6482 *Ind. Eng. Chem. Res.* **2010,** *49,* 6482–6498

Mixed-Integer Linear Programming Monolithic Formulations for Lot-Sizing and Scheduling of Single-Stage Batch Facilities

Pablo A. Marchetti, Carlos A. Méndez, and Jaime Cerdá*

INTEC (Universidad Nacional del Litoral−CONICET), Güemes 3450−3000 Santa Fe, Argentina

This paper presents a pair of mixed-integer linear programming (MILP) continuous-time formulations for the simultaneous lot-sizing and scheduling of single-stage multiproduct batch facilities. Both approaches can handle multiple customer orders per product at different due dates as well as variable processing times. To match product demands, several batches can be allocated to a single requirement and, at the same time, a single batch may be used to satisfy multiple orders. Through a novel procedure, a predefined set of batches for each order with enough elements to guarantee optimality is generated. The two proposed formulations deal with batch sequencing decisions in a different manner. One of them rigorously arranges individual batches assigned to the same unit, while the other sequences clusters of batches sharing the same product and due date, and processed in the same equipment item. Grouping batches into clusters seeks to reduce the number of product changeovers. The final contents of clusters are model decisions. Powerful symmetry breaking constraints based on allocation variables to avoid redundant solutions were also developed. Three cases studies involving up to 56 batches have been solved. The two formulations provide very good results at quite competitive CPU times when compared with prior monolithic techniques. Moreover, the approximate cluster-based method was able to solve very large problems in an efficient manner. It was validated by comparing its results with the ones provided by the rigorous model.

1. Introduction

Over the last two decades a considerable body of research has been done on short-term scheduling of batch manufacturing facilities. Most approaches can be regarded as rigorous solution methods relying on mathematical programming models. Extensive reviews can be found in Floudas and Lin[1] and Méndez et al.[2] Depending on whether or not the set of batches and their sizes are problem data, scheduling methodologies can be broadly classified into two separate groups: monolithic and sequential approaches.[2] Monolithic methods are those simultaneously selecting: (a) the set of batches to be scheduled and their sizes, (b) the assignment of processing tasks to production units/resources, (c) the sequencing of tasks allocated to the same resource item, and (d) the starting and completion time for each task. Moreover, they can cope with multipurpose batch scheduling problems involving complex product recipes and arbitrary network processes. In contrast, sequential approaches are specifically proposed for multistage batch facilities performing sequential processes. These methodologies mostly assume that the number and size of the batches to be processed are known beforehand, i.e. the lot-sizing problem has already been solved. In other words, the whole scheduling problem is decomposed into two separate stages with the first one solving the lot-sizing problem and the second just focused on batch scheduling. Sequential scheduling approaches also rely on the assumption that the identity of every batch is preserved throughout the whole processing system. Then, batch mixing and splitting, and material recycles are not permitted. In this way, material balance equations can be omitted and, consequently, a lower number of variables and constraints are required to model the problem. Since batch sizes are also known beforehand, task processing times are assumed to be unit-dependent fixed data.

Most monolithic approaches use state−task network (STN) or resource-task network (RTN) formulations, but different time representations. Some authors have developed scheduling methods based on a discrete representation of the time domain.[3–5] They used a fixed-time common grid for all production resources, and every task is committed to begin/end at the lower/upper limit of some predefined time-interval. Since state/resource balance equations are included, discrete time methods easily handle batch mixing and splitting operations and processes with material recycling. Other monolithic models are based on a continuous time representation with a variable time grid. In this case, a finite number of time events, either shared by all production resources[6–8] or unit-specific[9–11] are defined. Since less time points are needed with a variable time grid, a significant reduction on the model size can be achieved. Besides, continuous time formulations do not need rounding the problem data to reduce the grid size as discrete time approaches do. However, they include an important model parameter, i.e. the maximum number of time events. It may happen that the optimal solution is not found due to an insufficient number of events. As reported in the literature, problems requiring more than 15 event points become very difficult to solve because of the large CPU time needed to prove optimality.[7]

On the other hand, sequential methodologies assume that the set of batches for each product and their sizes, together with the batch release times and due dates are all known beforehand. Proposed formulations use different continuous-time representations such as time-slots,[12] unit-specific time grids,[13] direct precedence,[14,15] and global precedence.[16] A two-stage approach for the combined batching and scheduling of single-stage multiproduct batch facilities can be found in the work of Méndez et al.[17] For the lot-sizing problem, they used specific constraints to ensure that enough batches of adequate size are chosen to satisfy all production orders at minimum inventory cost. After selecting the set of batches, they next solved a direct-precedence continuous-time formulation to find the best production schedule.

In the last years, attention has been paid to the development of monolithic methods for sequential processes. Lim and

* To whom correspondence should be addressed. E-mail: jcerda@intec.unl.edu.ar.

10.1021/ie100054h
Published on Web 06/16/2010

Karimi[18] proposed a general mixed-integer linear programming (MILP) model that uses unit-dependent time slots to determine both the set of product batches and the batch schedule all at once, even if multiple orders per product are to be handled. Their model is suited for single-stage batch scheduling problems with parallel nonidentical units. Méndez and Cerdá[19] presented an effective precedence-based approach that integrates both lot-sizing and batch scheduling subproblems into a unique MILP model to solve the whole problem in one step. Production requirements with multiple due dates were considered. On the other hand, a number of batch scheduling models reported in the literature can deal with multiple orders but assuming each one associated to a different product. For instance, Castro et al.[20] developed two alternative MILP formulations, either using multiple time grids or global precedence sequencing variables, for the optimal scheduling of single-stage facilities. Their main contribution was the definition of explicit aggregated tasks by merging those performed on consecutive batches of the same product at the same equipment unit. In this way, the model size shows a substantial reduction. However, the two formulations not only assume known batch sizes but also a common due date for all products demands at the end of the time horizon. Therefore, intermediate due dates are not considered. More recently, new MILP mathematical formulations for the simultaneous batch selection, assignment, and sequencing in multistage multiproduct processes were proposed by Prasad and Maravelias[21] and Sundaramoorthy and Maravelias.[22] Both models exploit the sequential structure of multistage processes and are based on a continuous-time representation that includes precedence-based sequencing variables to easily account for sequence-dependent changeovers. Moreover, batch existence and sizing are also model variables. These formulations can consider multiple customer orders with different due dates, but each requirement involves a different product. A simple, practical rule for determining the minimum/maximum number of batches allocated to each production order was developed. To avoid redundant solutions, symmetry breaking constraints based on batch sizes were included. Besides, these models incorporate valid knapsack inequalities to exclude subsets of infeasible assignments and use time window information to fix some sequencing variables and identify forbidden manufacturing paths. Recent extensions of the described approach that also include additional constraints to handle intermediate storage[23] and utilities[24] have been published.

When batching and scheduling decisions are performed in one step, problems featuring multiple customer orders with different due dates for the same product have received limited attention in the literature.[18,19] Sequential approaches proposed by Prasad and Maravelias[21] and Sundaramoorthy and Maravelias[22] just consider multiple due dates for customer demands each one requiring a different product. On the other hand, most monolithic approaches based on a STN/RTN representation do not account for this problem feature either. The work of Ierapetritou et al.[10] can be regarded as an exception. It is a continuous-time unit-specific event-based model, tailored to accommodate intermediate due dates where multiple demands for each product have to be satisfied. Although due dates are allowed to be strictly or partially satisfied, the least overall tardiness has not been considered as the problem goal. Instead, this monolithic formulation seeks to minimize the total operation cost.

In this paper, a pair of new monolithic methods has been developed to address the combined lot-sizing and scheduling of single-stage multiproduct batch facilities. Not just one but multiple orders for each product with different due dates are allowed, and every batch can be used to satisfy more than one customer order. Both methods are based on mathematical formulations developed by using a continuous-time representation and the global precedence concept introduced by Méndez et al.[16] Through a novel procedure, a predefined set of batches for each product−due date pairing containing enough elements to guarantee optimality is generated. Batches selected by the model are just those allocated to some equipment item and, consequently, batch-existence 0−1 variables are omitted. Moreover, variable processing times can be handled. Compared with previous works, redundant solutions are avoided in a more efficient way by developing symmetry breaking constraints based on allocation variables. The two proposed formulations handle batch sequencing decisions in a different manner. The first one rigorously arranges individual batches assigned to the same unit, while the other sequences properly defined clusters of batches sharing the same product and due date and processed in the same equipment item. Contents of clusters in the final schedule are model decisions. In both models either the total tardiness or the makespan was chosen as the problem objective. The nonrigorous cluster-based approach has been validated by comparing its results with the ones found through the rigorous mathematical formulation.

The paper is organized as follows. Section 2 presents a formal statement of the problem to be solved, section 3 points out the major model assumptions, and section 4 introduces a rigorous monolithic batching and scheduling formulation for single-stage batch facilities. A careful explanation about how to define the set of batches for each product and due date pairing has been included. Section 5 presents a cluster-based formulation where aggregated tasks are defined in order to reduce the model size and complexity for tackling large batch scheduling problems. Computational results and comparisons with previous contributions have been included in section 6, while section 7 presents the final conclusions.

2. Problem Statement

The combined lot-sizing and scheduling problem for single-stage multiproduct batch facilities can be stated as follows. Given

(i) a single stage multiproduct batch plant with nonidentical parallel units $j \in J$,
(ii) a set of products $i \in I$ to be manufactured,
(iii) a set of production orders for every product and their related release times and due dates,
(iv) the amount r_{id} of product $i \in I$ to deliver before the due date $d \in D_i$, i.e. the size of order (i,d),
(v) the set of available units $J_i \subseteq J$ for processing product $i \in I$,
(vi) minimum ($q_{ij}^{\min}$) and maximum ($q_{ij}^{\max}$) sizes for a batch of product i at unit j,
(vii) the processing time for product i in unit $j \in J_i$, given by the summation of a fixed term (ft_{ij}) plus a variable term that depends on the batch size through the constant rate υt_{ij} denoting the variable processing time per unit batch size,
(viii) sequence-dependent setup times $\tau_{ii'j}$ between consecutive batches of different products processed in the same unit,
(ix) the length of the time horizon H.

The problem goal is to determine (a) the number and size of batches to be processed, (b) the allocation of equipment items to batches, (c) the batch processing queue at every equipment unit, and (d) the initial and completion times for each batch, such that all production orders are timely satisfied, plant operation constraints are fulfilled, and the selected performance criterion is optimized. Alternative problem objectives considered in this paper include the overall weighted tardiness and the makespan.

3. Model Assumptions

To derive the proposed monolithic formulation for the scheduling of single-stage batch facilities, the following assumptions have been made:

(1) Model parameters are all deterministic.
(2) Equipment units operate in nonpreemptive mode.
(3) Multiple customer orders can involve the same product and different due dates.
(4) Several customer orders of a given product can be totally or partially satisfied by the same batch.
(5) Several batches can be produced to meet a customer order.
(6) Processing times depend on both the product and the batch-size.
(7) Changeover times are sequence-dependent. In addition, changeover times between batches of the same product are neglected.
(8) Resources aside from processing units (storage vessels, utilities, manpower, raw materials) do not constitute bottleneck resources and, therefore, are ignored in the problem formulation.

Let us define the set $D = \cup_{i\in I} D_i$ comprising all order delivery due dates to be considered for the development of a feasible schedule. In the proposed mathematical formulation, the due dates $d \in D$ will be used not only to refer to an order due date (expressed in hours, for example) but also as a subscript to denote that a problem parameter or variable is associated to due date d. When replenishment orders for inventory rather than customer orders are considered, intermediate due dates are not specified and the requested product deliveries should be completed before the horizon end. In this case, the makespan is usually adopted as the objective function to be minimized.

4. Rigorous Batch-Sizing and Scheduling Problem Formulation

4.1. Estimation of the Number of Batches to Be Processed. In order to implement the proposed integrated approach, a systematic procedure is first presented to get a good, conservative estimation of the number of batches (nb_i) to be processed in order to satisfy the total demand of product i. An estimation of nb_i can be obtained based on the overall ith-product requirement and the capacity of the available equipment units. The proposed nb_i is expected to slightly "overestimate" the one really needed in the optimal production schedule so that the model size does not unnecessarily increase, and it can be solved to optimality in a reasonable CPU time. Both an approximate and a rigorous method are presented for computing nb_i. Since it overestimates the required number of batches of product i, only some of them will be finally processed to meet the total demand of product i.

Each batch element b of the set B_i comprising nb_i lots of product i will have its specific due date $d \in D_i$. If batch b is completed after its due date d, a nonzero tardiness on the fulfillment of demand r_{id} will arise. Besides being allocated to the demand r_{id}, batch b can also be used to satisfy additional requirements of product i due at times later than d. Therefore, a batch b can also be assigned to meet demands $r_{id'}$ with due dates $d' > d$, but it must be completed before the earliest one d to avoid tardiness. To compute the value of nb_i, the batch set B_i is divided into as many different subsets B_{id} as the number of customer orders for product i, i.e. a different one for each $d \in D_i$. Any element of B_{id}, if selected for processing, will be used to fulfill the demand r_{id} and, eventually, other requirements due at $d' > d$. Thus, $B_i = \cup_{d\in D_i} B_{id}$.

