several requirements in kilograms at different due dates. Furthermore, each order is not generally satisfied through an integer number of batches and, in addition, the set of batches allocated to it is not sequentially processed in a single unit. Therefore, some batches are shared by several orders and the proposed MILP batching problem formulation should first be solved. In this way, the equivalent set of batches to be processed is optimally generated so as to complete all the product orders just in time at minimum inventory cost. Since the short-term scheduling problem is formulated in terms of batches, those ones of similar and/or different sizes allocated to a certain order can be processed in different equipment items, even at the same time. Both batching and short-term scheduling problems were solved on a 400 MHz, 128 MB Pentium II PC using the GAMS modeling system (Brooke, Kendrick & Meeraus, 1992) and the OSL advanced branch-and-bound code (IBM, 1991). To get a much smaller scheduling problem and still find a near-optimal schedule, a strict/relaxed EDD rule was embedded as a preordering condition in the problem model.

Table 1
Production order data for Example 1

Order	Release time	Due date	Processing time (days)			
			U_1	U_2	U_3	U_4
O_1	0	10	6.8			
O_2	5	22			1.8	
O_3	0	25	5		3.3	
O_4	6	20		5.1		
O_5	0	28		5.6		2.55
O_6	2	30	4.8	3.6		
O_7	3	17			2.1	3.3
O_8	0	23				12.6
O_9	2	30	3.2			
O_{10}	6	21	5.2	3.8		
O_{11}	0	30			1.1	
O_{12}	1.5	28			3.4	2.1
O_{13}	0	15	3.5		2	
O_{14}	0	29		8		
O_{15}	5.5	12		4.44		4.17
O_{16}	0	19			2.6	
O_{17}	2	30	2.6			3.15
O_{18}	1.5	27	3.6	4.35		
O_{19}	0	30		5.55	2.95	
O_{20}	1	24		3.6		3.9
Unit ready time			0	3	2	3

Table 2
Sequence-dependent setup times (in days) for Example 1

Order	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}	O_{18}	O_{19}	O_{20}
O_1			F[a]			0.65			0.85	0.4			0.35				F	0.65		
O_2			1.1				F				0.25	F	0.7			0.25			0.3	
O_3	1.0	0.15				F	0.30		1.6	0.2	0.5	0.75	F			0.9	0.6	F	F	
O_4					0.05	F				0.5				0.7	0.45			0.5	F	0.75
O_5				0.3		0.7	0.90	0.6		F		0.9		0.8	F		F	0.7	1.3	F
O_6	1.4		0.3	0.7	F				1.2	F			1.2	0.55	0.20		0.35	F	0.9	0.8
O_7		1.8	F		0.85			0.45			1.0	1.1	F		F	0.8	0.5		0.8	0.25
O_8					F		1.65					1.05			0.1		F			0.3
O_9	2.1		1.25			0.8				0.65			0.85				0.15	1.2		
O_{10}	1.5		0.6	0.75	0.5	F			0.7				1.15	1.3	0.95		0.4	1.0	F	1.25
O_{11}		0.95	F				F					0.15	0.15			0.35			0.2	
O_{12}		F	0.8		0.4		1.0	0.2			0.6		F		1.3	1.0	0.8		0.95	F
O_{13}	0.3	0.55	1.3			1.3	1.55		0.25	1.15	1.4	0.4				0.5	0.25	F	0.35	
O_{14}				1.45	0.8	0.5				0.35					0.75			0.55	0.5	0.65
O_{15}				0.2	F	0.4	1.20	0.3		0.8		0.3		1.05			0.6	0.3	F	F
O_{16}		0.25	1.05				F				F	0.85	0.2						0.15	
O_{17}	F		0.8		0.3	0.9	1.1	0.5	F	0.75		F	0.45		0.2			0.15		0.3
O_{18}	0.4		F	0.5	0.45	F			0.35	0.6			F	0.65	0.55		0.3		0.6	0.45
O_{19}		0.7	F	F	0.65	0.85	0.80			F	0.7	0.9	F	0.5	1.05	0.75		0.45		F
O_{20}				0.15	F	F	0.55	0.45		0.4		F		0.4	F		F	F	F	

[a] F, forbidden sub-sequence.

Table 3
Results for Example 1 — Case A[a]

Unit	Order	Processing time	Setup starting time	Processing initial time	Processing completion time	Due date	Lateness
U_1	O_1	6.80		0.90	7.70	10	−2.30
	O_{10}	5.20	7.70	8.10	13.30	21	−7.70
	O_{18}	3.60	13.30	14.30	17.90	27	−9.10
	O_9	3.20	17.90	18.25	21.45	30	−8.55
	O_6	4.80	21.45	22.25	27.05	30	−2.95
	O_{17}	2.60	27.05	27.40	30.00	30	0.00
U_2	O_{15}	4.44		6.51	10.95	12	−1.05
	O_4	5.10	10.95	11.15	16.25	20	−3.75
	O_{20}	3.60	16.25	17.00	20.60	24	−3.40
	O_{14}	8.00	20.60	21.00	29.00	29	0.00
U_3	O_{13}	2.00		9.50	11.50	15	−3.50
	O_7	2.10	11.50	13.05	15.15	17	−1.85
	O_{16}	2.60	15.15	15.95	18.55	19	−0.45
	O_2	1.80	18.55	18.80	20.60	22	−1.40
	O_3	3.30	20.60	21.70	25.00	25	0.00
	O_{11}	1.10	25.25	25.75	26.85	30	−3.15
	O_{19}	2.95	26.85	27.05	30.00	30	0.00
U_4	O_8	12.60		9.30	21.90	23	−1.10
	O_{12}	2.10	21.90	22.95	25.05	28	−2.95
	O_5	2.55	25.05	25.45	28.00	28	0.00

[a] Strict EDD as a preordering condition — times are expressed in days.

