I&EC research
Industrial & Engineering Chemistry Research
Article
pubs.acs.org/IECR

Scheduling of Multistage Multiproduct Batch Plants Operating in a Campaign-Mode

Yanina Fumero,[†] Gabriela Corsano,[†] and Jorge M. Montagna[†,‡,*]

[†]INGAR—Instituto de Desarrollo y Diseño (CONICET-UTN), Avellaneda 3657, (S3002GJC) Santa Fe, Argentina

[‡]CIDISI—Centro de Investigación y Desarrollo en Ingeniería en Sistemas de Información, Universidad Tecnológica Nacional, Facultad Regional Santa Fe, Argentina

ABSTRACT: In this work, mixed integer linear programming models for scheduling multistage multiproduct batch plants operating under campaign mode are proposed. It is assumed that each plant stage includes identical parallel units operating out of phase. Given the plant topology and the number of batches of each product to be processed in the campaign, the objective is assigning batches to units in each stage in order to minimize the cycle time of the campaign. An asynchronous slot-based continuous-time representation for modeling the assignment of batches to units is used. These formulations require postulating a priori a suitable number of production slots for each unit that integrates the plant, which severely affects the model computational performance. Then, to reduce the computational effort, a solution strategy is proposed where a simplified model, which includes preordering constraints, is first solved. Finally, a detailed scheduling model is posed where the optimal cycle time of simplified model is used as bound for the cycle time and a novel expression for the number of proposed slots for each unit is considered. The strategy is highlighted through examples that show how the computational burden is reduced.

1. INTRODUCTION

Batch processing plants are characterized by their flexibility and ability to produce small quantities of high value-added products, sharing the same equipment. Given that a wide variety of different products can be manufactured in these plants, the production must be managed in order to improve plant operability, reduce idle times, avoid large inventory levels, etc. Therefore, a scheduling problem must be solved. Given the plant structure, this problem determines how resources that are shared among different products must be used in an optimal manner.

Batch process scheduling has been extensively researched over the past decades and, nowadays, it attracts the interest of both academic and practitioners' communities.

There are a lot of scheduling problems and optimization approaches to solve them in the batch process industry. They are classified according to a great diversity of factors and characteristics, which makes quite difficult the task of providing an unified model that address all the possible scenarios.

In the case of plants that must satisfy demands with high variability over the contemplated time horizon, the short-term operation is appropriate. In this case, the scheduling horizon is relatively short (e.g., several days) and the production plan must be determined for each period. Interesting reviews on short-term scheduling of batch processes have been reported by Pinto and Grossmann,[1] Kallrath,[2] Floudas and Lin,[3,4] Méndez et al.,[5] and Pan et al.[6] In particular, Méndez et al.[5] have presented a very detailed review of numerous modeling and optimization approaches based in Mixed Integer Linear Programming (MILP) methods, considering the computational performance, capabilities, and limitations of the resulting optimization models. Also, other modeling and solution paradigms, including Meta-Heuristics (Genetic Algorithms, Simulated Annealing, Tabu Search, etc.), Constraint Programming, and Artificial Intelligence techniques, have been discussed by these authors. These methods can also be combined with mathematical programming formulations, which give rise to hybrid methods. Mouret et al.[7] develop a unified modeling approach for batch scheduling problems in order to facilitate the evaluation of several time representations, both in terms of computational time and solution quality. The authors study four different types of time representations used in the literature, and clarify the relationships between them.

In contrast, when product demands are stable or can be accurately forecasted during a relatively long time horizon (e.g., several weeks or months), more efficient management and control of the production resources can be achieved if the plant is operated in a campaign-mode, that is, in a cyclic mode. A campaign includes several batches of different products that are going to be manufactured, and the same pattern is repeated at a constant frequency over the time horizon. In this case, given the campaign composition, the goal is to determine the batch sequencing in the campaign and the number of times that it is cyclically repeated.

The periodic mode of operation is very useful, especially in make-to-stock production systems. The basic advantages are, among others, more standardized production during certain periods of time, easier and more profitable operations decisions, and adequate inventory levels without generating excessive costs and minimizing the possibility of stock-outs. From the computational point of view, the cyclic scheduling allows reducing the size of the overall scheduling problem, which is often intractable.