4.1.1. Approximate Estimation Procedure. An approximate estimation of nb_i can be found based on the fact that a batch $b \in B_i$ assigned to a given due date $d \in D_i$ can also fulfill later ith demands if a product surplus remains after satisfying the amount r_{id}. Let us define the parameter bs_i as the reference batch size for product i, whose value is given by eq 1:

$$bs_i = \min_{j\in J_i}\{q_{ij}^{\min}\} \qquad \forall i \in I \tag{1}$$

Then, bs_i is equal to the lowest possible batch size for product i. Given the overall demand of product i at all time points $d \in D_i$, a conservative estimation of the integer nb_i is given by

$$nb_i = \left\lceil \frac{\sum_{d\in D_i} r_{id}}{bs_i} \right\rceil \qquad \forall i \in I \tag{2}$$

However, the choice of a larger bs_i will allow to reduce nb_i and, consequently, the model size. For instance, bs_i can be defined either as the arithmetic mean of the minimum batch sizes at the equipment units available for product i, as given by eq 3a, or the arithmetic mean of the average batch sizes over $j \in J_i$ provided by eq 3b.

$$bs_i = \frac{1}{|J_i|}\sum_{j\in J_i} q_{ij}^{\min} \qquad \forall i \in I \tag{3a}$$

$$bs_i = \frac{1}{|J_i|}\sum_{j\in J_i}\left(\frac{q_{ij}^{\max} + q_{ij}^{\min}}{2}\right) \qquad \forall i \in I \tag{3b}$$

Once an expression for bs_i is adopted, the maximum number of batches for product i (nb_i) and the set $B_i = \{b_k\}$ with $k = 1, ..., nb_i$ are both established through eq 2. As already mentioned, each batch $b \in B_i$ is primarily allocated to a production order and features its due date $d \in D_i$. This date d represents the latest completion time for batch b to avoid tardiness. For example, if there are three orders for product i with different due dates, i.e. $D_i = \{d_1, d_2, d_3\}$ such that $d_1 < d_2 < d_3$, some batches of B_i will be assigned to the order of product i due at d_1. However, it is possible that some amount of product i from one of those batches, for instance $b^{\#}$, will remain in inventory after fulfilling the order (i, d_1). Then, it becomes available to meet later requirements at times d_2 and d_3. Nevertheless, batch $b^{\#}$ has been assigned to d_1 and must be completed before d_1 because otherwise it will incur a positive tardiness. Moreover, a good estimation of the number of batches allocated to an order (i, d), nb_{id}, should consider the fact that another lot associated to an earlier due date $d' < d$ could partially satisfy the requirement r_{id}.

Based on the production requirement r_{id} and the reference batch size bs_i, eq 4 provides an estimation of the integer number of batches nb_{id} needed to satisfy the overall requirement of product i up to due date d.

$$nb_{id} = \left\lceil \frac{\sum_{\substack{d' \in D_i \\ d' \le d}} r_{id'}}{bs_i} \right\rceil \qquad \forall i \in I, d \in D_i \tag{4}$$

Notice that $nb_{id} = nb_i$ when d is the last due date for product i. Through eq 4, the set of batches B_i has been divided into several subsets B_{id}, one for each due date $d \in D_i$. The number of batches allocated to due date d, i.e. $|B_{id}|$, can be derived from eq 5:

$$|B_{id}| = \begin{cases} nb_{id}, & \text{if } d \text{ is the first due date of product } i \\ nb_{id} - nb_{i(d-1)}, & \text{otherwise} \end{cases} \tag{5}$$

In eq 5, the index $(d - 1)$ stands for the due date directly preceding d in the set D_i. If $D_i = \{d_1, d_2, d_3\}$, then the first $nb_{i,d1}$ elements of B_i will have a due date d_1, the next $(nb_{i,d2} - nb_{i,d1})$ batches should be completed before d_2, and the last ones $(nb_{i,d3} - nb_{i,d2})$ will feature a due date d_3. Then, $B_i = B_{i,d1} \cup B_{i,d2} \cup B_{i,d3}$. If $nb_{id} = nb_{i(d-1)}$, then $B_{id} = \emptyset$. The tentative set of batches $B_i = \cup_{d \in D_i} B_{id}$ generated by eqns 1, 4, and 5 is initially proposed to formulate the monolithic scheduling models.

It is worth noting that the number and size of batches finally included in the optimal schedule are model decisions. Consequently, the main purpose of eqns 1, 4, and 5 is to just postulate a sufficient number of batches for each product requirement. They usually provide a conservative estimation of the number of batches nb_i actually needed. Sometimes, however, the distribution of nb_i among the subsets $\{B_{id}\}$ may be inadequate to guarantee the discovery of the optimal solution, even if bs_i is given by eq 1. An unbalanced batch distribution may cause a lot of product i from the set B_{id} to be assigned to later demands due at $d' > d$. Otherwise, such later requirements could not be satisfied. If completed after time d, such a batch will erroneously generate a positive tardiness. Due to that fictitious tardiness, the best schedule can become nonoptimal and the model will fail to find the right solution. A simple example illustrating this possible failure of eqns 1, 4, and 5 is included in Appendix A.

Because it is more convenient to produce batches of larger size and, eventually, keep some product surplus in inventory to meet later demands, the average batch size finally selected by the model is usually larger than the reference size bs_i provided by eq 1. In many problems, therefore, the optimal solution requires fewer batches than the ones proposed by eqns 1, 4, and 5. Consequently, the approximate procedure usually provides good conservative estimations of both nb_i and nb_{id}. However, it should be remarked that the solution found by the model is proven optimal only if a large enough number of batches nb_{id}, greater than the one required at the optimum, $(nb_{id})^*$, is defined. Then, some special precautions are to be taken when the approximate procedure is applied. Let us assume that the problem formulation has been solved and some batch of the set B_{id}, $i \in I$, $d \in D_i$, has been partially/entirely allocated to order (i, d') with $d' > d$ at the optimum to meet the product requirement $r_{id'}$. To ensure optimality, it becomes necessary to increase $|B_{id'}|$ by one and solve the problem formulation again. The procedure should be repeated until either the unbalanced batch distribution no longer arises or the optimal objective value remains unchanged.

4.1.2. Rigorous Procedure. Though more conservative, a right estimation of the number of batches allocated to the requirement r_{id} that never fails is given by eq 6.

$$|B_{id}| = \left\lceil \frac{r_{id}}{\min_{j \in J_i}\{q_{ij}^{\min}\}} \right\rceil \qquad \forall i \in I, d \in D_i \tag{6}$$

Equation 6 ensures that enough batches of product i will always be available to meet the requirement due at every $d \in D_i$. It is proven in Appendix B that the set $B_i = \cup_{d \in D_i} B_{id}$ provided by eqns 1, 4, and 5 will have at most $|D_i| - 1$ fewer elements than the one determined by eq 6. A more detailed comparison of the sets B_{id} generated with both procedures can be found in section 6, where computational results are discussed. If either the exact procedure given by eq 6 or the approximate strategy given by eqns 1, 4, and 5 is applied, the proposed formulation systematically behaves better than event-based scheduling methods using an arbitrary number of events/time points.

4.2. Problem Constraints. Let us assume that a tentative set of batches $B_i = \cup_{d \in D_i} B_{id}$, using either the exact or the approximate procedure, has been defined for each product i. However, the number of individual batches actually processed will be determined by solving the proposed mathematical formulation. Some of the problem constraints given below are used to simultaneously choose and allocate batches to equipment items. In contrast to the approach of Prasad and Maravelias,[21] binary variables denoting the existence of batches are not required. In the problem representation, particular consideration to symmetry-breaking constraints avoiding redundant solutions has been given.

4.2.1. Allocation Constraints. Constraint 7 states that batch b of product i can at most be allocated to a single unit j. If the LHS of eq 7 is zero, then batch (b, i) is never processed. Therefore, it becomes a fictitious batch and any related variable will be ignored. Otherwise, batch b of product i does exist and it is processed in unit j.

$$\sum_{j \in J_i} Y_{bij} \le 1 \qquad \forall i \in I, b \in B_i \tag{7}$$

Figure 1 shows an example of a batch (b, i) assigned to unit j, i.e. $Y_{bij} = 1$. Variables related to batch (b, i) are the starting time ST_{bi}, the completion time CT_{bi}, and the processing time on the allotted unit PT_{bij}. If batch (b, i) is not assigned to any unit, then it is not scheduled and the associated variables become meaningless.

To illustrate the allocation of several batches of the same product $i \in I$ with different due dates $d \in D_i$, a more complex example is depicted in Figure 2. Eight batches of two different products are to be scheduled. The batch facility includes three parallel units $\{U_1, U_2, U_3\}$ to process the set of products $I = \{i_1, i_2\}$ with delivery due dates $D = \{24, 48\}$. For each product−due date pairing, the following sets of batches have been defined:

$$B_{i1,d1} = \{b_1, b_2, b_3, b_4\}; \quad B_{i1,d2} = \{b_5, b_6\}$$
$$B_{i2,d1} = \emptyset; \quad B_{i2,d2} = \{b_1, b_2, b_3\}$$

Figure 2 shows that batches b_1, b_2, and b_3 of product i_1 have been chosen to meet the requirement $r_{i1,d1}$. Instead, the last element b_4 of the set $B_{i1,d1}$ is a fictitious batch, i.e. the left-hand side (LHS) of eq 7 for b_4 is equal to zero. On the other hand, all the elements of $B_{i1,d2}$ and $B_{i2,d2}$ were assigned to units by the model.

4.2.2. Symmetry-Breaking Constraints. Before solving the proposed mathematical model, each batch $b \in B_{id}$ can be regarded as a generic batch. They are so because their sizes

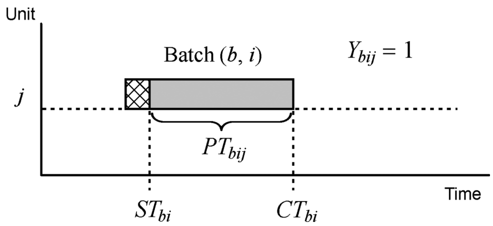

Figure 1. Variables related to batch (b, i) allocated to unit j.

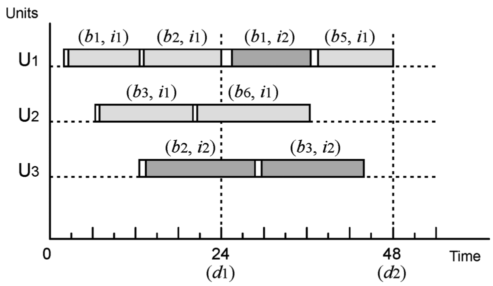

Figure 2. Selection and allocation of batches to meet requirements of products (i_1, i_2) at two different due dates (d_1 and d_2).

and processing times are established after assigning them to equipment units. If some fictitious batches are included in the set B_{id}, one of several equivalent subsets of B_{id} can be chosen by the model. Therefore, whenever $|B_{id}|$ is larger than the number of batches $(nb_{id})^*$ actually needed, the model solution space may include multiple equivalent solutions. Indeed, there are two possible sources of problem degeneracy that must be eliminated. They are the batch selection and the unit allocation processes. In fact,

(i) One of several equivalent subsets of B_{id} can be chosen if the set B_{id} includes fictitious batches.

(ii) One of several equivalent combinations of equipment units can be allocated to each selected subset of batches.