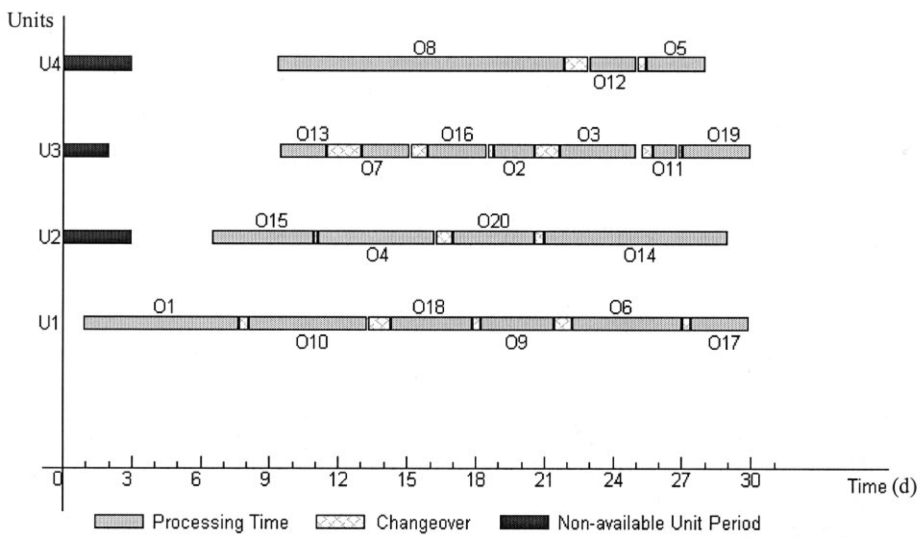

Fig. 2. Gantt diagram of the optimal scheduling for Example 1 — Case A.

Table 4
Model sizes and computational requirements for Examples 1 and 2

Example	Binary vars., cont. vars., rows	Objective function	Optimal value	CPU time[a]	Nodes	Iterations
1 — Case A	84, 105, 344	Weighted lateness	2.533	0.88	37	647
1 — Case B	90, 105, 361	Weighted lateness	2.064	11.48	1091	9891
2	185, 201, 864	Weighted lateness	3.777	22.19	476	13054

[a] Seconds on a Pentium II PC (400 MHz) with GAMS/OSL.

Table 5
Results for Example 1 — Case B[a]

Unit	Order	Processing time	Setup starting time	Processing initial time	Processing completion time	Due date	Lateness
U_1	O_1	6.80		2.10	8.90	10	−1.10
	O_{13}	3.50	8.90	9.25	12.75	15	−2.25
	O_{10}	5.20	12.75	13.90	19.10	21	−1.90
	O_{18}	3.60	19.10	20.10	23.70	27	−3.30
	O_9	3.20	23.70	24.05	27.25	30	−2.75
	O_{17}	2.60	27.25	27.40	30.00	30	0.00
U_2	O_{15}	4.44		7.46	11.90	12	−0.10
	O_4	5.10	11.90	12.10	17.20	20	−2.80
	O_{14}	8.00	17.20	17.90	25.90	29	−3.10
	O_6	3.60	25.90	26.40	30.00	30	0.00
U_3	O_7	2.10		13.05	15.15	17	−1.85
	O_{16}	2.60	15.15	15.95	18.55	19	−0.45
	O_2	1.80	18.55	18.80	20.60	22	−1.40
	O_3	3.30	20.60	21.70	25.00	25	0.00
	O_{11}	1.10	25.25	25.75	26.85	30	−3.15
	O_{19}	2.95	26.85	27.05	30.00	30	0.00
U_4	O_{20}	3.90		4.95	8.85	24	−15.15
	O_8	12.60	8.85	9.30	21.90	23	−1.10
	O_{12}	2.10	21.90	22.95	25.05	28	−2.95
	O_5	2.55	25.05	25.45	28.00	28	0.00

[a] Relaxed EDD as a preordering condition — times are expressed in days.

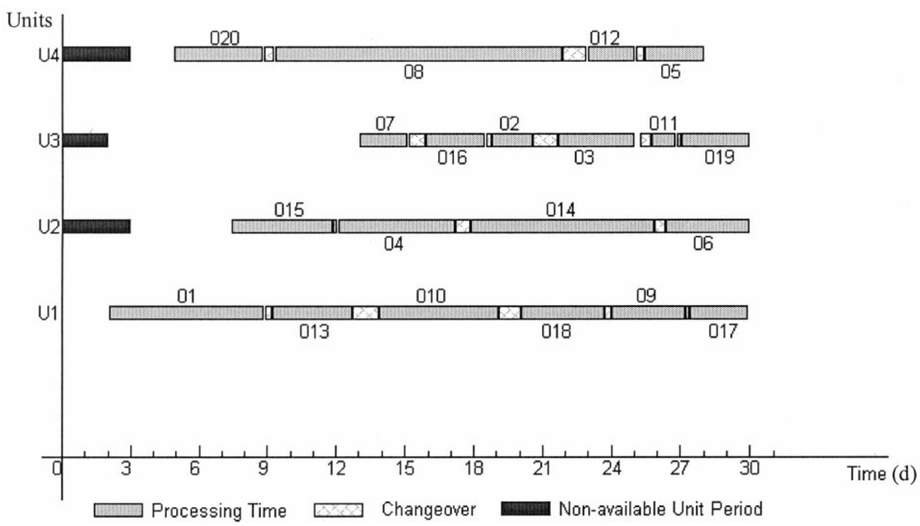

Fig. 3. Gantt diagram of the optimal scheduling for Example 1 — Case B.

In all examples, it has been observed a remarkable saving in binary variables with regards to prior continuous time approaches, and a good computational performance. This is achieved through: (a) the separate treatment of unit-assignment and sequencing decisions and (b) the use of soft preordering conditions embedded in the problem formulation. Let us consider a simple example to illustrate the impact of feature (a) on the number of 0–1 variables. Four parallel units are available to process 20 product orders and each unit can be assigned to any product. Moreover, let us suppose that there are just five feasible candidates to precede any product order or, alternatively, five time-slots have been pre-defined for every equipment item. By separately handling sequencing and unit-assignment decisions, the number of binary variables drops from 400 (if either the approach of Cerdá et al. (1997) or the one proposed by Pinto and Grossmann (1995) were applied) to 180. Note that variables W_{ij} can be regarded as continuous variables restricted to the [0, 1] interval. Despite that, they take 0–1 values only. Therefore, the new approach cuts the number of binary variables by more than a half. Such a reduction would be even sharper if the number of feasible predecessors per order or the number of time-slots pre-assigned to each unit take higher values.