While many papers have been published dealing with short-term scheduling of multistage multiproduct batch plants with

Received: August 8, 2011
Revised: February 7, 2012
Accepted: February 10, 2012
Published: February 11, 2012

dx.doi.org/10.1021/ie201757t | Ind. Eng. Chem. Res. 2012, 51, 3988−4001

parallel units and different storage configurations, few works have addressed the operation with campaigns for this type of plants. Approaches considering different representations and solution strategies have been presented by other authors. For example, Birewar and Grossmann[8] developed slot-based formulations for scheduling multiproduct batch plants that operate under production campaigns, considering different transfer policies (unlimited intermediate storage and zero-wait (ZW)). They determined the optimal cycle time of the campaign using a MILP formulation. However, they worked with simple multiproduct plants including only one unit per processing stage. Shah et al.[9] proposed a MILP formulation based on the state-task-network (STN) representation and used a discrete time representation for the scheduling of batch plants operated in a cyclic mode. The optimal cycle time within given bounds is determined by solving a sequence of fixed cycle time problems. In this case, the main limitation is the time discretization, which can lead to suboptimal solutions. Schilling and Pantelides[10] presented a mathematical formulation based on the resource-task-network (RTN) process representation with a continuous time representation for cyclic scheduling problem. The cycle time is divided into a number of intervals of variable duration so that all events taking place during the cycle coincide with one of the interval boundaries. The formulation results in a Mixed Integer Nonlinear Programming (MINLP) problem that cannot be exactly linearized. Therefore, a special branch-and-bound algorithm that branches on both discrete and continuous variables was proposed for solving this problem. Castro et al.[11] considered periodic scheduling for a real industrial batch plant. Both discrete and continuous time formulations, based on the RTN representation, were employed. However, their continuous time formulation is limited to small-sized problems. Wu and Ierapetritou[12] solved the cyclic scheduling of multiproduct batch plants using a MINLP model based on the STN representation of the plant and a continuous time formulation.

One of the main differences between periodic scheduling using campaigns and the short-term scheduling is the used objective function. Here, the same production sequence, the campaign, must be repeatedly executed during the available time horizon. To eliminate idle times between campaigns as much as possible and taking into account the parallel units in each stage, consecutive campaigns are overlapped. Makespan minimization cannot be employed as an appropriate optimization criterion. Instead, the cycle time of the campaign is more adequate, as will be discussed. Hence, the initial and final times of the campaign in each unit must be calculated, and consequently, the units of each stage must be individually treated. Assuming that normally the number of times that the campaign is repeated over the time horizon is large, the startup and shutdown phases of the global schedule can be neglected, and consequently, the total time horizon for the plant operation through campaigns can be approximated by the product of the campaign cycle time and the number of times that the campaign is carried out.

Focusing on approaches based on mathematical programming, the models using STN and RTN representations are the most general for addressing almost all classes of batch scheduling problems. They can handle network batch process with arbitrary topology, batch splitting/mixing, mass balances, discrete and continuous time representations, all types of intermediate storage and transfer policies, limitations of resources, and a wide range of objective functions. Castro et al.[13] addressed the optimal periodic scheduling of an industrial plant dedicated to the synthesis of active pharmaceutical ingredients. Compared with other works, they present a more realistic objective function that permits selection from the set of available equipment units, with those units that are better-suited for the production of the ingredient being considered. New scheduling approaches based on these types of representations have been also presented by Castro and Novais[14] and Shaik and Floudas.[15] They consider different intermediate storage options and alternative objective functions. In addition, the computational performance of their formulations are compared with a previous one presented in the literature.

However, Prasad and Maravelias[16] address that it is not clear how models using this type of representations can be simplified to exploit the sequential structure of multistage multiproduct process, as is the case studied in this paper. Because of this limitation, several articles propose less general but computationally more effective MILP formulations for the scheduling of multistage batch plants.

Liu and Karimi[17] studied the short-term scheduling problem of multistage, multiproduct batch plants with parallel units and ZW transfer policy. They analyzed and compared different approaches for the representation of this problem. They found that a slot-based continuous-time formulation works better for sequencing batches on stages with nonidentical parallel units, while a sequence-based continuous-time formulation is the best representation for stages with identical parallel units. Although this paper is focused on plants with identical units in each stage, as previously mentioned, the operation mode with campaigns requires that each unit is treated as an individual resource, and therefore, a slot-based continuous-time approach is more suitable.

To reduce the combinatorial complexity of the scheduling problem of multistage multiproduct plants with identical parallel units under a slot-based continuous-time formulation, Liu and Karimi[17] used asynchronous slots at each stage rather than each unit and remove the uncertainty of guessing the numbers of slots. In their work, each stage is treated as a black box where batches enter and exit without distinguishing among the involved individual units. Hence, the number of postulated slots for each stage is equal to the total number of batches to be processed. Besides, taking into account that the assignment of batches to specific units is avoided, the number of binary variables is greatly reduced. This representation is appropriate since the scheduling objective is the makespan minimization, which corresponds to the total time required to produce a given number of batches.

However, when the plant operates in campaign-mode, the cycle time must be minimized, and therefore, the initial and final operation times of the first slot and last slot assigned to each individual unit, respectively, must be calculated. In this case, the slot-based representation requires postulating a priori an appropriate number of production slots for each unit that integrates the plant, which is a key decision for the model computational performance.