If not avoided, sources (i) and (ii) could generate a huge number of equivalent, feasible schedules as the number of batches and units both increase. In order to prevent the possible source (i), the rule given by eq 8 can be used. This rule specifies that the model can choose to process batch $b \in B_{id}$ only if the preceding batch $(b - 1)$ has also been selected. Then, constraint 8 will allow batch b to also be processed only if $Y_{(b-1)ij} = 1$ for some $j \in J_i$. In case batch b is the $(k + 1)$th element of B_{id}, then it can exist only if the previous k batches in B_{id} have been assigned to processing units.

$$\sum_{j \in J_i} Y_{bij} \leq \sum_{j' \in J_i} Y_{(b-1)ij'} \quad \forall i \in I, d \in D_i, b \in B_{id}: (b - 1) \in B_{id} \tag{8}$$

To illustrate the possible source (ii), let us suppose that two batches b_1 and b_2 are selected from B_{id} and allocated to units U_1 and U_2, respectively. Because b_1 and b_2 are generic batches, the alternative allocation decisions, i.e. $\{b_1$ to U_1; b_2 to $U_2\}$ or $\{b_1$ to U_2; b_2 to $U_1\}$, are equivalent since batch sizes and processing times of batches b_1 and b_2 can be selected after allocating units to them. In other words, the generic names "b_1" and "b_2" are irrelevant and the actual decision to be taken by the model is the allocation of two batches of product i: one to unit U_1 and the other to unit U_2. When several batches in B_{id} and multiple units in J_i are considered, the degeneracy source (ii) generates a large number of equivalent allocation decisions for each possible subset of B_{id}. To overcome degeneracy source (ii), a more general rule for the allocation of units $j \in J_i$ to batches $b \in B_{id}$ is proposed. This rule, given by eq 9, establishes that a batch (b, i) with due date d can be selected and allocated to unit $j \in J_i$ ($Y_{bij} = 1$) only if the preceding element $b - 1$ in the set B_{id} is processed in some unit $j' \leq j$. Therefore, eq 9 prioritizes the allocation of batch b_1 ($<b_2$) to the earliest unit U_1 ($<U_2$) in the set J_i, i.e. $\{b_1$ to U_1; b_2 to $U_2\}$. If b_2 is also assigned to U_1, it should be performed after the completion of b_1.

$$Y_{bij} \leq \sum_{\substack{j' \in J_i \\ j' \leq j}} Y_{(b-1)ij'} \quad \forall i \in I, d \in D_i, b \in B_{id}, j \in J_i: (b - 1) \in B_{id} \tag{9}$$

Because the elements of B_{id} are generic batches, constraint 9 just forces the model to choose one of several equivalent solutions. It is like finding first the optimal schedule and then assigning names to the existing batches in such a way that eq 9 holds. Therefore, no solution is eliminated from the feasible space and constraint 9 is a valid cut. Since eq 9 accounts for both degeneracy sources (i) and (ii), eq 8 is no longer needed in the problem formulation. Constraints 9 are quite different and computationally more efficient than the symmetry-breaking equations based on batch sizes proposed by Prasad and Maravelias[21] and Sundaramoorthy and Maravelias.[22] To illustrate the power of the valid cut (9), a simple example is considered. Let us assume that requirements of product i all feature the same due date, and the sets B_i and J_i are given by $\{b_1, b_2, b_3, b_4\}$ and $\{U_1, U_2, U_3\}$, respectively. In case b_1 has been assigned to unit U_3, constraint 8 just indicates that b_2 can exist and be allocated to some unit $j \in J_i$. Instead, constraint 9 drives the allocation variables $\{Y_{b2,U1}, Y_{b2,U2}, Y_{b3,U1}, Y_{b3,U2}, Y_{b4,U1}, Y_{b4,U2}\}$ to zero. In this way, the solution space is significantly reduced. Obviously, the effect of cut 9 on the model size is stronger as the number of elements of sets $\{B_{id}\}$ increases.

Figure 2 shows a batch schedule that complies with the rule stated by eq 9. As defined before, $B_{i1,d1} = \{b_1, b_2, b_3, b_4\}$ is the tentative set of batches of product i_1 due at time d_1. Batches b_1 and b_2 have been allocated to unit U_1 while batch b_3 was assigned to the next unit U_2. Such assignment decisions satisfy eq 9. In contrast, batch b_4, the last element of $B_{i1,d1}$, is never processed because the production of $\{b_1, b_2, b_3\}$ is enough to meet the requirement $r_{i1,d1}$. Besides, $B_{i1,d2} = \{b_5, b_6\}$ comprises the batches of product i_1 associated to due date d_2. In accordance to eq 9, they have been allocated to units U_1 and U_2, respectively. In turn, product i_2 has only one due date at d_2 with three proposed batches $\{b_1, b_2, b_3\}$ that were all scheduled. Again, eq 9 holds because the processing of b_1 was assigned to unit U_1, while b_2 and b_3 in this order were scheduled at unit U_3.

4.2.3. Batch Size Constraints. The size of batch (b, i) depends on the assigned equipment item $j \in J_i$, as indicated by eq 10. The parameter $q_{ij}^{\min}$ denotes the minimum batch size for product i in unit j.

$$BS_{bi} = \sum_{j \in J_i} (q_{ij}^{\min} Y_{bij} + Q_{bij}) \qquad \forall i \in I, b \in B_i \tag{10}$$

The summation on the right-hand side (RHS) of eq 10 includes two terms. The constant part of BS_{bi}, given by the first term $q_{ij}^{\min} Y_{bij}$, provides the portion of the batch size that is common to any lot of product i assigned to unit j. Besides, the non-negative variable Q_{bij} stands for the variable portion of the batch size BS_{bi} that is specific for the lot (b, i) and is restrained to the range $[0, \Delta_{ij}]$ through constraint 11. Notice that $Q_{bij} = 0$ if $\Delta_{ij} = 0$.

$$Q_{bij} \le \Delta_{ij} Y_{bij} \qquad \forall i \in I, b \in B_i, j \in J_i \tag{11}$$

where: $\Delta_{ij} = (q_{ij}^{\max} - q_{ij}^{\min})$.

4.2.4. Product Demand Constraint. The amount of product $i \in I$ required at time d can be supplied by any batch (b, i) featuring a delivery due date $d' \le d$. In order to meet every ith-product demand, the amount of product i contained in batches with due dates not later than d should never be lower than the accumulated demand of product i up to time $d \in D_i$.

$$\sum_{\substack{d' \in D_i \\ d' \le d}} \sum_{b \in B_{id'}} BS_{bi} = \sum_{\substack{d' \in D_i \\ d' \le d}} \sum_{b \in B_{id'}} \sum_{j \in J_i} (q_{ij}^{\min} Y_{bij} + Q_{bij}) \ge \sum_{\substack{d' \in D_i \\ d' \le d}} r_{id'} \qquad \forall i \in I, d \in D_i \tag{12}$$

4.2.5. Batch Processing Times. The processing time of batch (b, i) in unit j, referred as PT_{bij}, is given by constraint 13. The fixed processing time of batch (b, i) in eq 13 is represented by the term $ft_{ij}Y_{bij}$, while the variable part is given by $vt_{ij}(q_{ij}^{\min} Y_{bij} + Q_{bij})$.

$$PT_{bij} = ft_{ij} Y_{bij} + vt_{ij}(q_{ij}^{\min} Y_{bij} + Q_{bij}) \qquad \forall i \in I, b \in B_i, j \in J_i \tag{13}$$

If the processing time is not a function of the batch size, then the variable component of PT_{bij} is omitted ($vt_{ij} = 0$). When $\Delta_{ij} = 0$, the variable processing time reduces to $vt_{ij}\, q_{ij}^{\min}$. Moreover, the condition $Y_{bij} = 0$ drives PT_{bij} to zero because of constraint 11. Besides, the relationship between the starting and completion times of batch (b, i) is defined by eq 14.

$$CT_{bi} = ST_{bi} + \sum_{j \in J_i} PT_{bij}; \quad ST_{bi} \ge rt_{id} \qquad \forall i \in I, d \in D_i, b \in B_{id} \tag{14}$$

4.2.6. Sequencing Constraints. Task sequencing constraints are required for each pair of batches of the same or different products that are processed in the same unit.

4.2.6.1. Batches of the Same Product. By definition, the elements of the set $B_i = \cup_{d \in D_i} B_{id}$ have been ordered by increasing due dates. If two batches $b, b' \in B_i$ of product i are processed in unit j ($Y_{bij} = Y_{b'ij} = 1$) and $b < b'$, in order to reduce the overall tardiness of product i, batch b should be scheduled earlier. Therefore, the sequencing constraint 15 establishes that the completion time of batch (b, i) must never exceed the starting time of (b', i) if both batches are processed in the same unit and $b < b'$. The changeover time between consecutive batches of the same product is neglected.

$$CT_{bi} \le ST_{b'i} + H(2 - Y_{bij} - Y_{b'ij}) \qquad \forall i \in I, b, b' \in B_i, j \in J_i: (b < b') \tag{15}$$

If both batches $b, b' \in B_{id}$ have the same due date d and $b < b'$, then lot b should still be processed before b', based on the same arguments used to derive eq 9.

4.2.6.2. Batches of Different Products. If two batches $b \in B_{id}$ and $b' \in B_{i'd'}$ of different products ($i \ne i'$) are allocated to the same queue, sequencing constraints are needed to decide which batch will be processed first. Using the global precedence concept of Méndez et al.,[16] such sequencing constraints 16a and 16b have been written in terms of the binary variables $X_{bi,b'i'}$. In those constraints, a single variable $X_{bi,b'i'}$ with $i < i'$ is just needed to sequence the pair of lots (b, i) and (b', i'). If $X_{bi,b'i'} = 1$, batch (b, i) precedes batch (b', i') in the queue of the assigned unit. Otherwise, $X_{bi,b'i'} = 0$ and batch (b', i') is processed before. If batches (b, b') are not allocated to the same unit, the value of $X_{bi,b'i'}$ becomes meaningless. The global precedence concept has the advantage of reducing the number of sequencing variables and, at the same time, the handling of sequence dependent changeovers becomes a straightforward task.

$$CT_{bi} + \tau_{ii'j} \le ST_{b'i'} + H(1 - X_{bi,b'i'}) + H(2 - Y_{bij} - Y_{b'i'j}) \qquad \forall i, i' \in I, d \in D_i, d' \in D_{i'}, b \in B_{id}, b' \in B_{i'd'}, j \in J_{ii'}: (i < i') \tag{16a}$$

$$CT_{b'i'} + \tau_{i'ij} \le ST_{bi} + HX_{bi,b'i'} + H(2 - Y_{bij} - Y_{b'i'j}) \qquad \forall i, i' \in I, d \in D_i, d' \in D_{i'}, b \in B_{id}, b' \in B_{i'd'}, j \in J_{ii'}: (i < i') \tag{16b}$$

When batches (b, i) and (b', i') feature different due dates d and d', and the difference $|d' - d|$ is large enough, it can be assumed that batch (b, i) will precede (b', i') at the best solution if $d < d'$. Let us define the parameter δ as the minimum absolute difference between due dates d and d' required to preorder the associated batches. When $|d' - d| \le \delta$, eqs 16a and 16b are used. Otherwise, eqs 16a and 16b can be replaced by constraint 16c. If the parameter δ is not large enough, constraint 16c may lead to suboptimal solutions.

$$CT_{bi} + \tau_{ii'j} \le ST_{b'i'} + H(2 - Y_{bij} - Y_{b'i'j}) \qquad \forall i, i' \in I, d \in D_i, d' \in D_{i'}, b \in B_{id}, b' \in B_{i'd'}, j \in J_{ii'}: (i \ne i') \wedge (d' - d > \delta) \tag{16c}$$

4.2.7. Makespan Definition.

$$CT_{bi} \le MK \qquad \forall i \in I, b \in B_i \tag{17}$$

4.2.8. Tardiness Definition. Unlike sequential scheduling models, the proposed formulation associates a non-negative variable T_{id} to each production requirement r_{id}. The continuous variable T_{id} represents the tardiness on satisfying the demand of product i required at due date d. As indicated by eq 18, the batch $b \in B_{id}$ that is completed last is the one that determines the tardiness T_{id}. Moreover, the overall tardiness of product i must account for every due date $d \in D_i$.

$$CT_{bi} - d \le T_{id} \qquad \forall i \in I, d \in D_i, b \in B_{id} \tag{18}$$

If tardiness is not allowed, then every order should be delivered on time and the RHS of eq 18 is driven to zero. Then,

$$CT_{bi} \le d \qquad \forall i \in I, d \in D_i, b \in B_{id} \tag{19}$$

4.3. Tightening Constraints. This section presents additional tightening constraints to be included in the mathematical model in order to accelerate the convergence rate of the MILP solver. These constraints are valid cuts that strongly reduce the integrality gap as the solution algorithm progresses. The optimality of the schedule found is not compromised since no integer solution is excluded from the feasible space.

4.3.1. Tightened Lower Bound for the Makespan. When the makespan is the problem goal to be minimized, it can be expected that the best schedule will feature very low equipment idle times. Therefore, the overall workload of each unit j can be very well approximated by just adding the processing times of the assigned batches. The largest unit-workload provides a valid lower bound for the makespan.

$$\sum_{i \in I_j} \sum_{b \in B_i} PT_{bij} \le MK \qquad \forall j \in J \tag{20}$$

The effectiveness of cut 20 becomes deteriorated if changeover times are non-negligible and sequence-dependent. However, using a similar strategy to the one presented in Marchetti and Cerdá,[25] additional work can be done to further tighten the lower bound of the makespan. Equation 20 can be replaced by constraints 21a including a conservative estimation of the necessary changeover for the first batch of each product allotted to unit j. In turn, eq 21b defines the continuous variable $\nu_{ij} \in [0, 1]$ so as to detect if at least one batch of product i is processed in such unit.

$$\sum_{i \in I_j} \sum_{b \in B_i} PT_{bij} + \sum_{i \in I_j} \sigma_{ij}^{\min} \nu_{ij} - \max_{i \in I_j}(\sigma_{ij}^{\min}) \leq MK \qquad \forall j \in J \tag{21a}$$

$$Y_{ibj} \leq \nu_{ij} \qquad \forall i \in I, b \in B_i, j \in J_i$$

$$\text{where } \sigma_{ij}^{\min} = \min_{\substack{i' \in I_j \\ i' \neq i}} \{\tau_{i'ij}\} \tag{21b}$$

4.3.2. Valid Cuts for the Overall Tardiness. An additional valid cut (eq 22) is proposed to generate a tighten lower bound for the total tardiness. It is based on the information provided by the processing times PT_{bij}. The LHS of constraint 22 determines the overall processing time of all lots of products $i \in I_j$ with $r_{id} > 0$ that are processed in unit j and allocated to orders (i, d') with due dates $d' \leq d$. On the other hand, the RHS of eq 22 is an estimation of the time at which the processing of such batches is completed. To this end, it adds the summation over all the related tardiness T_{id} to the time point d. It is worth noting that eq 22 is a valid cut for two reasons. First, the largest $(T_{id})^{\#}$ should be added to the time point d to determine a bound on the completion time of the batches considered in the LHS of eq 22. Instead, it is replaced by the summation of all related tardiness, including $(T_{id})^{\#}$. Besides, the value of T_{id} can be determined by a batch $b \in B_{id}$ processed in another unit $j' \neq j$. Constraint 22 is defined only for units $j \in J$ that can process batches with due date d.

$$\sum_{\substack{i \in I_j \\ r_{id} > 0}} \sum_{\substack{d' \in D \\ d' \leq d}} \sum_{b \in B_{id'}} PT_{bij} \leq d + \sum_{\substack{i \in I_j \\ r_{id} > 0}} T_{id} \qquad \forall d \in D, j \in J_d \tag{22}$$

where: $J_d = \{j \in J | \exists\, i \in I: j \in J_i \wedge r_{id} > 0\}$.