8.1. Example 1

This example, first studied by Cerdá et al. (1997), is concerned with the scheduling of a single-stage multiproduct batch plant, with four units in parallel, where a total of twenty production orders are to be manufactured within a specified time horizon. Data for Example 1 are given in Tables 1 and 2. Information about each single-batch production order including release-time, due date, alternative units where it can be processed and processing times is provided in Table 1. It is observed that each order can be manufactured in some of the available units and the processing time varies with the unit. Moreover, Table 1 also indicates the ready time for each equipment item. In turn, Table 2 shows the sequence-dependent setup times for each production order, regardless of the unit in which it is processed. For example, the setup time for order O_1 drops from 2.1 to 1.5 when O_{10} rather than O_9 is previously processed in the assigned unit. Setup times for prohibited sub-sequences $(i, i')\in FS$ are not reported in Table 2. This is the case of sub-sequence (O_1, O_{17}) whose related cell shows a letter F to indicate that is forbidden. Though the orders O_1 and O_{17} can both be processed in unit U_1, operational constraints dictate that O_{17} never directly succeeds O_1. In turn, a blank indicates that the related pair of orders (i, i') can never be allocated to the same equipment item because $J_{ii'}$ is an empty set. Since the plant schedule is developed for a 1-month period, all the orders must be completed within a time horizon of 30 days.

To reduce the problem size and speed up the convergence to the optimal solution, the strict EDD rule was applied as a preordering condition (Case A). Therefore, in addition to the specified prohibited sub-sequences, a pair of orders (i, i') will belong to the forbidden set FS whenever $d_i > d_{i'}$ and $J_{ii'}$ is not an empty set. The problem formulation now includes 84 binary variables, 105 continuous variables and 344 linear constraints. Since the problem objective is to maximize the customer satisfaction at minimum inventory cost, then the weighted lateness given by (18) was chosen as the objective function. The optimal schedule is reported in Table 3 and Fig. 2. A negative value for the order lateness means that the order was completed earlier than required. Instead, a positive value indicates that the order is tardy. The best schedule for Example 1 (Case A) presents a null tardiness and an average earliness of 2.66 days. It was found in 0.88 s and 37 nodes were enumerated (see Table 4). Table 3 shows not only the time at which the unit setup before processing begins but also the starting time of the actual processing itself. If the EDD rule is not embedded in the problem formulation, then the number of binary variables rises from 105 to 157, and the number of constraints from 344 to 542. On the other hand, if the formulation of Pinto and Grossmann (1995) were applied to solve this example and six time slots per unit were considered, then 1130 additional binary variables would be required to just account for sequence-dependent changeovers (Pinto & Grossmann, 1998).

To improve the proposed schedule for the problem, the preordering condition was relaxed so as to consider a larger solution space. This time, sub-sequences of orders (i, i') will belong to FS whenever $d_i > d_{i'} + 1$ day (relaxed EDD) and $J_{ii'}$ is not an empty set. Consequently, the number of binary variables and the number of constraints will both rise. The optimal schedule was found in 11.5 s and 1091 nodes were enumerated (see Table 4). Selected order-unit assignments are somewhat different with regards to Case A (see Table 5 and Fig. 3). It now features a null tardiness and an average order earliness of merely 2.16 days, smaller than the one found with the strict EDD (2.66 days) and also well below 6.66 days previously reported by Cerdá et al. (1997). Since these authors chose the overall tardiness as the problem objective, they really made no effort to reduce the average earliness. As shown by Figs. 2 and 3, the processing of orders at each unit is properly delayed so that all of them are consecutively processed without any interruption and, consequently, the average earliness is reduced as much as possible.

Table 6
Product families defined for Example 2

Family ID	Tint	Products
F1	White–cream	P1, P2, P3, P4
F2	Yellow–orange	P5, P6, P7, P8, P9, P10
F3	Red	P11, P12, P13, P14, P15
F4	Brown	P16, P17, P18
F5	Green	P19, P20, P21, P22
F6	Blue	P23, P24, P25, P26, P27, P28
F7	Dark gray–black	P29, P30, P31, P32

Table 7
Setup time for each family (Example 2)

Family	Setup times (h)				
	U1	U2	U3	U4	U5
F1	0.7	0.7	1		
F2		0.7	1.2		
F3	1.2	1.2		0.9	
F4				0.8	0.6
F5			1.2	1.2	0.7
F6	0.5		0.5	1	0.9
F7			0.9		0.6

Table 8
Product processing rates for Example 2

Family ID	Product ID	Processing rates (t/h)				
		U1	U2	U3	U4	U5
F1	P1	1	1	1.25		
	P2	0.857	0.857	1.059		
	P3	0.808	0.808	0.955		
	P4	0.666	0.666	0.923		
F2	P5		0.882	1		
	P6-P7		1.1	1.20		
	P8		1	1.25		
	P9-P10		0.957	1.048		
F3	P11	0.885	0.885		0.958	
	P12	0.9	0.9		0.857	
	P13	0.91	0.91		1.053	
	P14	0.933	0.933		0.848	
	P15	0.889	0.889		0.8	
F4	P16				0.962	1.042
	P17				1.02	1.115
	P18				0.964	0.871
F5	P19			1.053	0.77	0.87
	P20			1.09	0.889	1
	P21			1.083	1.04	0.929
	P22			1.235	0.808	0.913
F6	P23	0.824		1.077	0.824	0.933
	P24	0.9		1.059	0.783	0.857
	P25	0.958		1.095	0.958	0.958
	P26	0.944		1.062	0.85	0.944
	P27	1.038		1.227	1.08	1.227
	P28	0.95		1	0.95	1
F7	P29			0.91		0.91
	P30			0.95		0.95
	P31-P32			1.167		0.824