In this paper, a formulation and resolution strategy based on MILP programming are presented for determining the optimal sequence of batches that compose the production campaign on multistage multiproduct batch plants with parallel units and ZW storage policy. Broadly, given the plant topology and the number of batches of each product to be processed in the campaign, the objective is assigning batches to units in each stage in order to minimize the cycle time of the campaign. Taking into account that each unit is individually treated, the combinatorial complexity of the problem is increased, and therefore, a two-phase resolution strategy is proposed. In the first phase, a simplified slot-based continuous-time MILP model is solved. This formulation involves preordering constraints (heuristic rules) on the assignment of batches to slots in each stage, which affect the optimality of the detailed scheduling model. Nevertheless, the objective function value of this simplified model represents a good upper bound for the cycle time of the detailed scheduling problem, which is solved in the second phase. A novel expression for the number of slots postulated for each unit, which guarantee the optimality of the solution and significantly reduce the computational effort, is proposed for the detailed model.

The proposed methodology allows obtaining a production schedule for each unit, including initial and final operation times for each batch that integrate the production campaign, in a reasonable CPU time.

The remainder of this paper is structured as follows: First, the addressed problem and the main assumptions are introduced in section 2. Then, in the subsequent section, a resolution strategy is presented. The MILP formulations that integrate the resolution scheme are developed in section 4. Finally, several examples are considered to demonstrate the applicability of the proposed hierarchical approach and to assess its performance.

2. PROBLEM DEFINITION

Let J be the set of processing stages that compose the plant and K be the set of all units in the plant. A set I of batches, where some of them may be of identical products, must be manufactured in the plant following the same sequence of stages. The elements of the sets I, J, and K are identified with the indexes $i = 1, 2, ..., |I|$; $j = 1, 2, ..., |J|$; and $k = 1, 2, ..., |K|$; respectively. Let K_j be the subset ($K_j \subseteq K$) of identical batch units that operate in parallel and out-of-phase in stage j. Then

$$K = K_1 \cup K_2 \cup ... \cup K_{|J|}$$

$$K_j \cap K_{j'} = \emptyset \quad \forall\, j,\ j' \in J,\ j \neq j'$$

If the cardinality of K_j is N_j, then the elements of each set K_j can be represented by: $K_1 = \{1, 2,..., N_1\}$, $K_2 = \{N_1+1, N_1+2, ..., N_1+N_2\}$, $K_3 = \{N_1+N_2+1, ..., N_1+N_2+N_3\}$, and so on. In general,

$$K_j = \left\{ \sum_{s=1}^{j-1} N_s + 1,\ ...,\ \sum_{s=1}^{j} N_s \right\} \qquad \forall\ 1 < j \leq |J|$$

Interstage storage is not allowed. Therefore, taking into account the configuration of the plant, there is no batch splitting or mixing; that is, each batch is treated as a discrete entity through the whole process. Considering that intermediate storage tanks are not allocated between stages and that a batch can not wait in the unit after finishing its processing, the ZW transfer policy between stages is adopted; that is, after being processed in stage j, a batch i is immediately transferred to the next stage $j+1$. Besides, batch transfer times between units are assumed very small compared to process operation times and, consequently, negligible, or they are included in the processing times. On the other hand, since in this type of plants products recipes are similar, sequence-dependent changeovers times can be assumed negligible and, therefore, they are not considered in the current formulation. Finally, since parallel units in each stage j are assumed to be identical, a batch i can be processed on any unit with the same processing time t_{ij}.

As previously stated, the problem consists of determining the assignment of batches to units in each stage, the production sequence on each unit, and initial and final processing times for batches that compose the campaign in each processing unit.

3. SOLUTION STRATEGY

Production scheduling in multistage multiproduct batch plants is a complex optimization problem. As previously mentioned, very different modeling approaches and solution strategies are analyzed in the literature on this area. Nevertheless, a number of major challenges and questions remain unsolved.[5]

In this paper, the main characteristic of the proposed formulations is the use of an asynchronous slot-based continuous-time representation for modeling the assignment of batches to units.[8,18−21] The slots correspond to time intervals of variable length where batches will be assigned. In each slot l of a specific unit k, at most one batch i can be processed, and if no product is assigned to slot l, its length will be zero. In this work, the number of postulated slots for each unit can differ from one unit to another.

In models based on this type of representation, the selection of the number of slots postulated for each unit is a not trivial decision because of the computational performance strongly depends on this parameter. In general, reduced values compromise the optimality of the model, and on the contrary, higher values severely affect its computational performance.

Pinto and Grossmann[18] proposed a model and a solution strategy for the short-term scheduling of multistage batch plants that may contain parallel units. In their formulation, under the assumption that production is uniformly allocated to similar units, the calculation of the number of time slots is based on the number of batches allowed to be processed in a unit of a stage and the number of parallel units of this stage. Therefore, the proposed expression does not guarantee the solution optimality. Then, Ierapetritou and Floudas[22] proposed an iterative procedure to determine the number of slots. Their proposal starts with a small number of slots, and then, gradually increases this number by 1, until no improvement in the optimal objective function can be achieved. This technique has been extensively used in scheduling fields. However, this iterative procedure does not offer theoretical proof of optimality, and it can result in suboptimal solutions in some cases. On the other hand, Chen et al.[19] employed heuristic rules for the number of postulated slots for each unit to cut down the size of the proposed model. These rules are limited to the case of short-term scheduling of multiproduct single-stage batch plants with parallel lines. More recently, Li and Floudas[23] developed a general framework to obtain the optimal number of slots for short-term scheduling of multipurpose batch plants. Their proposal mainly focuses on intermediate storage requirements such as unlimited intermediate storage (UIS) and finite intermediate storage (FIS).