4.4. Objective Function. The proposed unified model is well-suited for the combined batching and scheduling of single-stage batch plants with multiple due dates per product. In general, the main goal of the scheduling task is to complete all production requirements in a timely manner. Thus, the usual problem target given by eq 23 is to reach the maximum customer satisfaction by minimizing the overall weighted tardiness. The weighting coefficient ε_{id} is a measure of the importance of timely satisfying the customer order r_{id}.

$$\text{minimize} \sum_{i \in I} \sum_{d \in D_i} \varepsilon_{id} T_{id} \tag{23}$$

Alternatively, eq 24 uses the makespan as the objective function to be minimized. In this case, the definition of MK given by constraint 17 should be included on the mathematical model.

$$\text{minimize } MK \tag{24}$$

In this way, a rigorous MILP formulation for the unified batch-sizing and scheduling of single-stage batch facilities has been developed. It comprises the set of eqs 7 and 9−16(a−c). In addition, it includes constraints 17 and 21 if the problem objective is given by eq 24 or eqs 18 and 23 when the problem goal is the minimum overall tardiness.

5. Cluster-Based Batch-Sizing and Scheduling Problem Formulation

Real-life scheduling problems usually require scheduling dozens or hundreds of batches in order to meet multiple customer orders at different due dates over a weekly or monthly horizon. To solve such industrial-size problems in less CPU time, this section introduces a group-based formulation relying on the idea of replacing individual batches by properly defined clusters of batches containing the same product and due at the same promised date. This approximate group-based approach reduces the size and complexity of the mathematical formulation by substantially decreasing the number of sequencing variables.

If several batches of product i with due date d are all allocated to the same unit j, it is very likely that they are consecutively processed. Therefore, they can be treated and assigned to units as a cluster. This is a usual practice in industry where batches of the same product destined to the same customer order and allocated to the same equipment item are processed one after the other. Its primary goal is to decrease the number of product changeovers. Since transitions between compatible products are commonly favored by the scheduler, there is often a direct correlation between the number of product changeovers and the related total setup time. To embed this practical assumption in the problem formulation, batches of the same product with an equal due date and allocated to the same unit are grouped together and handled as a single entity with regard to sequencing decisions. Variables associated to such groups of batches are identified with a superscript G and three subscripts (i, d, j) standing for the product that it contains, the common delivery date, and the assigned unit. For example, the variable Y_{idj}^G stands for the existence of a cluster g_{idj} containing batches of product i due at time d and processed in unit $j \in J_i$. Let us suppose that $B_{id} = \{b_1, b_2, b_3\}$ and $J_i = \{j_1, j_2\}$. Then, two clusters of batches $\{g_{id,j1}, g_{id,j2}\}$ comprising the set $B_{id} = \{b_1, b_2, b_3\}$ and preassigned to units j_1 and j_2, respectively, can be defined to meet the requirement r_{id}. However, the sequence of batches of product i finally processed in each unit will be a model decision. Thus, it may occur that unit j_1 just processes the sequence $\{b_1, b_2\}$ while batch b_3 is produced in unit j_2. Sometimes, one of the predefined clusters is never performed. In other words, the contents of clusters $\{g_{id,j1}, g_{id,j2}\}$ in the optimal schedule are determined by solving the proposed cluster-based formulation. Similar groups of batches for product i can be defined for each due date $d \in D_i$ with $r_{id} > 0$, whenever $B_{id} \neq \emptyset$. Therefore, the number of clusters for product i to be considered in the problem formulation is not larger than $(|D_i| \times |J_i|)$. To allow that the final contents of clusters become model decisions, allocation (Y_{bid}) and sizing (Q_{bid}) variables related to individual batches $b \in B_{id}$ together with eqs 7 and 9−13 are still needed in the new problem formulation. As presented in the rigorous formulation, allocation decisions will be restricted by the rule imposed by eq 9. Although it is an approximate method, the group-based approach is capable of finding good quality solutions and even becomes a rigorous one when some conditions on product changeover times hold (see Appendix C).

Figure 3 shows the continuous variables representing the existence (Y_{idj}^G), the starting time (ST_{idj}^G), and the completion time (CT_{idj}^G) for cluster g_{idj}. The condition $Y_{idj}^G = 0$ denotes that the group g_{idj} is not selected by the model, and therefore, no batch

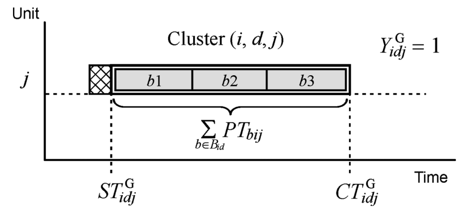

Figure 3. Variables related to cluster g_{idj}.

$b \in B_{id}$ is processed in unit $j \in J_i$. If so, all the variables related to g_{idj} become meaningless. Besides, the proposed cluster-based approach just includes sequencing variables ($X_{id,i'd',j}^G$) to order pairs of clusters g_{idj} and $g_{i'd'j}$ processed in the same unit $j \in J_{ii'}$ rather than individual batches.

5.1. Cluster Existence Constraint. A group of batches g_{idj} does exist only if one or several batches of B_{id} have been allocated to unit $j \in J_i$. To indicate the existence of g_{idj}, a new continuous variable $Y_{idj}^G \in [0, 1]$ is introduced. When at least one batch of product i due at time d is assigned to unit j, constraint 25 makes $Y_{idj}^G = 1$.

$$Y_{bij} \le Y_{idj}^G \qquad \forall i \in I, d \in D_i, b \in B_{id}, j \in J_i \tag{25}$$

5.2. Cluster Processing Times. The processing time of a group g_{idj} is obtained by adding the processing times of all batches $b \in B_{id}$ that were assigned to unit $j \in J_i$. Then,

$$CT_{idj}^G = ST_{idj}^G + \sum_{b \in B_{id}} PT_{bij} \qquad \forall i \in I, d \in D_i, j \in J_i \tag{26}$$

where $ST_{idj}^G \ge rt_{id}$.

5.3. Cluster Sequencing Constraints. Sequencing constraints are required between different groups of batches allocated to the same unit. These constraints will be similar to the sequencing constraints defined by eqs 15 and 16(a−c).

5.3.1. Sequencing Batch Clusters Involving the Same Product. If two groups containing batches of the same product i with different due dates d and d' are allocated to the same unit j ($Y_{idj}^G = Y_{id'j}^G = 1$), then the group with the lowest due date must be queued before in order to reduce the overall tardiness. Sequencing constraint 27 indicates that the starting time of $g_{id'j}$ must never be lower than the completion time of cluster g_{idj} if both groups are associated to the same unit and $d < d'$.

$$CT_{idj}^G \le ST_{id'j}^G + H(2 - Y_{idj}^G - Y_{id'j}^G) \quad \forall i \in I, d, d' \in D_i, j \in J_i:(d < d') \tag{27}$$

Thus, batches belonging to the group g_{idj} will be packed together and performed before the batches pertaining to $g_{id'j}$.

5.3.2. Sequencing Clusters of Different Products. Similar to the sequencing of individual batches, constraints 28a and 28b are incorporated in the problem formulation to order pairs of clusters of different products, g_{idj} and $g_{i'd'j}$, processed in the same unit $j \in J_{ii'}$. A group g_{idj} precedes or succeeds $g_{i'd'j}$ on the queue of unit j when the new 0−1 variable $X_{id,i'd',j}^G$ equals 1 or 0, respectively. A single variable $X_{id,i'd',j}^G$ (with $i < i'$) is needed to ordering the pair of clusters g_{idj} and $g_{i'd'j}$. By using this sequencing scheme, binary variables $X_{bi,b'i'}$ are replaced by 0−1 variables $X_{id,i'd',j}^G$ to get a problem formulation with a lower number of sequencing variables and constraints. If one or both groups are not selected by the model, constraints 28a and 28b are never active and the value of $X_{id,i'd',j}^G$ becomes meaningless. Besides, eqs 28a and 28b can account for sequence-dependent changeovers between clusters of different products.

$$CT_{idj}^G + \tau_{ii'j} \le ST_{i'd'j}^G + H(1 - X_{id,i'd',j}^G) + H(2 - Y_{idj}^G - Y_{i'd'j}^G) \quad \forall i, i' \in I, d \in D_i, d' \in D_{i'}, j \in J_{ii'}:(i < i') \tag{28a}$$

$$CT_{i'd'j}^G + \tau_{i'ij} \le ST_{idj}^G + HX_{id,i'd',j}^G + H(2 - Y_{idj}^G - Y_{i'd'j}^G) \quad \forall i, i' \in I, d \in D_i, d' \in D_{i'}, j \in J_{ii'}:(i < i') \tag{28b}$$

When the difference $|d' - d|$ is large enough because $d \ll d'$, it can be assumed that the group g_{idj} will be processed before $g_{i'd'j}$, if some batches of products i and i' with due dates d and d', respectively, have been assigned to unit j. Otherwise, a large tardiness on satisfying the demand r_{id} may arise. Similar to eq 16c, the positive parameter δ is used to determine if groups g_{idj} and $g_{i'd'j}$ can be preordered. If $(d' - d) > \delta$, then the cluster g_{idj} is processed before and constraint 28c can be applied. Otherwise, $|d' - d| \le \delta$ and constraints 28a and 28b should be used to sequence clusters g_{idj} and $g_{i'd'j}$.

$$CT_{idj}^G + \tau_{ii'j} \le ST_{i'd'j}^G + H(2 - Y_{idj}^G - Y_{i'd'j}^G) \quad \forall i, i' \in I, d \in D_i, d' \in D_{i'}, j \in J_{ii'}:(i \ne i') \text{ and } (d' - d > \delta) \tag{28c}$$

5.4. Makespan Definition.

$$CT_{idj}^G \le MK \qquad \forall i \in I, d \in D_i, j \in J_i \tag{29}$$

5.5. Tardiness Definition. As indicated by constraint 30, the difference between the completion time CT_{idj}^G at unit $j \in J_i$ and the time point d is a lower bound for T_{id}.

$$CT_{idj}^G - d \le T_{id} \qquad \forall i \in I, d \in D_i, j \in J_i \tag{30}$$

If no tardiness is allowed, constraint 30 should be replaced by eq 31.

$$CT_{idj}^G \le d \qquad \forall i \in I, d \in D_i, j \in J_i \tag{31}$$

Either the minimum makespan or the least overall tardiness can be chosen as the problem objective. Then, the cluster-based formulation includes the set of constraints composed by eqs 7, 9−13, and 25−28. Depending on the selected objective function, constraints 29 or 30 should also be considered.

6. Computational Results

In this section, three case studies with a size and complexity gradually increasing from example 1 to example 3 have been solved. They allow testing the computational efficiency and the quality of the solution found through the proposed MILP monolithic formulations. The three examples consider: (a) sequence-dependent changeover times, (b) fixed or variable batch sizes, (c) fixed/variable processing times, and (d) multiple orders per product with different due dates. To find the tentative number of batches for each product and its distribution among the related due dates, the approximate and the exact procedures presented in section 4.1 have been applied. The rigorous unified model presented in section 4 and the cluster-based formulation developed in section 5 were both implemented with GAMS modeling system, and all examples were solved to optimality using CPLEX 11.0 mixed-integer optimizer on a 1.8 GHz 1 Gb Pentium IV PC. Besides, different values of the parameter δ were used to indicate the lowest due date difference $|d - d'|$ above which a pair of batches or clusters of batches featuring due dates (d, d') with $|d - d'| > \delta$ can be preordered using the earliest due date (EDD) rule. All the examples were solved using

Table 1. Product Requirements (kg) and Related Due Dates (h) for Example 1

	due date (h)					
product	24	48	72	96	120	144
P_1			18000			
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	

Table 2. Product Batch Sizes (kg) and Processing Times (h) for Example 1

	batch size							
unit	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
processing time	8	10	12	12	8	16	12	20

Table 3. Sequence-Dependent Changeovers (h) for Example 1

$\tau_{ii'}$	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

both the proposed rigorous and cluster-based formulations. The larger example 3 is presented to show the computational advantages of the cluster-based approach to cope with real-life scheduling problems involving dozens of tasks.