8.2. Example 2

This example addresses the scheduling problem faced in a PVC compounding plant having one limiting processing stage and five extruders in parallel. The plant that operates 6 days a week, three shifts per day, should manufacture 25 different colored plastics the next week. As shown in Table 6, plastics have been grouped into seven families according to their main tint. For example, different types of yellow and orange compounds (P5 to P10) were grouped into family F2. Products from families F1 and F2 only differ in one color degree, those from families F1 and F3 in two color degrees and so on. Setup times in hours and products' processing rates in ton/h have been included in Tables 7 and 8, respectively. From such tables, it can be noted that (a) each family can be processed just at certain extruders and (b) though varying with the unit, the setup time takes a similar value for any member of a family. For instance, clear colors can be mostly processed in units 1 and 2, while dark colors are usually assigned to extruders 4 and 5. Instead, the extruder 3 is a general-purpose piece of equipment that can be used for processing a wide variety of families. On the other hand, Table 9 depicts the sequence-dependent changeover times. As indicated with a letter *F*, there are many forbidden sub-sequences. Such plant operating constraints come from the fact that important changeover times and

Table 9
Changeover times between families (Example 2)

	Changeover times (h)						
	F1	F2	F3	F4	F5	F6	F7
F1	0	1.5	1.7		2.3	2.7	3
F2	4.1	0	1.1		1.6	1.9	2.1
F3	F[a]	3	0	1.5	1.8	1.9	
F4	F	F	3.1	0	1.5	1.6	1.9
F5	F	F	F	3.3	0	1.4	1.7
F6	F	F	F	F	3.3	0	1.4
F7	F	F	F	F	F	3.5	0

[a] F, forbidden sub-sequence.

Table 10
Product order data for Example 2

Order	Product	Size (t)	Due date	
			Day-shift[a1]	Hour
O1	P1	20	Mo-N	24
O2	P2	18	Tu-N	48
O3	P3	21	Th-M	80
O4	P4	12	Sa-M	128
O5	P5	15	Tu-N	48
O6	P10	22	Fr-N	120
O7	P11	23	Tu-M	32
O8	P12	18	We-A	64
O9	P13	20	Th-M	80
O10	P14	28	Fr-N	120
O11	P15	16	Sa-N	144
O12	P16	25	Tu-A	40
O13	P18	27	Sa-M	128
O14	P19	20	Tu-A	40
O15	P20	24	We-N	72
O16	P21	26	Th-M	80
O17	P22	21	Sa-N	144
O18	P23	14	Mo-N	24
O19	P24	18	Tu-M	32
O20	P25	23	We-N	72
O21	P26	17	Th-N	96
O22	P27	27	Th-A	88
O23	P28	19	Fr-A	112
O24	P29	20	We-N	72
O25	P32	14	Fr-M	104

[a] M, morning; A, afternoon; N, night.

costs are incurred when the processing in a given unit goes from dark to clear colors. Therefore, the following rule is enforced in the plant. The consecutive processing of two products belonging to two different families (f, f') is forbidden whenever f' is clearer than f in more than one tint degree. While the consecutive processing of products belonging to families F4 and F3 is then allowed, with a changeover time of 3.1 h, the successive manufacture of products pertaining to families F4 and F2 or F4 and F1 is instead not permitted. In other words, there is no constraint in going from clear to dark colors but restrictions will apply if there is a decrease in more that one color degree.

During a 6-day time horizon, 25 orders each involving a different product have to be scheduled. Production order data including product to be manufactured, quantity and due date are given in Table 10. Since the plant is operated following a just-in-time policy, then the minimum weighted lateness has been adopted as the problem objective. As shown in Table 10, order due-dates always happen at the end of a shift and are expressed in hours with regards to the start of the time horizon (Monday, 06:00 h). Moreover, all orders and equipment items are ready at the beginning of the time horizon. To get a reasonable problem size, the strict EDD rule was embedded as a preordering condition in the mathematical formulation. The optimal schedule for this case study is depicted in Table 11, and the Gantt chart representation is shown in Fig. 4. It requires a CPU time of 22.2 s and 1091 nodes were enumerated. Even though sequence-dependent setup times were considered, the problem model comprises 185 binary variables, 201 continuous variables and 864 constraints (Table 4) well below the number of variables and constraints reported by other continuous-time approaches for examples of similar size.

An analysis of the reported solution reveals that it features no tardy orders, 12 on time and 13 completed earlier than the specified due-date. Among the early orders, only three have an earliness over 10 h. Moreover, the optimal schedule has a makespan of 144 h and a mean order earliness of 3.9 h. In addition, there is a good load balance among the units and it often occurs that a product is clearer than its predecessor in the processing sequence. However, the operating rule stated before is never violated since in all cases the succeeding product is clearer in just one tint degree. For instance, the first product manufactured in extruder 2 belongs to family F3, the second to family F2 and the third to family F1. Since no order is late and the mean earliness is really low, relaxation of the EDD rule to further improve the proposed schedule was not really required.

8.3. Example 3

This example deals with the production of eight different intermediate products in a single-stage multiproduct batch plant with seven units in parallel and a fixed time horizon of seven days. Every entry in Table 12 stands for a product order to be satisfied in the scheduling period, i.e. a total of 29 orders. Each product requirement is not only characterized by the product and the amount being requested but by its due date as well. Most of the products feature several requirements at different due dates. For instance, there are five orders for product P4 involving 4950, 5053, 1827, 7069 and 4101 kg, respectively.

Due dates always happen at the end of the last daily shift. For example, if day 3 is the due date of a particular order, it should then be completed at most 72 h after the start of the time horizon. Table 13 shows the batch size for each product at any unit where it can be processed. A blank in Table 13 means that the unit cannot be assigned to the product. From such a Table, it follows that each product requirement does not generally match up with the size of a batch. For instance, P4 features batch sizes equal to 6000 or 5000 kg depending on the unit being assigned. Therefore, none of the orders involving P4 exactly corresponds to a single product batch. To get a better utilization of the plant capacity, it is a common industrial practice to allocate a single batch to two or more orders requiring the same product but due at different times. Furthermore, Table 14 provides the batch processing time for every product assuming that it depends on neither the batch size nor the equipment item. As usually found in industry, a small number of batch sizes are handled. In this case: {6000 kg, 5000 kg, 4500 kg}. Product order data are again displayed in the first four columns of Table 15 but in this case the due dates have been expressed in hours with regards to the beginning of the time horizon. To diminish the problem size, all the units $j \in J_s$ processing the same subset of products $P_s \subseteq P$ and featuring a similar batch size B_{ps} and processing time

Table 11
Optimal production schedule for Example 2[a1]