In this paper, the number of slots postulated for each unit is tightly proposed in each formulation to reduce the set of alternative solutions without affecting the model optimality.

To decrease the computational expense of solving the detailed scheduling problem, a solution strategy is presented in this work based on the decomposition of the problem in two MILP models.

First, a simplified MILP problem (SP) is solved, whose complexity is reduced with respect to the detailed MILP scheduling problem (DP). In problem SP, a preordering constraint in the assignment of batches is introduced in order to diminish the search space. This constraint requires that the assignment of batches to slots follows the same relationship in all the stages; that is, a product batch is processed in exactly the same slot in all stages. The solution of this model provides a good upper bound on the objective function of problem DP, and many times, it will coincide with its global optimum. Therefore, using the optimal value of the objective function of problem SP, the detailed MILP formulation for the scheduling of a campaign in a multiproduct, multistage batch plant with parallel units can be solved in a reasonable computational time.

As will be shown in the Results and Discussion section, the upper bound for the cycle time, obtained from the objective function of model SP usually coincides with the optimal solution of DP.

4. MATHEMATICAL FORMULATIONS

In this section, the proposed models that integrate the solution strategy previously presented are described. Both models are slot-based MILP formulations, where a continuous-time domain representation is employed and the slots are asynchronous across the parallel units. However, the handling of the slot concept in both models is slightly different.

As was already mentioned, the number of slots postulated for unit k of stage j, denoted by L_{kj}, is not known a priori because the set of batches assigned to each unit is an optimization variable. This number severely affects the computational performance of the model, and its selection is crucial for keep the model optimality. Taking into account that units in each stage are identical, the more conservative option of assuming $L_{kj} = |I|$ for each unit k can be relaxed. An expression for the parameter L_{kj} is proposed in each formulation, which depends on how slots are handled in the allocation decisions of each model.

4.1.1. Simplified Model SP. *Maximum Number of Slots Postulated for Each Unit.* If k^j_{first} denotes the first unit of stage j, then the possible number of slots considered for this unit is equal to the number of batches that must be processed in the plant. In particular, if a stage consists of exactly one unit, then $|I|$ slots are assigned to it for processing all batches.

Taking into account that units in each stage are identical, it is assumed that they are used in ascending order. This keeps the model generality and allows reducing the number of slots proposed for each unit k. Then, for each unit k, $k \in K_j - \{k^j_{\text{first}}\}$, as at least one slot is occupied in each previous unit to it, the maximum number of slots postulated for unit k of stage j is given by $L_{kj} = L_{k-1j} - 1$ (Figure 1). Thus, the proposed number of slots for each unit of stage j can be recursively defined as

$$L_{kj} = |I| \quad k = k^j_{\text{first}} \quad j \in J \tag{1}$$

$$L_{kj} = L_{k-1j} - 1 \quad k \in K_j - \{k^j_{\text{first}}\} \quad j \in J \tag{2}$$

This assumption improves the performance when $|I|$ and $|K_j|$ are large.

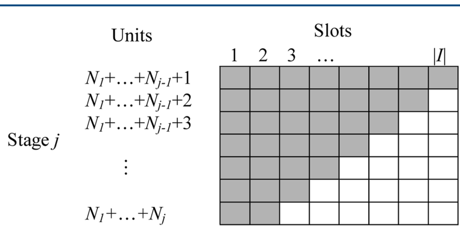

Figure 1. Proposed slots representation for each unit of stage j for model SP.

4.1.2. Assignment Constraints. Batches must be assigned, in each stage, to specific slots of units. In this formulation, a preordering constraint is imposed in the assignment of batches, which allows simplifying the definition of the allocation variables, reducing the number of indices needed. This heuristic rule assures that each batch is assigned to the same slot on all stages. Then, the assignment of batches to slots on different units is defined through two sets of binary variables:

$$Z_{il} = \begin{cases} 1 & \text{if batch } i \text{ is assigned to slot } l \\ 0 & \text{otherwise} \end{cases}$$

$$X_{kl} = \begin{cases} 1 & \text{if slot } l \text{ of unit } k \text{ is employed for processing} \\ & \quad \text{a batch} \\ 0 & \text{otherwise} \end{cases}$$