6.1. Example 1. Example 1 has been introduced by Méndez et al.[17] and was later studied by Lim and Karimi.[18] It involves a single-stage multiproduct batch facility with seven parallel units producing eight different products. Production demands are given in terms of 29 customer orders as shown in Table 1. Most of the products have several requirements at different due dates. Table 2 presents the fixed batch size for each product at every available unit. From Tables 1 and 2, it follows that batch sizes and production requirements do not perfectly match up. Few of the allowed batch sizes can exactly fulfill a given product requirement. Besides, fixed processing times for each product that do not depend on the assigned unit, and sequence-dependent changeover times between different products are shown in Tables 2 and 3, respectively. Moreover, the length of the time horizon is 7 days (168 h), and the overall tardiness is the objective function to be minimized. Since the work of Méndez et al.[17] uses a different problem goal, comparison of results can only be made with the model of Lim and Karimi.[18]

In this example, the tentative set of batches for each product requirement r_{id} was determined using the approximate procedure. Table 4 reports the sets B_{id} generated through eqs 4 and 5, with the reference batch size (bs_i) given by eq 1. To illustrate the use of the approximate procedure, results found for product P_4 are analyzed. Product P_4 has five production orders with the first due at $d_2 = 48$ h and a reference (minimum) batch size

Table 4. Tentative Set of Batches for Example 1 Using the Approximate Procedure

	due date (h)					
product	24	48	72	96	120	144
P_1			b_1, b_2, b_3			
P_2		b_1	b_2			
P_3		b_1, b_2				
P_4		b_1	b_2, b_3		b_4	b_5
P_5	b_1		b_2			
P_6		b_1		b_2		b_3
P_7	b_1	b_2		b_3	b_4	
P_8	b_1				b_2	

Table 5. Computational Results for Example 1

	Lim and Karimi (2003)	proposed model (section 4)	
		$\delta = 0$	$\delta = 120^a$
binary variables	209	180	368
continuous variables	265	76	76
constraints	1583	1811	2841
MILP solution	0	0	0
CPU time (s)	3.8^b	**0.15**	**0.46**
nodes	20	0	10
Iterations	1281	134	275

[a] Without preordering. [b] Best CPU time reported by Lim and Karimi[18] using GAMS/CPLEX 6.6 on an HP7194-116.

$bs_{P4} = 5000$ kg. Consequently, a single batch is enough to handle the requirement of 4950 kg at d_2, i.e. $nb_{P4,d2} = 1$, $B_{P4,d2} = \{b_1\}$. Therefore, 50 kg remain in inventory to satisfy the next requirement at time $d_3 = 72$ h. Because 5053 kg of product P_4 are to be delivered at d_3, two additional batches are needed. Then, $nb_{P4,d3} = 3$ and $B_{P4,d3} = \{b_2, b_3\}$. However, the optimal solution must not necessarily include both batches because a batch b_2 containing 6000 kg of P_4 can be produced in units different from U_5-U_6. Since enough inventory (4997 kg) will be available after satisfying the demand of P_4 at $d_3 = 72$ h, no further batches were required for $B_{P4,d4}$. Besides, demands due at times d_5 and d_6 will be covered with the sets $B_{P4,d5} = \{b_4\}$ and $B_{P4,d6} = \{b_5\}$, respectively. Thus, a total of $nb_{P4} = 5$ batches will be considered to meet the production requirements of P_4. Overall, the total number of batches initially proposed amounts to 23. In general, a better utilization of the plant capacity is obtained when units are employed at full capacity and a single batch can be used to fulfill several production orders. If several batches are needed for each customer order, the cluster-based formulation proposed in section 5 becomes a more convenient approach.

Once a tentative set of batches B_i is determined for each product i, the proposed unified batching and scheduling formulation can be solved. Valid cuts (9) were included to avoid symmetric solutions. Model statistics and computational results for two alternative values of δ (0 h, 120 h) using the rigorous mathematical formulation are presented in Table 5. For comparison, the best results reported by Lim and Karimi[18] are also included in that table. If the preordering parameter $\delta = 0$ is adopted, batches of different products are ordered by increasing due dates when allocated to the same unit, while no batch preordering is made if $\delta = 120$ h because $|d - d'| \leq 120$ for every $d, d' \in D$. For both values of δ, the same optimal solution with zero tardiness has been found. In sequencing constraints 15−16(a−c), the horizon length $H = 7$ days (168 h) was adopted as the big-M parameter, and tightening constraint 22 has been included to effectively bound the total tardiness selected as the problem objective.

The optimal schedule is depicted in Figure 4. It includes 21 batches, two less than the number of batches estimated by the

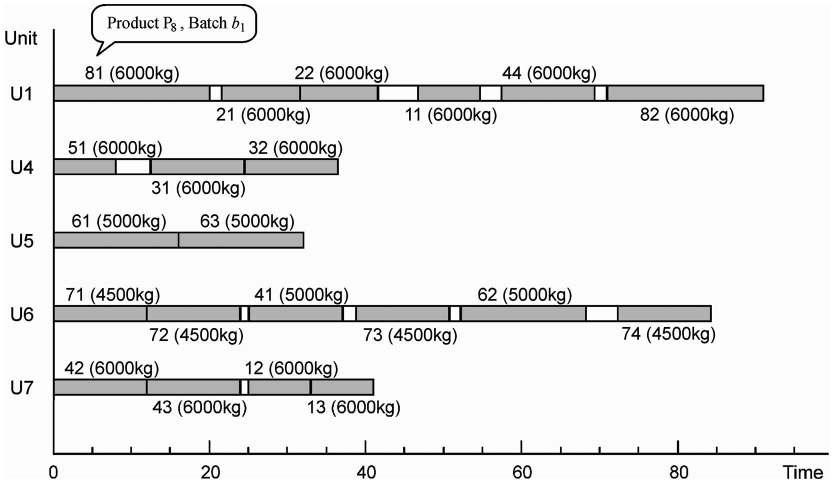

Figure 4. Optimal solution found for example 1.

approximate procedure. The two fictitious batches correspond to products P_4 and P_5. Figure 4 clearly supports the idea of grouping batches featuring the same product, due date, and assigned unit. In fact, it is observed that: (i) batches $\{b_1, b_2\}$ of product P_3 with $d = 48$ h are consecutively processed in unit U_4; (ii) batches $\{b_2, b_3\}$ of P_4 with $d = 72$ h are produced one after another in U_7; and (iii) batches $\{b_2, b_3\}$ of P_1 with $d = 72$ h are successively processed in unit U_7. These represent the only three cases where lots featuring the same product and due date have been assigned to the same equipment item. Obviously, the best schedule is also found by solving the cluster-based formulation.

Though Table 5 shows an important reduction in the number of binary variables for $\delta = 0$, the problem is indeed easily solved in less than a second and few iterations of the solver algorithm even if $\delta = 120$ and no batch preordering is made. Despite the fact that the model of Lim and Karimi[18] presents a lower number of 0−1 variables, it requires a CPU time almost 8 times larger than the proposed approach with $\delta = 120$. In addition, the number of explored nodes and iterations is also higher. However, it is worth noting that Lim and Karimi's results were obtained with GAMS/CPLEX 6.6 on an HP7194-116 machine, probably with lower performance than the machine/solver configuration used in this paper. Here, it should be remarked that Lim and Karimi[18] used a strategy where time slots at each unit are assigned beforehand to due dates, in such a way that the workload is properly balanced among the units. As a result, few slots are generally proposed and the model size is thus reduced. In contrast, the proposed approach estimates a conservative number of batches for each product to guarantee optimality and still shows a better computational performance.

On the other hand, the use of symmetric-breaking constraints 8 instead of eq 9 produces a rise in the CPU time for $\delta = 0$ (from 0.15 to 0.36 s) and for $\delta = 120$ (from 0.46 to 0.87 s). Since only three sets $\{B_{id}\}$ comprise multiple batches in example 1, the extra effect of cut 9 on the computational effort is rather limited. Moreover, a total of 36 batches are suggested by the exact procedure described in section 4.1.2, i.e. 13 batches more than the previous estimation. However, the solution time just grows from 0.15 to 0.29 s for $\delta = 0$, and from 0.46 to 0.84 s for $\delta = 120$.

6.2. Example 2. The next example was proposed by Lim and Karimi[18] and involves a multiproduct batch plant with three parallel units producing four different products. Multiple customer orders required at four different due dates are to be satisfied (see Table 6). Sequence-dependent changeover times

Table 6. Product Demands and Sequence-Dependent Changeovers for Example 2

	product demand (kg) at due date (h)				sequence-dependent changeover time (h)			
	24	48	72	96	P_1	P_2	P_3	P_4
P_1	50	100	100	100	0.0	5.1	1.6	1.0
P_2	100	100	100	200	1.5	0.0	2.3	2.5
P_3	50	100	100		1.6	1.3	0.0	2.1
P_4		200	100	100	2.7	4.8	1.4	0.0

Table 7. Batch Sizes and Processing Time Coefficients for Example 2

	lower/upper batch size (kg) fixed (h)/variable (h/kg) processing time coefficient			
unit	P_1	P_2	P_3	P_4
U_1		100/140 3/0.15	100/150 2/0.17	150/200 2.5/0.15
U_2	100/120 5/0.15		100/120 2/0.17	100/150 3/0.155
U_3	140/160 5/0.145	80/120 4/0.155		

Table 8. Tentative Set of Batches for Example 2 Using the Approximate Procedure

	due date (h)			
product	24	48	72	96
P_1	b_1	b_2	b_3	b_4
P_2	b_1, b_2	b_3	b_4	b_5, b_6, b_7
P_3	b_1	b_2	b_3	
P_4		b_1, b_2	b_3	b_4

between batches of different products are also listed in Table 6. Besides, Table 7 presents the feasible range of batch sizes, and fixed and variable processing time coefficients for each product. The scheduling objective is to minimize the overall tardiness, and the length of the time horizon is $H = 120$ h.

Again, the approximate strategy is applied to determine the tentative sets of batches shown in Table 8. Product P_3 requires three batches, and products P_1 and P_4 each require four batches, while seven batches are proposed for product P_2. The bigger cardinality of B_{P2} comes from the minimum batch size for P_2 equal to 80 kg at unit U_3, and its overall requirement of 500 kg. Thus, $nb_{P2} = \lceil 500/80 \rceil = 7$. However, not all the batches proposed have been selected at the best schedule.

Computational results obtained by the rigorous batching and scheduling formulation with $\delta = 0$ and $\delta = 72$ h, and the best results found by Lim and Karimi[18] are all presented in Table 9. Both approaches were able to find the optimal solution with an

Table 9. Computational Results and Model Statistics for Example 2

	Lim and Karimi (2003)	proposed model (section 4)	
		$\delta = 0$	$\delta = 72^a$
binary variables	77	63	153
continuous variables	94	87	87
constraints	483	379	478
MILP solution	30.51	30.51	30.51
CPU time (s)	59.0[b]	**5.23**	**9.51**
nodes	20014	3581	7139
iterations	412012	49889	80675

[a] Without preordering. [b] Best CPU time reported by Lim and Karimi[18] using GAMS/CPLEX 6.6 on an HP7194-116.

Table 10. Allocation of Batches to Production Orders at the Optimum of Example 2

product	due date	req	selected batches					tardiness	remaining inventory
			batch	unit	size	start	end		
P_1	24	50.00	b_1	U_2	100.00	0.00	20.00	0.00	50.00
	48	100.00	b_2	U_3	150.00	21.00	47.75	0.00	100.00
	72	100.00						0.00	0.00
	96	100.00	b_4	U_3	140.00	72.61	97.91	1.91	40.00
P_2	24	100.00	b_1	U_3	100.00	0.00	19.50	0.00	0.00
	48	100.00	b_3	U_1	108.00	28.80	48.00	0.00	8.00
	72	100.00	b_4	U_3	92.00	52.85	71.11	0.00	0.00
	96	200.00	b_5	U_1	100.00	70.60	88.60		
			b_6	U_1	100.00	88.60	106.60	10.60	0.00
P_3	24	50.00	b_1	U_1	150.00	0.00	27.50	3.50	100.00
	48	100.00						0.00	0.00
	72	100.00	b_3	U_1	100.00	50.30	69.30	0.00	0.00
P_4	48	200.00	b_1	U_2	100.00	21.00	39.50		
			b_2	U_2	100.00	39.50	58.00	10.00	0.00
	72	100.00	b_3	U_2	100.00	58.00	76.50	4.50	0.00
	96	100.00	b_4	U_2	100.00	76.50	95.00	0.00	0.00

Table 11. Product Requirements (kg) for the Two Instances of Example 3

	case A: tardiness				case B: makespan			
	due date				due date			
product	48	96	168	216	48	96	120	orders for inventory
P_1	400	600				300		1000
P_2		400	250	800	350			900
P_3	350	150				150	200	500
P_4			350	600		250		1000
P_5	500	200		750	200			850
P_6		500	300			100		1200

overall tardiness of 30.51 h but requiring different CPU times. Results presented in Table 9 were obtained by using eq 22 to estimate a valid lower bound for tardiness variables. The optimal schedule found is presented in Figure 5. Whether the parameter δ is set to 0 or 72 h, example 2 was solved using the rigorous formulation in less than 10 s, while almost 60 s were required using the approach of Lim and Karimi.[18] Although CPU times are not strictly comparable because different computers and solvers were used, it is observed an important reduction on the number of explored nodes and iterations. Besides, it is worth noting that the result included in Table 9 is precisely the best one obtained by Lim and Karimi,[18] while the worst one reported in their work needs up to 791.8 s. Despite demanding more binary variables when a larger δ is used, our rigorous approach shows a small deterioration of the CPU times. Besides, the optimal schedule shown in Figure 5 also supports the soundness of the batch cluster idea. There are two instances where the cluster notion can be applied: (i) batches $\{b_5, b_6\}$ of P_2 with d = 96 h processed in unit U_1; (ii) batches $\{b_1, b_2\}$ of P_4 with d = 48 h assigned to unit U_2. In both cases, such batches are processed one after another. As a result, the cluster-based approach also provides the optimal schedule. Finally, Table 10 includes, for each product, detailed information about the batches selected to fulfill every customer order. Batches b_3 of product P_1, b_2 and b_7 of product P_2, and b_2 of product P_3 have not been selected by the model. Thus, only 14 from a total of 18 batches were finally used. Results in Figure 5 and Table 10 are in agreement with those reported by Lim and Karimi[18] for this example.