Unit	Order	Processing time	Setup starting time	Processing initial time	Processing completion time	Due date	Lateness
U_1	O_1	20.00	2.40	3.10	23.10	24	−0.90
	O_2	21.00	23.10	23.80	44.80	48	−3.20
	O_{20}	24.00	44.80	48.00	72.00	72	0.00
	O_{21}	18.00	73.00	73.50	91.50	96	−4.50
	O_{23}	20.00	91.50	92.00	112.00	112	0.00
U_2	O_7	26.00	0.10	1.30	27.30	32	−4.70
	O_5	17.00	27.30	31.00	48.00	48	0.00
	O_3	26.00	49.20	54.00	80.00	80	0.00
	O_6	23.00	80.00	82.20	105.20	120	−14.18
	O_4	18.00	105.20	110.00	128.00	128	0.00
U_3	O_{18}	13.00	0.40	0.90	13.90	24	−10.10
	O_{14}	19.00	13.90	18.40	37.40	40	−2.60
	O_{24}	22.00	37.40	40.00	62.00	72	−10.00
	O_{22}	22.00	62.00	66.00	88.00	88	0.00
	O_{25}	12.00	89.70	92.00	104.00	104	0.00
U_4	O_{12}	26.00	8.30	9.10	35.10	40	−4.90
	O_8	21.00	35.10	39.10	60.10	64	−3.90
	O_9	19.00	60.10	61.00	80.00	80	0.00
	O_{10}	33.00	86.10	87.00	120.00	120	0.00
	O_{11}	20.00	123.10	124.00	144.00	144	0.00
U_5	O_{19}	21.00	1.40	2.30	23.30	32	−8.70
	O_{15}	24.00	23.30	27.30	51.30	72	−20.70
	O_{16}	28.00	51.30	52.00	80.00	80	0.00
	O_{13}	31.00	83.90	87.80	118.80	128	−9.20
	O_{17}	23.00	118.80	121.00	144.00	144	0.00

[a] Times are expressed in hours.

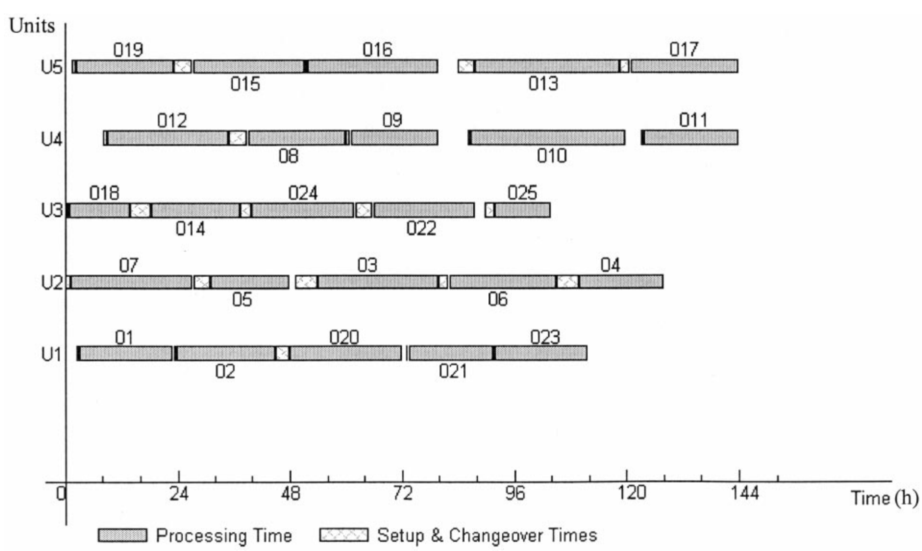

Fig. 4. Gantt diagram of the optimal scheduling for Example 2.

Table 12
Product requirements (kg) and due dates for Example 3

Product	Due date (day)						
	1	2	3	4	5	6	7
P_1			18 000				
P_2		6000	6000				
P_3		6983	2014		3003		
P_4		4950	5053	1827	7069		4101
P_5	2301		3699				
P_6		1254	3627	1036	3032		6051
P_7	1111	3765	3765	1255	3765		4339
P_8	1680	1680	1680	420		6540	

θ_{ps} for each product $p \in P_s$ have been grouped into a single cluster $s \in S$. Example 3 comprises three clusters of units: $S = \{\{U1 - U4\}, \{U5 - U6\}, \{U7\}\}$.

The MILP batching problem model for Example 3 involving 29 orders and seven units in parallel comprises 72 integer variables, 30 continuous variables and 48 linear constraints. The optimal solution to the batching problem is shown on the right of Table 15 under the *production* and *work-in-process inventory* labels. It was found in 107.8 s and 17769 nodes were enumerated. Results depicted in Table 15 indicate that order O_1 will be satisfied by processing three batches of 6000 kg in any of the units featuring such a batch size for product P1, i.e. units U1–U4 and U7. Those batches will all be entirely allocated to O_1 and feature a common due date $d = 72$ h, i.e. the due date of order O_1. A different situation arises for order O_4 involving a requirement of product P3 for 6983 kg. In this case, a pair of batches of 6000 kg will be manufactured to fulfill order O_4, thus leaving an inventory of 5017 kg. Such an inventory of P3 is then allocated to the later orders O_5 and O_6 requesting the same product P3. The two batches will have the same due date as the order O_4 ($d = 48$ h), but one of them will be completely allocated to order O_4 while the other one will be shared by orders O_4 (983 kg), O_5 (2014 kg) and O_6 (3003 kg).

However, the relevant information for the short-term scheduling problem to be subsequently solved is just the set of batches to be processed, their corresponding sizes and due dates, and the alternative units where each one can be produced as well. For instance, $NB_{psd} = 3$ means that three batches of size B_{ps} are to be processed in some units featuring such a batch size for product p to meet the order due at time d. Therefore, such batches will all have a common due date d, a batch size B_{ps} and each one can be processed in any unit $j \in J_p$ featuring such a batch size for product p. Table 16 shows sizes, products and due dates for the 21 batches to be manufactured to timely meet the 29 orders at minimum inventory cost. Four of 4500 kg, four of 5000 kg and 13 of 6000 kg are to be processed during the scheduling period. Table 17 includes the sequence-dependent setup times for the short-term scheduling problem. The optimal production schedule for Example 3 that minimizes the weighted lateness is depicted in Table 18 and Fig. 5. Model sizes and CPU times are given in Table 19. From Table 18, it follows that every batch is produced just in time. However, every equipment item is idle during a large fraction of the time horizon. So, it is better to process the batches by running a lower number of units. Table 20 and Fig. 6 show the optimal schedule when four rather than seven units are run. All the batches are still manufactured without tardiness but just 14 are produced right on time. As already revealed by Table 15, there are many instances where a batch is shared by several orders involving the same product but different due dates.