Note that variable Z_{il} defines the batch−slot relation. Therefore, the proposed heuristic rule, which states that each batch must be allocated to the same slot on different stages, is posed through the following condition:

$$\sum_{1 \le l \le |I|} Z_{il} = 1 \quad \forall\, i \in I \tag{3}$$

Moreover, for each stage of the plant, slot l is only used for processing one batch on exactly one unit, then

$$\sum_{i \in I} Z_{il} = 1 \quad 1 \le l \le |I| \tag{4}$$

$$\sum_{\substack{k \in K_j \\ k/l \le L_{kj}}} X_{kl} = 1 \quad \forall\, j \in J,\ 1 \le l \le |I| \tag{5}$$

In other words, if slot l is proposed in different units of the same stage, this slot is only used in one of them to process a product batch. In the remaining units of that stage, this slot cannot be occupied. The length of empty slots is zero, and therefore, initial and final times are equal and coincide with the end time of the previous slot. Then, taking into account that the number of slots proposed in each unit is overestimated, some of them may be empty. For example, Figure 2 shows the slots proposed for each unit of a stage j and slots effectively used for processing batches, which are indicated by an X. In this case, according to the rule previously mentioned, slots 1 and 6 are only used on unit 2, slots 2 and 4 are used on unit 3, and slots 3, 5, and 7 are only used on unit 1.

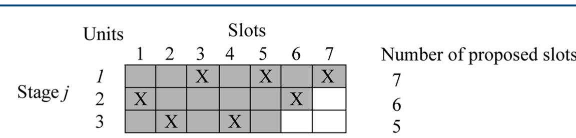

Figure 2. Illustration of the rule represented by eqs 4 and 5.

To reduce the search space, others assumptions about slots utilization are introduced in this formulation. Without loss of generality, the following constraint is imposed:

$$\sum_{\substack{l \\ l \le L_{kj}}} 2^l X_{kl} \ge \sum_{\substack{l \\ l \le L_{k+1j}}} 2^l X_{k+1l} \quad \forall\, j \in J,\ k,\ k+1 \in K_j \tag{6}$$

This inequality establishes that the succession formed by the weighted sum of the slots occupied in each unit of a stage forms a decreasing succession. Note that, taking into account that the *nth* power of 2 is greater than the sum of the previous $n - 1$ powers, the last slot, $l = |I|$, will be always used on the first unit for processing a product batch.

The proposed preordering constraint accelerates the MILP model resolution, reducing the number of nodes enumerated. However, in some cases, suboptimal solutions can be obtained.

4.1.3. Processing Times of the Slots. Nonnegative continuous variables, TI_{kl} and TF_{kl}, are used to represent the initial and final processing times, respectively, of the proposed slots in each unit k. The relation between these variables and the binary variables Z_{il} and X_{kl} can be written as

$$TF_{kl} = TI_{kl} + \sum_{i \in I} t_{ij} Z_{il} X_{kl} \quad \forall\, j \in J,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{7}$$

Taking into account that slot l in unit k may not be necessarily occupied, then, when no batch is assigned to it (i.e., $X_{kl} = 0$), the initial and final times of this slot are equal (i.e. $TI_{kl} = TF_{kl}$).

Constraint 7 is nonlinear because of the bilinear product $Z_{il}X_{kl}$. Then, to eliminate this nonlinearity, a new nonnegative variable Y_{ikl} is defined as

$$Y_{ikl} = \begin{cases} 1 & \text{if both } Z_{il} \text{ and } X_{kl} \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Thus, eq 7 can be rewritten as

$$TF_{kl} = TI_{kl} + \sum_{i \in I} t_{ij} Y_{ikl} \quad \forall\, j \in J,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{8}$$

Given that variables X_{kl} and Z_{il} are binary, Y_{ikl} can be treated as a continuous variable in the interval [0, 1]. Thus, the number of binary variables is reduced. Taking into account that if batch i is not assigned to slot l, then none of the units of each stage can employ slot l to process batch i, and that if slot l of unit k at stage j is not utilized, then none of the products is processed in it, the following logical implications must be true:

$$Z_{il} = 0 \Rightarrow Y_{ikl} = 0 \quad \forall\, j \in J,\ i \in I,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{9}$$

$$X_{kl} = 0 \Rightarrow Y_{ikl} = 0 \quad \forall\, j \in J,\ i \in I,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{10}$$

The above propositions are guaranteed by the following constraints:

$$\sum_{\substack{k \in K_j \\ k/l \le L_{kj}}} Y_{ikl} = Z_{il} \quad \forall\, i \in I,\ j \in J,\ 1 \le l \le |I| \tag{11}$$

$$\sum_{i \in I} Y_{ikl} = X_{kl} \quad \forall\, j \in J,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{12}$$

Reciprocally, Y_{ikl} must take value 1 only if both X_{kl} and Z_{il} are one. Therefore, the following proposition must be true:

$$Z_{il} = 1 \wedge X_{kl} = 1 \Rightarrow Y_{ikl} = 1 \quad \forall\, j \in J,\ i \in I,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{13}$$

This logical relation is enforced by the following linear inequality:

$$Y_{ikl} \ge X_{kl} + Z_{il} - 1 \quad \forall\, j \in J,\ i \in I,\ k \in K_j,\ 1 \le l \le L_{kj} \tag{14}$$

In summary, if $Z_{il} = 0$, then, taking into account constraint 11 and that variable Y_{ikl} is not negative, $Y_{ikl} = 0$ for all unit k. Similarly, if $X_{kl} = 0$, then, taking into account constraint 12, $Y_{ikl} = 0$ all batch i. In both cases, inequality 14 is redundant. On the contrary, if Z_{il} and X_{kl} are simultaneously equal to 1, taking into account that the upper bound for continuous variable Y_{ikl} is equal a 1, constraint 14 ensures that $Y_{ikl} = 1$.