In example 2, the replacement of eq 9 by constraint 8 produces an increase of the solution time for $\delta = 0$ (from 5.23 to 5.67 s) and for $\delta = 72$ (from 9.51 to 12.39 s). Again, only three sets $\{B_{id}\}$ comprise multiple batches, and consequently, the additional effect of cut 9 on the CPU time is rather minor. Furthermore, the total number of batches suggested by the exact procedure rises to 20, i.e. two batches more than the previous estimation. Using this number of batches, the solution time grows from 5.23 to 5.72 s for $\delta = 0$ and from 9.51 to 15.08 s for $\delta = 72$ h.

6.3. Example 3. In order to cope with industrial-sized problems featuring a high number of batches for each product, a third case study is presented. Example 3 refers to a single-stage batch plant with four parallel units producing six different products. Two instances of example 3 with different production requirements are presented in Table 11. On one hand, example 3a involves 14 customer orders to be satisfied at four different due dates in such a way that the total tardiness is minimized. On the other hand, example 3b comprises 7 customer orders due at time points 48, 96, and 120 h to be satisfied on time (without tardiness) and some inventory replenishment orders, a single one for each product, to be fulfilled by the end of the time horizon. Problem data for instances 3a and 3b are presented in Tables 12 and 13. Minimum/maximum batch sizes for each unit, together with fixed (ft_{ij}) and variable (vt_{ij}) processing times coefficients are listed in Table 12. Besides, Table 13 provides the sequence-dependent changeover times between pairs of

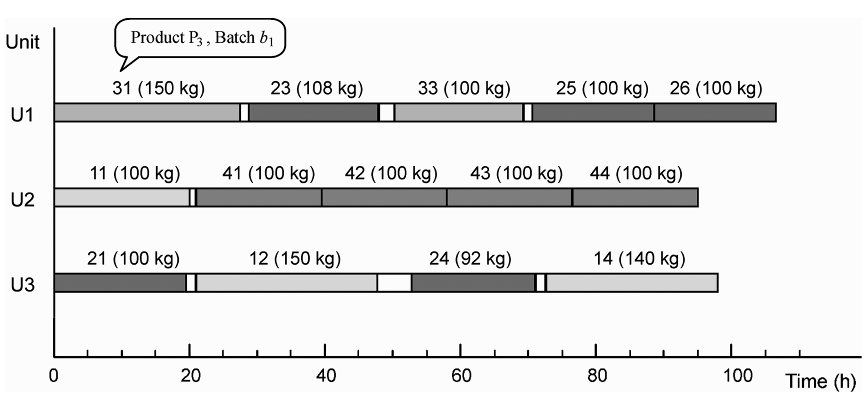

Figure 5. Gantt chart of the optimal solution for example 2.

Table 12. Batch Size Limits and Processing Time Coefficients for Example 3

unit	lower/upper batch size (kg) fixed (h)/variable (h/kg) processing time coefficient					
	P_1	P_2	P_3	P_4	P_5	P_6
U_1		200/270 6/0.08	200/250 4/0.1	190/225 5/0.12	250/320 6/0.06	
U_2	100/125 4/0.1		100/120 6/0.12		200/250 7/0.12	160/240 4/0.18
U_3		225/325 4/0.1		175/200 5/0.1		100/275 5/0.12
U_4	150/200 6/0.08	180/210 5/0.12	150/175 4/0.1		225/300 8/0.08	140/175 8/0.15

Table 13. Sequence-Dependent Changeovers (h) for Example 3

$\tau_{ii'}$	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0.0	3.1	5.1	1.6	3.4	3.3
P_2	6.1	0.0	1.3	3.4	2.5	1.7
P_3	2.7	3.0	0.0	4.7	1.4	1.4
P_4	3.3	3.3	1.6	0.0	1.2	4.4
P_5	4.3	6.9	3.1	1.8	0.0	1.7
P_6	1.8	1.7	5.5	2.0	3.1	0.0

Table 14. Tentative Set of Batches for Example 3a Using the Exact Procedure

product	due dates (h) 48	96	168	216
P_1	b_1-b_4	b_5-b_{10}		
P_2		b_1-b_3	b_4, b_5	b_6-b_{10}
P_3	b_1-b_4	b_5, b_6		
P_4			b_1, b_2	b_3-b_6
P_5	b_1-b_3	b_4		b_5-b_8
P_6		b_1-b_5	b_6-b_8	

Table 15. Computational Results for Example 3a Using Both the Rigorous and Cluster-Based Formulations

	Example 3a. Tardiness Minimization					
	cluster-based formulation (section 5)			rigorous formulation (section 4)		
	$\delta = 0$	$\delta = 72$	$\delta = 168^a$	$\delta = 0$	$\delta = 72$	$\delta = 168^a$
binary variables	165	222	268	347	658	1020
continuous variables	257	257	257	239	239	239
constraints	705	762	808	2902	3474	4090
MILP solution	14.90	14.90	14.90	14.90	14.90	14.90
CPU time (s)	**27.39**	**32.65**	**39.39**	**124.73**	**784.12**	**942.18**
nodes	8488	11579	13016	6179	37197	17622
iterations	264245	255758	327008	208861	2012490	1685888

[a] Without preordering.

different products. For both instances of example 3, the time horizon is $H = 240$ h (10 days).

Example 3a. Table 14 shows the tentative set of batches for example 3a obtained with the exact procedure presented in section 4.1.2. Overall, 48 batches are included. Since more batches than in previous examples are needed, and multiple batches of the same product are associated to the same due date, it is convenient to use the cluster-based MILP formulation proposed in section 5. In order to compare its performance with that of the rigorous formulation used as a reference, example 3a has been solved using both methodologies. Tightening constraint 22 for batch tardiness will be used in both models.

Table 15 presents the computational results obtained for three different values of the parameter δ. For every value of δ, both the rigorous and the cluster-based formulations find the same optimal schedule. However, the cluster-based formulation is able to reduce the required CPU time by various orders of magnitude (see Table 15). The CPU time drops approximately 4 times for $\delta = 0$ and 24 times for both $\delta = 72$ and $\delta = 168$. This computational time saving can be explained by the important reduction in the number of sequencing variables and constraints. Therefore, the criterion of grouping batches related to the same product, due date, and unit seems to be, in practice, a very useful idea. The CPU time required by the cluster-based approach grows from 27.4 to 39.4 s as the value of δ increases.

The Gantt chart of the best solution for example 3a obtained with the group-based formulation using $\delta = 0$ is depicted in Figure 6. Only 34 batches from a total of 48 proposed by the exact procedure have been processed. Besides, Table 16 shows the detailed timing and sizing decisions for the batches produced in each unit. By analyzing the resulting batch sizes, it follows that 23 batches use the minimum batch size, 6 batches have the maximum size, and 5 of them feature an intermediate size between the minimum and maximum. Table 17 shows how the 34 batches included in the optimal solution are allocated to product requirements in example 3a. Only three orders are tardily satisfied. The use of the weaker cuts 8 instead of constraint 9 produces a sharp increase in the CPU time by at least 2 orders of magnitude. This is because 13 sets $\{B_{id}\}$ include multiple batches to meet the corresponding requirement r_{id}. For $\delta = 0$ h/72 h/168 h, using the cluster-based approach, the solution time rises from (27.39 s/32.65 s/39.39 s) to (1380.1 s/851.96 s/945.2 s). Finally, it is worth mentioning that the same solution value but lower CPU times were obtained by adopting the number of batches provided by the approximate procedure. In this case, the number of tentative batches drops to 46 and, consequently, the solution time decreases (i) from 27.39 to 10.39 s for $\delta = 0$, (ii) from 32.65 to 15.66 s for $\delta = 72$ h, and (iii) from 39.39 to 25.42 s for $\delta = 168$ h.

Example 3b. In Example 3b, customer and inventory replenishment orders are to be satisfied (see Table 11). Because the length of the time horizon H is set to 240 h and inventory replenishment orders should be completed before the horizon end, the associated due date for such orders is $d = 240$ h. In turn, customer orders should be fulfilled without tardiness. Using either the approximate or the exact procedure, it has been found the same set of batches for each product, which is shown in Table 18. A total of 13 batches are proposed to timely meet customer orders featuring three different due dates at times 48, 96, and 120 h. Besides, 43 additional batches are initially associated to inventory requirements with a common due date $d = 240$ h. Thus, the overall number of batches to be handled by the model increases to 56. Since most of the production requirements are destined for inventory and the customer orders should be timely satisfied, then the minimum makespan has been selected as the problem target.

Similarly to example 3a, the new instance of example 3 has also been solved using both the cluster-based continuous-time formulation and the rigorous model at the batch level. Since customer orders must be delivered on time, eqs 19 or 31 should be considered. Besides, tightening constraints 21(a−b) are used to reduce the integrality gap.

Computational results for two alternative values of the preordering parameter δ are reported in Table 19. The best solution for example 3b has an optimal makespan of 223.21 h. Although it finds a feasible schedule with the same objective value, the rigorous model presented in section 4 was unable to prove optimality within 1 h of CPU time. In contrast, the group-based formulation (section 5) efficiently solves the problem to optimality in less than 288 s of CPU time for both values of δ. The Gantt chart of the best solution found with the cluster-

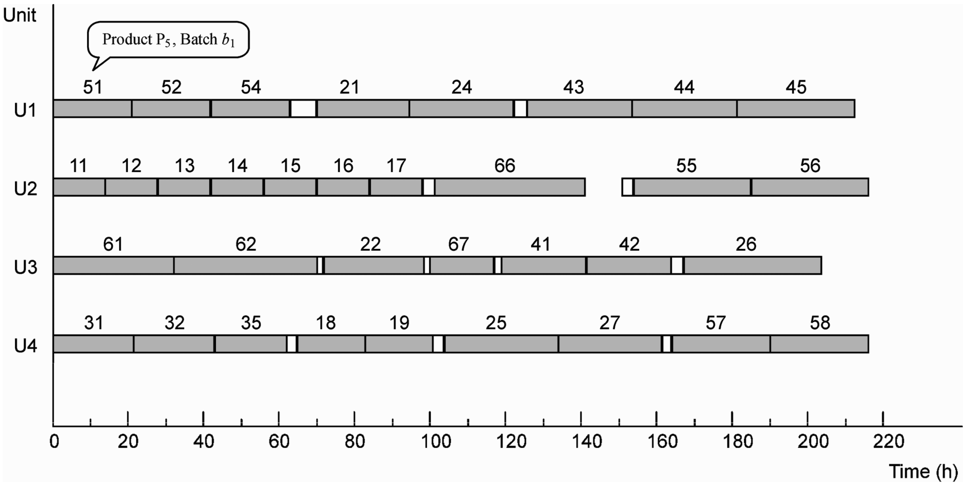

Figure 6. Optimal schedule found for example 3a (tardiness minimization).

Table 16. Detailed Schedule Found for Example 3a Using the Group-Based Formulation with $\delta = 0$

unit	product	due date	batch	changeover	start	end	size (kg)
U_1	P_5	48	b_1		0.0	21.0	250
			b_2		21.0	42.0	250
	P_5	96	b_4		42.0	63.0	250
	P_2	96	b_1	6.9	69.9	94.5	232.5
	P_2	168	b_4		94.5	122.1	270
	P_4	216	b_3	3.4	125.5	153.3	190
			b_4		153.3	181.1	190
			b_5		181.1	212.5	220
U_2	P_1	48	b_1		0.0	14.0	100
			b_2		14.0	28.0	100
			b_3		28.0	42.0	100
			b_4		42.0	56.0	100
	P_1	96	b_5		56.0	70.0	100
			b_6		70.0	84.0	100
			b_7		84.0	98.0	100
	P_6	168	b_6	3.3	101.3	141.3	200
	P_5	216	b_5	3.1	154.0	185.0	200
			b_6		185.0	216.0	200
U_3	P_6	96	b_1		0.0	32.0	225
			b_2		32.0	70.0	275
	P_2	96	b_2	1.7	71.7	98.2	225
	P_6	168	b_7	1.7	99.9	116.9	100
	P_4	168	b_1	2.0	118.9	141.4	175
			b_2		141.4	163.9	175
	P_2	216	b_6	3.3	167.2	203.7	325
U_4	P_3	48	b_1		0.0	21.5	175
			b_2		21.5	43.0	175
	P_3	96	b_5		43.0	62.0	150
	P_1	96	b_8	2.7	64.7	82.7	150
			b_9		82.7	100.7	150
	P_2	168	b_5	3.1	103.8	134.0	210
	P_2	216	b_7		134.0	161.5	187.5
	P_5	216	b_7	2.5	164.0	190.0	225
			b_8		190.0	216.0	225

based formulation is presented in Figure 7. Besides, detailed information on the optimal schedule is included in Table 20 where the lots processed at each unit, together with their sizes and starting/completion times are all reported. The best schedule for example 3b requires only 35 batches of the total 56 initially proposed. Eleven batches are employed to fulfill customer orders, and the remaining ones are allocated to inventory orders.