Table 13
Product batch sizes at each unit (Example 3)

Units	Products							
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
U_1	6000	6000	6000	6000	6000	6000	6000	6000
U_2	6000	6000	6000	6000	6000	6000	6000	6000
U_3	6000	6000	6000	6000	6000	6000	6000	6000
U_4	6000	6000	6000	6000	6000	6000	6000	6000
U_5				5000	5000	5000	4500	
U_6				5000	5000	5000	4500	
U_7	6000	6000	6000	6000		6000	4500	6000

Table 14
Batch processing times for Example 3

Product	Processing time (h/batch)
P_1	8
P_2	10
P_3	12
P_4	12
P_5	8
P_6	16
P_7	12
P_8	20

Table 15
Optimal solution for the batching problem (Example 3)

Order	Product	Due date (day–hour)	Requirements (kg)	Production 6000 kg	Production 5000 kg	Production 4500 kg	Work-in-process inventory
O_1	P_1	3–72	18 000	18 000			
O_2	P_2	2–48	6000	6000			
O_3	P_2	3–72	6000	6000			
O_4	P_3	2–48	6983	12 000			5017
O_5	P_3	3–72	2014				3003
O_6	P_3	5–120	3003				
O_7	P_4	2–48	4950	6000			1050
O_8	P_4	3–72	5053	6000			1997
O_9	P_4	4–96	1827				170
O_{10}	P_4	5–120	7069	6000	5000		4101
O_{11}	P_4	6–144	4101				
O_{12}	P_5	1–24	2301	6000			3699
O_{13}	P_5	3–72	3699				
O_{14}	P_6	2–48	1254		5000		3746
O_{15}	P_6	3–72	3627				119
O_{16}	P_6	4–96	1036		5000		4083
O_{17}	P_6	5–120	3032				1051
O_{18}	P_6	6–144	6051		5000		
O_{19}	P_7	1–24	1111			4500	3389
O_{20}	P_7	2–48	3765			4500	4124
O_{21}	P_7	3–72	3765				359
O_{22}	P_7	4–96	1255			4500	3604
O_{23}	P_7	5–120	3765			4500	4339
O_{24}	P_7	6–144	4339				
O_{25}	P_8	1–24	1680	6000			4320
O_{26}	P_8	2–48	1680				2640
O_{27}	P_8	3–72	1680				960
O_{28}	P_8	4–96	420				540
O_{29}	P_8	5–120	6540	6000			

Table 16
Optimal set of batches to be processed (Example 3)

Batch	Size (kg)	Product	Due date (h)	Processing time (h)	Available units
B_1	6000	P_1	72	8	U_1, U_2, U_3, U_4, U_7
B_2	6000	P_1	72	8	U_1, U_2, U_3, U_4, U_7
B_3	6000	P_1	72	8	U_1, U_2, U_3, U_4, U_7
B_4	6000	P_2	48	10	U_1, U_2, U_3, U_4, U_7
B_5	6000	P_2	72	10	U_1, U_2, U_3, U_4, U_7
B_6	6000	P_3	48	12	U_1, U_2, U_3, U_4, U_7
B_7	6000	P_3	48	12	U_1, U_2, U_3, U_4, U_7
B_8	6000	P_4	48	12	U_1, U_2, U_3, U_4, U_7
B_9	6000	P_4	72	12	U_1, U_2, U_3, U_4, U_7
B_{10}	6000	P_4	120	12	U_1, U_2, U_3, U_4, U_7
B_{11}	5000	P_4	120	12	U_5, U_6
B_{12}	6000	P_5	24	8	U_1, U_2, U_3, U_4
B_{13}	5000	P_6	48	16	U_5, U_6
B_{14}	5000	P_6	96	16	U_5, U_6
B_{15}	5000	P_6	144	16	U_5, U_6
B_{16}	4500	P_7	24	12	U_5, U_6, U_7
B_{17}	4500	P_7	48	12	U_5, U_6, U_7
B_{18}	4500	P_7	96	12	U_5, U_6, U_7
B_{19}	4500	P_7	120	12	U_5, U_6, U_7
B_{20}	6000	P_8	24	20	U_1, U_2, U_3, U_4, U_7
B_{21}	6000	P_8	120	20	U_1, U_2, U_3, U_4, U_7

Table 17
Sequence-dependent changeover times for Example 3[a1]

Product	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
P_1	0.0	1.5	1.6	2.7	2.4	4	5	2
P_2	5.1	0.0	1.3	4.8	2.1	3.4	6.1	1.2
P_3	1.6	2.3	0.0	1.4	2.5	3.5	1.4	4.5
P_4	1.0	2.5	2.1	0.0	5.9	1.4	1.7	1.6
P_5	1.5	3.1	4.5	1.4	0.0	1.7	5.0	4.2
P_6	4.1	4.1	3.0	1.0	1.4	0.0	4.1	1.6
P_7	3.2	2.3	3.4	1.1	2.1	1.4	0.0	3.2
P_8	2.7	1.6	3.9	1.5	1.4	1.8	1.5	0.0

[a] Expressed in hours.

9. Conclusions

A new MILP mathematical formulation for the short-term scheduling of a multiproduct batch plant involving a unique processing stage with several units in parallel has been presented. It assumes that the set of batches to be scheduled within a fixed time horizon is a problem data. It is based on a continuous time domain representation not relying on the definition of time slots or events. Instead, the proposed formulation takes advantage of the linear precedence structure of the processing sequence at each unit. To get a significant saving in 0–1 variables, assignment and sequencing decisions are separately handled. Moreover, the proposed approach can accommodate positive release times for orders and units as well as sequence-dependent changeovers and forbidden processing sub-sequences without affecting the number of binary variables and constraints. In contrast to prior methods (Pinto & Grossmann, 1995), a significant decrease of the problem size is observed when dedicated units and forbidden sub-sequences are considered. Interestingly, the computational effort will not depend at all on the number of time slots or time events to be defined beforehand. In addition, relaxed pre-ordering conditions can be easily embedded into the mathematical model to further cut down the problem size and still find a very good schedule.