In this way, the assignment of variable Y_{ikl} does not need to be defined as binary, and hence, the number of binary variables of the model remains unchanged.

Considering eqs 3 and 11 and expressions 5 and 12, the following constraints must be satisfied. Though they are redundant with the previous ones, they reduce the resolution time:

$$\sum_{k \in K_j} \sum_{1 \le l \le L_{kj}} Y_{ikl} = 1 \quad \forall\, i \in I,\ j \in J \tag{15}$$

$$\sum_{i \in I} \sum_{\substack{k \in K_j \\ k/l \le L_{kj}}} Y_{ikl} = 1 \quad \forall\, j \in J,\ 1 \le l \le |I| \tag{16}$$

To avoid overlapping between the processing times of different slots in a unit, the following constraint is added:

$$TF_{kl} \le TI_{kl+1} \quad \forall\, j \in J,\ k \in K_j,\ 1 \le l < L_{kj} \tag{17}$$

Besides, if no batch is assigned to slot l+1 of unit k ($X_{kl+1} = 0$), then the initial time of this slot is enforced to be equal to the final time of slot l. Then, taking into account that eq 17 is satisfied for successive slots in a unit, this new condition is represented by

$$TF_{kl} - TI_{kl+1} \ge -M_1 X_{kl+1} \quad \forall\, j \in J,\ k \in K_j,\ 1 \le l < L_{kj} \tag{18}$$

where M_1 is a sufficiently large number that makes the constraint redundant when a batch is assigned to slot l+1.

4.1.4. Zero-Wait Transfer Policy. As already mentioned, the ZW transfer policy assumes that a batch, after finishing its processing at a stage, must be transferred immediately to the next stage. Therefore, when a batch processed in slot l utilizes unit k in stage j and k' in stage j+1, the following equation must be satisfied:

$$TF_{kl} = TI_{k'l} \quad \forall\, j,\ j+1 \in J,\ k \in K_j,\ k' \in K_{j+1},\ k/X_{kl} = 1,\ k'/X_{k'l} = 1,\ 1 \le l \le \min\{L_{kj},\ L_{k'j+1}\} \tag{19}$$

Given that this constraint must be only satisfied when a batch is assigned to those units, then this condition can be expressed through constraints of Big-M type, as

$$TF_{kl} - TI_{k'l} \geq M_2(X_{kl} + X_{k'l} - 2) \quad \forall\, j,\ j+1 \in J,\ k \in K_j,\ k' \in K_{j+1},\quad 1 \leq l \leq \min\{L_{kj},\ L_{k'j+1}\} \tag{20a}$$

$$-TF_{kl} + TI_{k'l} \geq M_2(X_{kl} + X_{k'l} - 2) \quad \forall\, j,\ j+1 \in J,\ k \in K_j,\ k' \in K_{j+1},\quad 1 \leq l \leq \min\{L_{kj},\ L_{k'j+1}\} \tag{20b}$$

where M_2 is a sufficiently large number that relaxes these constraints when the batch processed in slot l does not utilize unit k in stage j or k' in stage j+1.

4.1.5. Campaign Cycle Time. An expression for the cycle time of the campaign, CT, must be obtained. Using the initial and final times of the proposed slots for each unit in all stages, CT is calculated as

$$CT = \max_{j \in J}\{\max_{k \in K}\{TF_{kL_{kj}} - TI_{k\tilde{l}_k}\}\} \tag{21}$$

where $\tilde{l}_k$ represents the first slot effectively used in unit k for processing one batch; that is, $\tilde{l}_k = \min\{\ 1 \leq l \leq L/X_{kl} = 1\}$. This equation can be represented using a Big-M representation, as

$$CT - TF_{kL_k} + TI_{kl} \geq M_3\Big((X_{kl} - 1) - \sum_{\substack{l' \\ 1 \leq l' < l}} X_{kl'}\Big) \quad \forall\, j \in J,\ k \in K_j,\ 1 \leq l \leq L_{kj} \tag{22}$$

where M_3 is a sufficiently large number that makes the constraint redundant for all the previous and subsequent slots, if any, to the first not empty one in unit k.