Similarly to example 3a, the use of the valid cuts 8 instead of constraints 9 produces a significant increase in the solution time of the cluster-based formulation. After the time limit of 3600 s, the relative gap was still 1.09% for $\delta = 0$ and 0.79% for $\delta = 72$ h when eq 8 is applied. Again, several sets $\{B_{id}\}$ include multiple batches to meet the corresponding requirement r_{id}.

Table 17. Satisfaction of Production Requirements for Example 3a

product	due date	req[a]	batches	batch sizes[a]	completion	tardiness	remaining inventory[a]
P_1	48	400	b_1-b_4	400	56.0	8.0	
	96	600	b_5-b_9	600	100.7	4.7	
P_2	96	400	b_1, b_2	457.5	98.2	2.2	57.5
	168	250	b_4, b_5	480	134.0		287.5
	216	800	b_6, b_7	512.5	203.7		
P_3	48	350	b_1, b_2	350	43.0		
	96	150	b_5	150	62.0		
P_4	168	350	b_1, b_2	350	163.9		
	216	600	b_3-b_5	600	212.5		
P_5	48	500	b_1, b_2	500	42.0		
	96	200	b_4	250	63.0		50.0
	216	750	b_5-b_8	950	216.0		
P_6	96	500	b_1, b_2	500	70.0		
	168	300	b_6, b_7	300	141.3		

[a] Given in kilograms.

Table 18. Tentative Set of Batches for Example 3b Using Either the Approximate or the Exact Procedure

	due dates (h)			
product	48	96	120	H
P_1		b_1-b_3		b_4-b_{13}
P_2	b_1, b_2			b_3-b_7
P_3		b_1, b_2	b_3, b_4	b_5-b_9
P_4		b_1, b_2		b_3-b_8
P_5	b_1			b_2-b_6
P_6		b_1		b_2-b_{13}

7. Conclusions

Two MILP mathematical formulations for the simultaneous lot-sizing and scheduling of single-stage batch facilities have been developed. They both use precedence-based, continuous-time representations that are suited to consider multiple orders of the same product with different due dates. Moreover, batch size-dependent processing times and sequence-dependent changeovers can also be handled. The two proposed MILP models differ in the way that sequencing decisions are taken. The rigorous approach deals with the sequencing of individual batches processed in the same unit, while the approximate cluster-based method arranges groups of batches, each one featuring the same product, due date, and assigned unit. Since cluster members are often consecutively processed, each cluster can be treated and assigned to units as a single entity for sequencing purpose. However, final contents of clusters

Table 19. Computational Results for Example 3b Using Both the Rigorous and Cluster-Based Formulations

	Example 3b. Makespan Minimization			
	group-based formulation (section 5)		rigorous formulation (section 4)	
	$\delta = 0$	$\delta = 72^a$	$\delta = 0$	$\delta = 72^a$
binary variables	182	207	853	895
continuous variables	269	269	276	276
constraints	780	805	4614	4690
MILP solution	223.2123	223.2123	223.2123	223.2123
CPU time (s)	**159.10**	**287.54**	$3600^{b,c}$	$3600^{b,d}$
nodes	62254	93583	172127	163357
iterations	1626044	2742826	5617795	5677665

[a] Without preordering. [b] Time limit exceeded. [c] Best possible solution = 221.1062, relative gap ≈ 0.94%. [d] Best possible solution = 221.1161, relative gap ≈ 0.94%.

Table 20. Optimal Schedule for Example 3b Using the Cluster-Based Formulation with $\delta = 0$

unit	product	due date	batch	changeover	start	end	size (kg)
U_1	P_2	48	b_1		0.0	22.0	200
			b_2		22.0	47.2	240
	P_2	240	b_3		47.2	74.8	270
			b_4		74.8	102.4	270
			b_5		102.4	130.0	270
	P_3	240	b_5	1.3	131.3	155.3	200
	P_5	240	b_2	1.4	156.7	181.2	308.5
			b_3		181.2	202.2	250
			b_4		202.2	223.2	250
U_2	P_1	96	b_1		0.0	15.1	111.3
			b_2		15.1	29.1	100
			b_3		29.1	43.1	100
	P_6	96	b_1	3.3	46.4	81.6	173.2
	P_6	240	b_2		81.6	128.8	240
			b_3		128.8	176.0	240
			b_4		176.0	223.2	240
U_3	P_4	96	b_1		0.0	25.0	200
			b_2		25.0	47.5	175
	P_4	240	b_3		47.5	70.0	175
			b_4		70.0	92.5	175
			b_5		92.5	115.0	175
			b_6		115.0	137.5	175
			b_7		137.5	160.0	175
	P_6	240	b_5	4.4	164.4	185.2	131.8
			b_6		185.2	223.2	275
U_4	P_5	48	b_1		0.0	27.3	241.5
	P_3	96	b_1	3.1	30.4	51.9	175
			b_2		51.9	73.4	175
	P_3	240	b_6		73.4	92.4	150
			b_7		92.4	111.4	150
	P_1	240	b_4	2.7	114.1	136.1	200
			b_5		136.1	158.1	200
			b_6		158.1	179.2	188.7
			b_7		179.2	201.2	200
			b_8		201.2	223.2	200

become model decisions. This group-based approach reduces the size and complexity of the mathematical formulation by substantially decreasing the number of sequencing variables. When the associated due dates are largely different, batch preordering rules can be embedded in the problem formulation. To implement the proposed integrated formulations, a pair of systematic procedures is first presented to get good, conservative estimations of the number of batches to be processed. Since the feasible space usually includes a substantial number of symmetric solutions, strong valid cuts based on assignment variables that never exclude feasible schedules have been developed.

In order to test the proposed approaches in terms of computational efficiency and solution quality, three case studies, including two instances of the largest one, have been tackled. They all consider: (a) sequence-dependent changeover times, (b) fixed or variable batch sizes, (c) fixed/variable processing times, and (d) multiple orders for each product with different due dates. The first two permits to compare the computational performance of the rigorous model with respect to a previous contribution of Lim and Karimi.[18] Though these authors used different hardware/solver platforms with lower performance, there is a trend toward larger computational savings when the proposed approach is applied to the scheduling of congested batch facilities where some customer orders are tardily satisfied. In both examples, the results support the soundness of the batch cluster idea. Every time batches of the same product and due date were assigned to the same unit, they have been consecutively processed. Two instances of the third case study were solved using both the rigorous and the cluster-based formulations to validate the results found using the cluster notion. In one of the instances, a total of 56 batches are to be considered. The cluster-based approach was able to solve both instances of the largest example in much less CPU time. From this computational experience, the cluster-based approach arises as a very promising tool to discover near-optimal schedules in industrial environments.

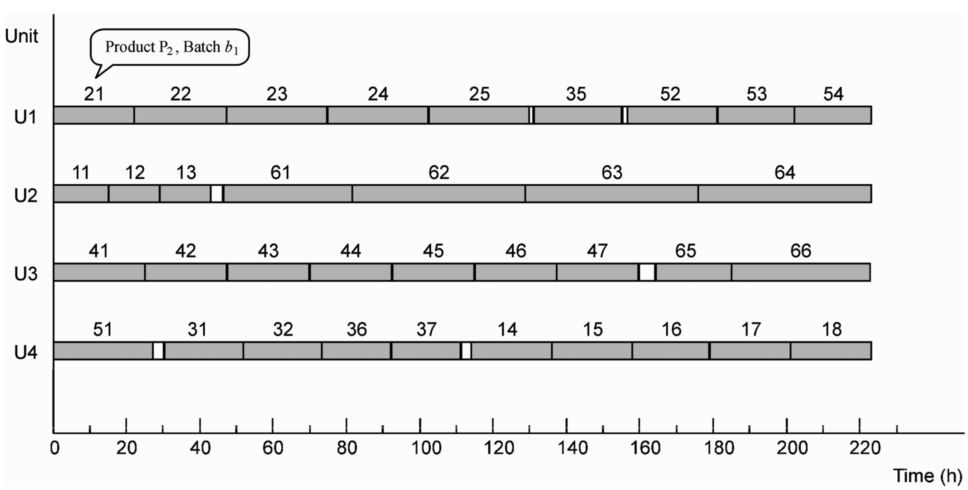

Figure 7. Optimal schedule for example 3b (makespan minimization).

Acknowledgment

The authors would like to acknowledge financial support from FONCyT-ANPCyT under Grant PICT 01837, from CONICET under Grant PIP-2221, and from "Universidad Nacional del Litoral" under CAI+D 66335.

Appendix A. A Simple Case Study Where the Approximate Procedure to Estimate the Number of Batches Fails

A simple case study is presented with one product, a single unit, and a pair of due dates: $I = \{i\}$, $J = \{j\}$, $D = \{24, 48\}$.

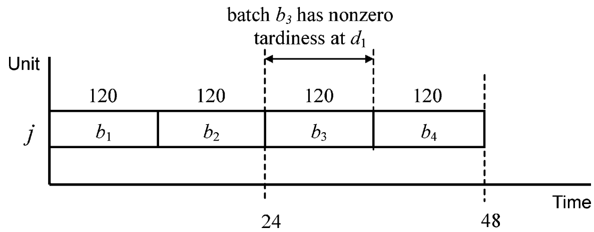

Figure A1. Optimal schedule using the sets of batches provided by the approximate procedure.

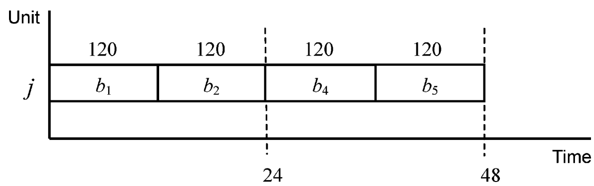

Figure A2. Optimal schedule using the set of batches provided by the exact procedure.

Table A1. Optimal Schedule Found Using the Approximate Procedure

			batches allocated			
due date	requirement	tardiness	batch	start	size	remaining inventory
24	220	12.00	b_1	0	120	
			b_2	12	120	
			b_3	24	120	140
48	180	0.00	b_4	36	120	80

Table A2. Optimal Schedule Results Obtained with the Exact Procedure

			batches allocated			
due date	requirement	tardiness	batch	start	size	remaining inventory
24	220	0.00	b_1	0	120	
			b_2	12	120	20
48	180	0.00	b_4	24	120	
			b_5	36	120	80

The other problem data are the following:

- requirements of product i at each due date: $r_{i,24} = 220$, $r_{i,48} = 180$;
- fixed processing time at unit j: $ft_{ij} = 12$;
- minimum and maximum batch sizes: $q_{ij}^{\min} = 100$; $q_{ij}^{\max} = 120$.

Using the approximate procedure given by eqs 3−5, the tentative sets of batches proposed for each due date are $B_{i,24} = \{b_1, b_2, b_3\}$ and $B_{i,48} = \{b_4\}$. If the overall tardiness is the objective function to be minimized, the optimal schedule found is shown in Figure A1. Even though the Gantt chart seems to be satisfactory, the optimal objective value is 12.0, as shown in Table A1. This situation arises because batch b_3, associated to due date d_1, is fully allocated to the next due date ($d_2 = 48$ h). As a result, the model computes a non-zero tardiness for b_3 with regards to d_1, although such a positive tardiness does not exist because all requirements were fulfilled in a timely manner. On the other hand, the use of the exact procedure leads to find the sets $B_{i,24} = \{b_1, b_2, b_3\}$ and $B_{i,48} = \{b_4, b_5\}$. In this case, the same optimal schedule has been found but the optimal tardiness is equal to zero (see Figure A2 and Table A2).