Similarly to other continuous time methodologies, the proposed MILP short-term scheduling model has been formulated in terms of product orders rather than batches by assuming that each order involves a single batch. In industry, however, product orders usually comprise several batches of different sizes that are often processed at various units and even at the same time. Usually, there are several orders for a particular product featuring different due dates. To tackle such real-world batch scheduling problems, the so-called MILP product batching problem has also been introduced to optimally convert product orders into batches before solving the batch scheduling formulation. In this way, the short-term scheduling problem can still be expressed in terms of batches rather than product orders. Moreover, if the orders do not comprise an integer number of batches, then some batches can be allocated to multiple orders so as to get a more effective utilization

Table 18
Optimal schedule for Example 3[a1]

Unit	Order	Processing time	Setup starting time	Processing initial time	Completion time	Due date	Lateness
U_1	B_{12}	8.00	16.00	16.00	24.00	24.00	0.00
	B_9	12.00	58.60	60.00	72.00	72.00	0.00
	B_{21}	20.00	98.40	100.00	120.00	120.00	0.00
U_2	B_7	12.00	36.00	36.00	48.00	48.00	0.00
	B_3	8.00	62.40	64.00	72.00	72.00	0.00
U_3	B_6	12.00	36.00	36.00	48.00	48.00	0.00
	B_5	10.00	59.70	62.00	72.00	72.00	0.00
U_4	B_{20}	20.00	4.00	4.00	24.00	24.00	0.00
	B_8	12.00	34.50	36.00	48.00	48.00	0.00
	B_2	8.00	63.00	64.00	72.00	72.00	0.00
	B_{10}	12.00	105.30	108.00	120.00	120.00	0.00
U_5	B_{16}	12.00	12.00	12.00	24.00	24.00	0.00
	B_{17}	12.00	36.00	36.00	48.00	48.00	0.00
	B_{11}	12.00	106.90	108.00	120.00	120.00	0.00
	B_{15}	16.00	126.60	128.00	144.00	144.00	0.00
U_6	B_{13}	16.00	32.00	32.00	48.00	48.00	0.00
	B_{14}	16.00	80.00	80.00	96.00	96.00	0.00
	B_{19}	12.00	103.90	108.00	120.00	120.00	0.00
U_7	B_4	10.00	38.00	38.00	48.00	48.00	0.00
	B_1	8.00	58.90	64.00	72.00	72.00	0.00
	B_{18}	12.00	79.00	84.00	96.00	96.00	0.00

[a] Times are expressed in hours.

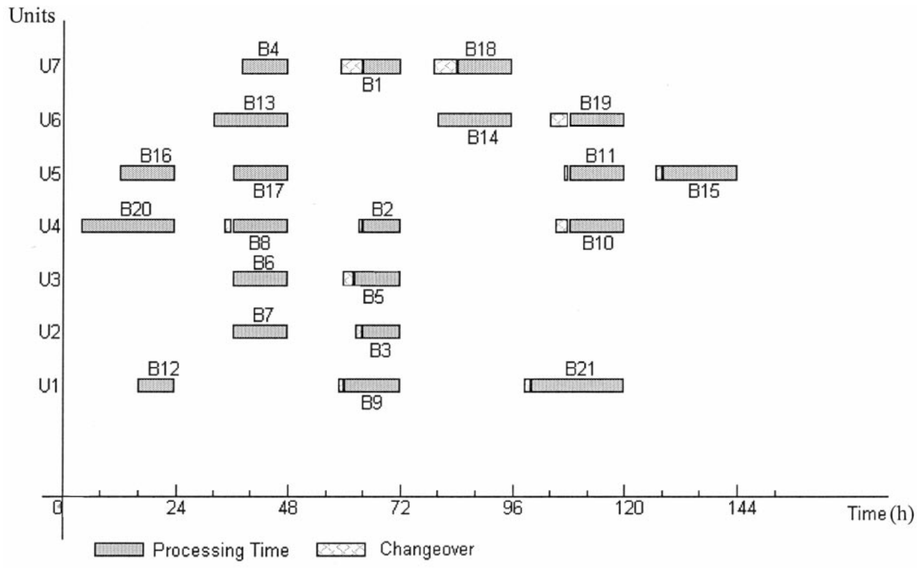

Fig. 5. Gantt diagram of the optimal scheduling for Example 3.

Table 19
Computational results for Example 3

Problem	Integer vars., cont. vars., rows	Objective function	Optimal value	CPU time[a1]	Nodes	Iterations
Batching	72, 30, 48	inventory	59013 kg	107.773	17769	17876
Scheduling (7 units)	203, 211, 1170	weighted lateness	0.00	25.16	657	11016
Scheduling (4 units)	203, 143, 827	weighted lateness	3.927	12.68	426	7797

[a] Seconds on a Pentium II PC (400 MHz) with GAMS/OSL.

Table 20
Optimal schedule for Example 3[a2] (by running only four units)

Unit	Order	Processing time	setup starting time	Processing starting time	Processing completion time	Due date	Lateness
U_1	B_{12}	8.00	16.00	16.00	24.00	24.00	0.00
	B_7	12.00	26.40	30.90	42.90	48.00	−5.10
	B_5	10.00	42.90	45.20	55.20	72.00	−16.80
	B_9	12.00	55.20	60.00	72.00	72.00	0.00
U_2	B_4	10.00	11.30	11.30	21.30	48.00	−26.70
	B_6	12.00	21.30	22.60	34.60	48.00	−13.40
	B_8	12.00	34.60	36.00	48.00	48.00	0.00
	B_2	8.00	55.00	56.00	64.00	72.00	−8.00
	B_1	8.00	64.00	64.00	72.00	72.00	0.00
	B_{21}	20.00	98.00	100.00	120.00	120.00	0.00
U_5	B_{16}	12.00	12.00	12.00	24.00	24.00	0.00
	B_{13}	16.00	30.60	32.00	48.00	48.00	0.00
	B_{14}	16.00	77.30	77.30	93.30	96.00	−2.70
	B_{11}	12.00	93.30	94.30	106.30	120.00	−13.70
	B_{19}	12.00	106.30	108.00	120.00	120.00	0.00
	B_{15}	16.00	126.30	128.00	144.00	144.00	0.00
U_7	B_{20}	20.00	4.00	4.00	24.00	24.00	0.00
	B_{17}	12.00	34.50	36.00	48.00	48.00	0.00
	B_3	8.00	60.80	64.00	72.00	72.00	0.00
	B_{18}	12.00	79.00	84.00	96.00	96.00	0.00
	B_{10}	12.00	106.90	108.00	120.00	120.00	0.00

[a] Times are expressed in hours.