4.1.6. Additional Constraints. Scheduling formulations in the process systems engineering (PSE) community are most often expressed as mathematical programming models, especially through MILP. This approach has been widely explored in this area due to its capability for modeling a large variety of problems and the availability of powerful commercial solvers. However, taking into account the NP nature of scheduling problems, the computational complexity of MILP models greatly increases for large scale problems. Therefore, the incorporation of additional constraints into the formulation of a MILP model may notably reduce the required computational effort for solving the optimization problem. These constraints may refer to equations indeed redundant, bounds for variables, explicit relationships among binary variables, etc.[4]

In the proposed formulation, additional constraints are incorporated to reduce the search space. New bounds for timing variables are added. Upper and lower bounds for the initial and final times, respectively, of proposed slots in each unit of a stage are considered:

$$TI_{kl} \leq TFMax_j - \min_i\{t_{ij}\} \quad \forall\, j \in J,\ k \in K_j,\ 1 \leq l \leq L_{kj} \tag{23}$$

$$TF_{kl} \geq \min_i\{t_{ij}\} X_{kl} \quad \forall\, j \in J,\ k \in K_j,\ 1 \leq l \leq L_{kj} \tag{24}$$

The parameter $TFMax_j$ is previously calculated by solving a simpler model. It considers only one unit in each stage (i.e. without parallel units) where the objective is to schedule the same set of proposed batches. Then, $TFMax_j$ corresponds to the final time of the last assigned slot to stage j. Constraint 24 ensures a positive minimum value for the final time of slot l corresponding to unit k when a batch is processed in this slot.

Significant savings in computational time have been achieved by establishing a good lower bound on the cycle time of the campaign. In fact, if the idle time in each unit is zero during the processing of the batches, then

$$CT \geq \sum_{i \in I} \sum_{\substack{l \\ 1 \leq l \leq L_{kj}}} t_{ij} Y_{ikl} \quad \forall\, j \in J,\ k \in K_j \tag{25}$$

4.1.7. Objective Function. The goal for this simplified model is the minimization of the production cycle time. Then, the MILP formulation of model SP consists of minimizing the objective function given by

$$f = CT \tag{26}$$

subject to constraints 3−6, 8, 11, 12, 14−18, 20a, 20b, 22−25.

The objective function value of this model represents a good upper bound for the cycle time of detailed scheduling problem DP, which is presented next.

4.2. Detailed Model DP. Taking advantage on the optimal value obtained from simplified model SP, detailed model DP can be solved in a reduced space in reasonable computation time. In this formulation, preordering constraints defined in SP, which affect the optimality of the scheduling solution, are removed. On the contrary, DP considers assumptions that keep the model generality and significantly decrease the computational time. Moreover, in this formulation, a tighter new expression for the number of production slots postulated for each unit k at stage j, L_{kj}, is proposed, which is further reduced compared with the proposed value in model SP, as will be seen later.

Taking into account the preordering constraint of model SP is avoided, the definition of the variable Z_{il} does not make sense. Moreover, equations such as 4 and 11 are not valid in this new formulation, and therefore, the variable is not used.

To assign batches to slots on each unit, variable Y_{ikl}, previously used in model SP, will be again employed, with the same sense. However, taking into account the remaining used variables and assumptions of model DP, now it is defined as binary variable. Although binary variable Y_{ikl} is enough for formulating the general scheduling problem, binary variable X_{kl} will be also used to reduce the search space and, therefore, to improve the computational performance.

4.2.1. Maximum Number of Slots Postulated for Each Unit. As previously mentioned, the computational performance can be improved by introducing the assignment variable X_{kl} defined in model SP, which determines the slots set utilized in unit k for processing batches.

Without loss of generality and in order to reduce the search space, it is assumed that slots of each unit are occupied in ascending order. Hence, the slots of zero length take place at the end of each unit. The following constraint establishes that

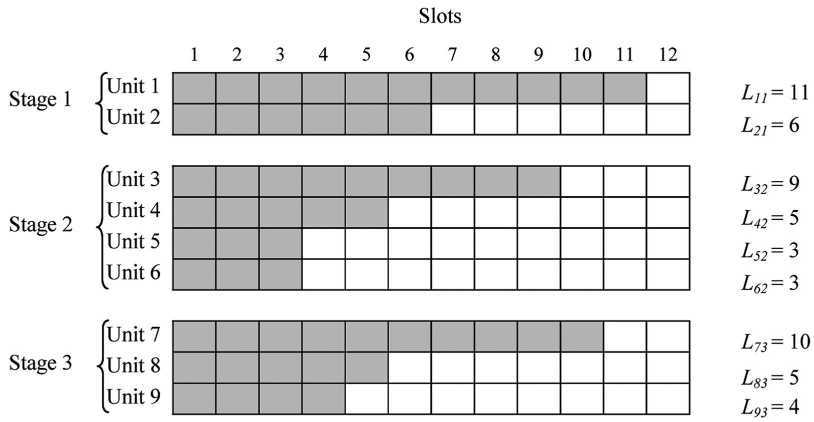

Figure 3. Slots postulated for each unit in different stages for model DP.