Appendix B. Relation between the Number of Batches Obtained with Both the Approximate and the Exact Procedures

If $|D_i|$ is the cardinality of the set of delivery dates related to product i, it is easy to prove that $B_i = \cup_{d \in D_i} B_{id}$, as generated by eqs 1, 4, and 5, will have no more than $|D_i| - 1$ batches less than the same set when defined by eq 6. The following alternatives are considered:

(i) If d_1 is the first due date of product i, then both strategies determine the same number of batches for the set $B_{i,d1}$.

$$|B_{i,d1}| = nb_{i,d1} = \left\lceil \frac{r_{i,d1}}{bs_i} \right\rceil$$

(ii) Otherwise, from eq 4 let $nb_{i,d} = \lceil nb_{i,d}^{\#} \rceil$, where:

$$nb_{i,d}^{\#} = \frac{\sum_{\substack{d' \in D_i \\ d' \le d}} r_{id'}}{bs_i} = \frac{\left(\sum_{\substack{d' \in D_i \\ d' \le (d-1)}} r_{id'} \right) + r_{id}}{bs_i} = nb_{i,d-1}^{\#} + \frac{r_{id}}{bs_i} \tag{B.1}$$

Based on this expression, it is possible to develop upper and lower bounds for the parameter nb_{id}, using the following property of the ceiling function:

Given $x, y \in \mathscr{R}$ it is true that: $\lceil x \rceil + \lceil y \rceil - 1 \le \lceil x + y \rceil \le \lceil x \rceil + \lceil y \rceil$

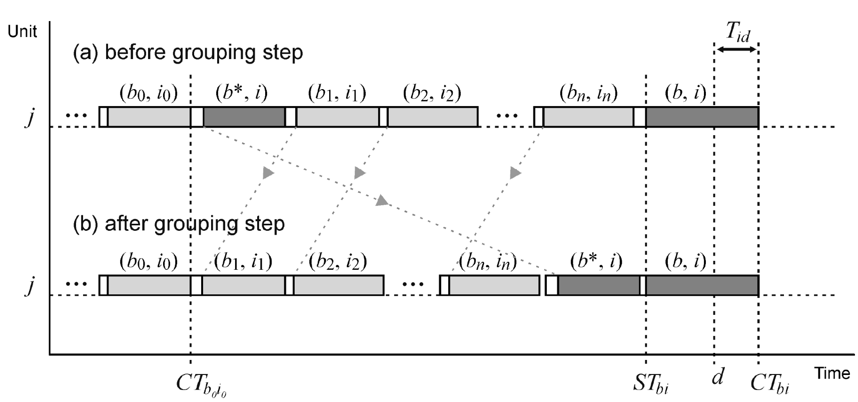

Figure C1. Grouping step used to generate a feasible solution of the cluster-based approach.

Therefore:

$$nb_{i,d} = \left\lceil nb^{\#}_{i,d-1} + \frac{r_{id}}{bs_i} \right\rceil \geq \lceil nb^{\#}_{i,d-1} \rceil + \left\lceil \frac{r_{id}}{bs_i} \right\rceil - 1 = nb_{i,d-1} + \left\lceil \frac{r_{id}}{bs_i} \right\rceil - 1 \quad \text{(B.2)}$$

$$nb_{i,d} = \left\lceil nb^{\#}_{i,d-1} + \frac{r_{id}}{bs_i} \right\rceil \leq \lceil nb^{\#}_{i,d-1} \rceil + \left\lceil \frac{r_{id}}{bs_i} \right\rceil = nb_{i,d-1} + \left\lceil \frac{r_{id}}{bs_i} \right\rceil \quad \text{(B.3)}$$

Using eqs B.2, B.3, and 5, it follows that

$$\left\lceil \frac{r_{id}}{bs_i} \right\rceil - 1 \leq |B_{id}| \leq \left\lceil \frac{r_{id}}{bs_i} \right\rceil \quad \text{(B.4)}$$

Consequently, if bs_i is given by eq 2, for each due date $d \in D_i$ different from the first one, eq 5 suggests at most one batch less than the number of batches determined through eq 6. Since alternative (ii) includes $|D_i| - 1$ due dates, then the number of batches provided by the exact procedure is at most $|D_i| - 1$ higher than the one found with the approximate strategy. Thus, the proof is completed.

Appendix C. Conditions for the Exactness of the Cluster-Based Formulation

Proposition. The best schedule provided by the cluster-based approach is the true optimal solution when the following two conditions hold:

$$(1)\ \tau_{i'ij} + \tau_{ii''j} \geq \tau_{i'i''j} + \tau_{iij} \quad \forall (i, i', i'') \in I, j \in (J_{ii'} \cap J_{ii''}): i \neq i' \wedge i \neq i''$$

$$(2)\ rt_{id} = 0 \qquad \forall i \in I, d \in D_i$$

The proposition is valid if either the makespan or the overall weighted tardiness is to be minimized.

Proof. Under conditions 1 and 2, it will be proved that every feasible solution of the proposed rigorous formulation (section 4) has an associated solution of the cluster-based model (section 5) with an equivalent or lower value of the objective function. If the solution to the rigorous formulation is the optimal one, then the associated solution of the cluster-based approach will also feature the lowest objective value. Let us consider a feasible solution **S** of the rigorous model. To derive the associated solution of the cluster-based approach, an iterative procedure generating groups of batches with the same product, due date, and assigned unit is applied. In each iteration, two batches (b*, b) featuring the same product i and due date d, and assigned the same unit, but not successively processed, are put together to generate a cluster g_{idj} (see Figure C1). The cluster-generating procedure comprises two steps: (I) *Identify a pair of batches to be merged into a cluster*. Batches (b*, i) and (b, i) (with $b^* < b$) are associated to the same order (i, d) and allocated to unit j in the schedule **S** shown in Figure C1. However, they are not successively processed. Then, the grouping step can be applied to batches (b*, b). If no candidate batches for merging are found, we proceed to the next unit j. If all the units have been considered, the procedure is stopped and a solution of the cluster-based model associated to the schedule **S** has been developed. (II) *Perform a grouping step on the pair* (b*, b). Let (b_0, i_0) be the batch processed immediately before batch (b*, i), if any. Besides, let {(b_1, i_1), ..., (b_n, i_n)} be the sequence of batches processed between (b*, i) and (b, i) in unit j. Postpone batch (b*, i) downward as in Figure C1 so that it is processed right before the companion batch (b, i). Return to step I.

At step II, the processing times do not change since the same batches are still allocated to unit j. Only the changeover times $\tau_{i0,i,j}$ and $\tau_{i,i1,j}$ are replaced by $\tau_{i0,i1,j}$ and $\tau_{i,i,j}$. If condition 1 holds, the overall changeover time at unit j does not increase, and therefore, the starting/completion of batch (b, i) shows no delay. Sometimes, the processing of batch (b, i) may even be anticipated. Beyond batch (b, i), no change occurs. Then, the makespan and the tardiness of order (i, d), if determined by batch (b, i), are not deteriorated at all by the grouping step. A similar statement holds for the tardiness of batches {(b_1, i_1), ..., (b_n, i_n)}, which can be anticipated because of condition 2. As a result, the overall tardiness is not increased by step II. Consequently, the proposition has been proved.

Nomenclature

Subscripts

b = batch
d = due date
i = product
j = equipment unit

Sets

B_i = tentative batches for product i
B_{id} = tentative batches for product i associated to due date d
D_i = due dates of product i
I = products
J = available equipment units
J_i = equipment units available for product i

Parameters

ε_{id} = weighting penalty for the tardiness of order (i,d)
δ = minimum difference between due dates required to pre-order batches
Δ_{ij} = difference between $q_{ij}^{\max}$ and $q_{ij}^{\min}$
$\tau_{ii'j}$ = sequence-dependent changeover time
bs_i = reference batch size for product i
ft_{ij} = fixed processing time for product i at unit j
H = length of the scheduling horizon
nb_i = estimation of the maximum number of batches needed to satisfy the total demand of product i
nb_{id} = estimation of the number of batches needed to satisfy the overall requirement of product i up to due date d
$q_{ij}^{\min}$ = minimum batch size for product i at unit j
$q_{ij}^{\max}$ = maximum batch size for product i at unit j
r_{id} = requirement of product i at delivery date d
rt_{id} = release time of order (i,d)
vt_{ij} = variable processing time rate for product i and unit j

Binary Variables

Y_{bij} = binary variable denoting that batch b of product i is allocated to unit j
$X_{bi,b'i'}$ = binary variable denoting that batch (b, i) is run before or after batch (b', i') if both are allocated to the same unit
$X^G_{id,i'd',j}$ = binary variable denoting that cluster g_{idj} is run before or after cluster $g_{i'd'j}$

Continuous Variables

BS_{bi} = size of batch (b, i)
CT_{bi} = completion time of batch (b, i)
CT^G_{idj} = completion time of cluster g_{idj}
MK = makespan
PT_{bij} = processing time of batch b of product i at unit j
Q_{bij} = variable portion of the size of batch (b, i) at unit j
ST_{bi} = starting time of batch (b, i)

ST^G_{idj} = starting time of cluster g_{idj}
T_{id} = tardiness of order (i,d)
Y^G_{idj} = continuous variable indicating the existence of cluster g_{idj}

Literature Cited

(1) Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput. Chem. Eng.* **2004**, *28*, 2109.

(2) Méndez, C. A.; Cerdá, J.; Grossmann, I. E.; Harjunkoski, I.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* **2006**, *30*, 913–946.

(3) Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations - I. MILP formulation. *Comput. Chem. Eng.* **1993**, *17*, 211–227.

(4) Pantelides, C. C. Unified frameworks for optimal process planning and scheduling. *Proceedings of the Second International Conference on Foundations of Computer-Aided Process Operations*, July 18−23, 1993, Crested Butte, CO, 1994; pp 253−274.

(5) Rodrigues, M. T. M.; Latre, L. G.; Rodrigues, L. C. A. Short-term planning and scheduling in multipurpose batch chemical plants: a multi-level approach. *Comput. Chem. Eng.* **2000**, *24*, 2247–2258.

(6) Giannelos, N. F.; Georgiadis, M. C. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. *Ind. Eng. Chem. Res.* **2002**, *41*, 2178–2184.

(7) Maravelias, C. T.; Grossmann, I. E. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.* **2003**, *42*, 3056–3074.

(8) Castro, P. M.; Barbosa-Póvoa, A. P.; Matos, H. A.; Novais, A. Q. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind. Eng. Chem. Res.* **2004**, *43*, 105–118.

(9) Ierapetritou, M. G.; Floudas, C. A. Effective continuous-time formulation for short-term scheduling: I. Multipurpose batch processes. *Ind. Eng. Chem. Res.* **1998**, *37*, 4341–4359.

(10) Ierapetritou, M. G.; Hené, T. S.; Floudas, C. A. Effective continuous-time formulation for short-term scheduling: 3. Multiple intermediate due dates. *Ind. Eng. Chem. Res.* **1999**, *38*, 3446–3461.

(11) Janak, S. L.; Lin, X.; Floudas, C. A. Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. *Ind. Eng. Chem. Res.* **2004**, *43*, 2516–2533.

(12) Pinto, J. M.; Grossmann, I. E. A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Ind. Eng. Chem. Res.* **1995**, *34*, 3037–3051.

(13) Castro, P. M.; Grossmann, I. E. New continuous-time MILP model for the short-term scheduling of multistage batch plants. *Ind. Eng. Chem. Res.* **2005**, *44*, 9175.

(14) Cerdá, J.; Henning, G. P.; Grossmann, I. E. A mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. *Ind. Eng. Chem. Res.* **1997**, *36* (5), 1695–1707.

(15) Gupta, S.; Karimi, I. A. An improved MILP formulation for scheduling multiproduct, multistage batch plants. *Ind. Eng. Chem. Res.* **2003**, *42*, 2365–2380.

(16) Méndez, C. A.; Henning, G. P.; Cerdá, J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput. Chem. Eng.* **2001**, *25*, 701–711.

(17) Méndez, C. A.; Henning, G. P.; Cerdá, J. Optimal scheduling of batch plants satisfying multiple product orders with different due dates. *Comput. Chem. Eng.* **2000**, *24*, 2223–2245.

(18) Lim, M.; Karimi, I. A. A slot-based formulation for single-stage multiproduct batch plants with multiple orders per product. *Ind. Eng. Chem. Res.* **2003**, *42*, 1914–1924.

(19) Méndez, C. A.; Cerdá, J. A precedence-based monolithic approach to lot-sizing and scheduling of multiproduct batch plants. *Comput.-Aided Chem. Eng.* **2007**, *24*, 679–684.

(20) Castro, P. M.; Erdirik-Dogan, M.; Grossmann, I. E. Simultaneous batching and scheduling of single stage batch plants with parallel units. *AIChE J.* **2008**, *54*, 183–193.

(21) Prasad, P.; Maravelias, C. T. Batch selection, assignment and sequencing in multi-stage multi-product processes. *Comput. Chem. Eng.* **2008**, *32*, 1106–1119.

(22) Sundaramoorthy, A.; Maravelias, C. T. Simultaneous batching and scheduling in multistage multiproduct processes. *Ind. Eng. Chem. Res.* **2008**, *47*, 1546–1555.

(23) Sundaramoorthy, A.; Maravelias, C. T. Modeling of storage in batching and scheduling of multistage processes. *Ind. Eng. Chem. Res.* **2008**, *47*, 6648–6660.

(24) Sundaramoorthy, A.; Maravelias, C. T.; Prasad, P. Scheduling of multistage batch processes under utility constraints. *Ind. Eng. Chem. Res.* **2009**, *48*, 6050–6058.

(25) Marchetti, P. A.; Cerdá, J. A continuous-time tightened formulation for single-stage batch scheduling with sequence dependent changeovers. *Ind. Eng. Chem. Res.* **2009**, *48*, 483–498.

Received for review January 9, 2010
Revised manuscript received May 4, 2010
Accepted May 5, 2010

IE100054H