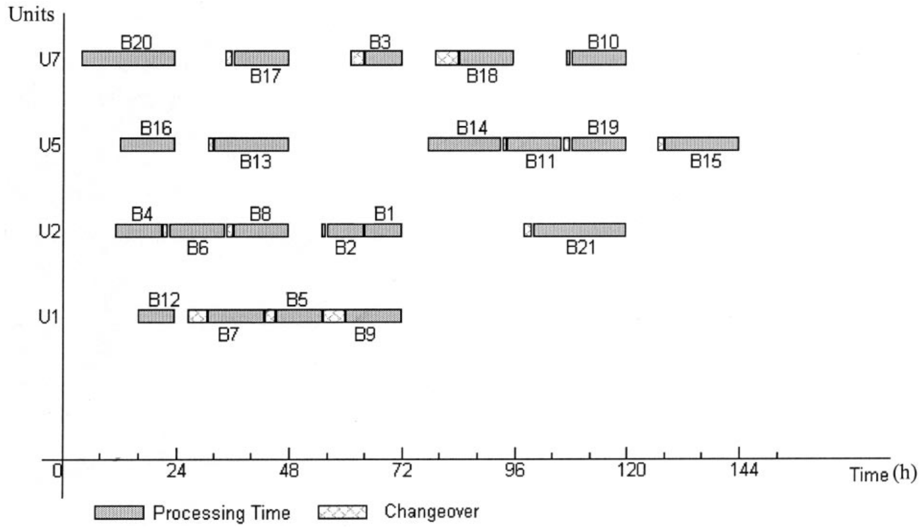

Fig. 6. Gantt diagram of the optimal scheduling for Example 3 (by running only four units).

of the plant capacity. Additional tools are also provided to refine the set of product orders (sizes and due dates) to be processed before applying the scheduling algorithm. Such a prior refinement of the specified production orders and due dates is usually necessary because of a rough estimation of the plant capacity requirements and/or a poor choice of order due dates.

Three real-world examples have been successfully tackled by solving low-size MILP mathematical models in a rather small computing time. Comparison with other approaches reveals an important reduction in binary variables and constraints, and a much better computational performance, especially if sequence-dependent changeovers and forbidden subsequences are to be considered.

Acknowledgements

The authors acknowledge financial support from FONCYT under Grant 14-00000-00356, and from 'Universidad Nacional del Litoral' under CAI + D 048 and CAI + D 121.

References

Applequist, G., Samikoglu, O., Pekny, J., & Reklaitis, G. V. (1997). Issues in the use, design and evolution of process scheduling and planning systems, *ISA Transactions*, *36* (2), 81.

Brooke, A., Kendrick, D., & Meeraus, A. A. (1992). *GAMS — a user's guide (release 2.25)*. San Francisco, CA: The Scientific Press.

Cerdá, J., Henning, G. P., & Grossmann, I. E. (1997). A mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. *Industrial Engineering and Chemical Research*, *36*, 1695.

Karimi, I. A., & McDonald, C. M. (1997). Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling. *Industrial Engineering and Chemical Research*, *36*, 2701.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operation — I. MILP formulation. *Computers and Chemical Engineering*, *17*, 211.

Ierapetritou, M. G., & Floudas, C. A. (1998). Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Industrial Engineering and Chemical Research*, *37*, 4341.

Gooding, W. B., Pekny, J. F., & McCroskey, P. S. (1994). Enumerative approaches to parallel flowshop scheduling via problem transformation. *Computers and Chemical Engineering*, *18*, 909.

Mockus, L., & Reklaitis, G. V. (1997). Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Computers and Chemical Engineering*, *21*, 1147.

OSL (1991). *OSL, guide and reference (release 2)*, Kingston, NY: IBM.

Pinto, J. M., & Grossmann, I. E. (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial Engineering and Chemical Research*, *34*, 3037.

Pinto, J. M., & Grossmann, I. E. (1996). A continuous time MILP model for short term scheduling of batch plants with pre-ordering constraints. *Computers and Chemical Engineering*, *20*, S1197.

Pinto, J. M., & Grossmann, I. E. (1998). Assignment and sequencing models for the scheduling of process systems. *Annals of Operations Research*, *81*, 433.

Reklaitis, G. V. (1992), Overview of scheduling and planning of batch process operations. *NATO advanced study institute-batch process systems engineering*, Antalya, Turkey.

Schilling, G., & Pantelides, C. C. (1996). A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers and Chemical Engineering*, *20*, S1221.

Shah, N. (1998). Single and multisite planning and scheduling: current status and future challenges. In: J. Pekny, G. Blau, *Foundations of computer aided process operations*, American Institute of Chemical Engineers Symposium Series 320, 94, pp. 91–110.

Shah, N., Pantelides, C. C., & Sargent, R. W. H. (1993). A General algorithm for short-term scheduling of batch operations — II. Computational issues. *Computers and Chemical Engineering*, *17*, 229.

Zhang, X., & Sargent, R. W. H. (1994). The optimal operation of mixed production facilities — a general formulation and some approaches for the solution. In: Proceedings of the 5th international symposium on process systems engineering, Kyongju, Korea, pp. 171.

Zentner, M. G., & Pekny, J. F. (1994). Learning to solve process scheduling problems: the role of rigorous knowledge acquisition frameworks. *Foundations of Computer Aided Process Operations*, 275.