for each unit k, slot l+1 is only used if slot l has been already allocated:

$$X_{kl} \geq X_{kl+1} \quad \forall\, j \in J,\ k \in K_j,\ 1 \leq l \leq L_{kj} \tag{27}$$

To eliminate alternative optimal solutions and to reduce the number of postulated slots for each unit of a stage, the following constraint is used:

$$\sum_{\substack{l \\ l \leq L_{kj}}} X_{kl} \geq \sum_{\substack{l \\ l \leq L_{k+1j}}} X_{k+1l} \quad \forall\, j \in J,\ k,\ k+1 \in K_j \tag{28}$$

This equation establishes that for each stage, the number of processed slots in a unit is greater than or equal to the number of processed slots in the following unit.

Then, eqs 27 and 28 cut down the search space of the proposed model, and therefore, an improvement in computing time is obtained. Besides, the addition of these constraints into the model allows proposing a tighter number of slots for each unit.

Let p_k be the position of unit k in stage j. Then, taking into account eqs 27 and 28, the maximum number of slots postulated for this unit is specified by

$$L_{kj} = \left\lfloor \frac{|I| - |K_j| + p_k}{p_k} \right\rfloor \quad \forall\, j \in J,\ k \in K_j \tag{29}$$

where $\lfloor \cdot \rfloor$ is integer part operator.

Unlike the number of slots proposed for units of model SP, this expression can assign a different number of slots at units k, k', $k \in K_j$, $k' \in K_j'$, $j \neq j'$ where $p_k = p_{k'}$, because it depends on the cardinalities of sets K_j and $K_{j'}$, that is, the number of units of each stage.

If k is the first unit of stage j, that is, $p_k = 1$, then according to constraint 27 and assuming that at least one slot is occupied in the units of stage j whose position is greater than 1, the maximum number of slots considered for the first unit is given by $L_{kj} = |I| - (|K_j| - 1)$. Analogously, if k is the second unit of stage j, that is, $p_k = 2$, then according to constraint 27 and assuming that at least one slot is occupied in the units of stage j whose position is greater than 2, the maximum total number of slots that can be used at units of positions 1 and 2 is equal to $|I| - (|K_j| - 2)$. So, taking into account constraint 28, the maximum number of postulated slots that can be used in the second unit is $\lfloor |I| - (|K_j| - 2)/2 \rfloor$. In general, if k is the p_k-th unit of stage j, where $p_k \leq |K_j|$, and assuming that at least one slot is occupied in the units of stage j whose position is greater than p_k, according to constraint 28 the maximum number of slots considered for the p_kth-unit is given by expression 29.

As an example, consider a multiproduct plant with 3 stages, where $|I|$= 12 batches must be manufactured following the same sequence of operations. The sets of identical units that integrate the different stages are $K_1 = \{1, 2\}$, $K_2 = \{3, 4, 5, 6\}$, and $K_3 = \{7, 8, 9\}$, and the corresponding positions of the units are $p_1 = 1$, $p_2 = 2$, $p_3 = 1$, $p_4 = 2$, $p_5 = 3$, $p_6 = 4$, $p_7 = 1$, $p_8 = 2$, and $p_9 = 3$. Then, taking into account that $L = 12$ and that the function $\lfloor x \rfloor$ is the greatest integer not exceeding x, the number of slots postulated for each unit at different stages is illustrated in Figure 3.

This proposal significantly reduces the number of slots postulated at each stage when compared to that of the simplified model. For the presented example in Figure 3, the number of slots proposed in each stage is reduced by 26%, 52%, and 42.5%, respectively.

Similarly, as it was illustrated for model SP, the Figure 4 displays the conditions imposed by eqs 27 and 28 about the use of slots in each unit of each stage. In fact, the slots of each unit are consecutively used in ascending order, and on each stage, the numbers of slots utilized in each unit, starting from the first one, constitute a decreasing sequence. For this example, this sequence is 3, 2, 2.

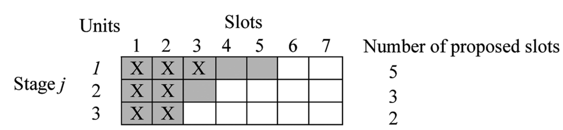

Figure 4. Illustration of the rules represented by restrictions 27 and 28

4.2.2. Assignment Constraints. To assign a batch i to slot l to be processed in unit k, the following decision variable is introduced:

$$Y_{ikl} = \begin{cases} 1 & \text{if batch } i \text{ is processed in slot } l \text{ of unit } k \\ 0 & \text{otherwise} \end{cases}$$

Then, the following constraints are imposed:

$$\sum_{i \in I} Y_{ikl} \leq 1 \quad \forall\, j \in J,\ k \in K_j,\ 1 \leq l \leq L_{kj} \tag{30}$$

$$\sum_{k \in K_j} \sum_{\substack{l \\ 1 \leq l \leq L_{kj}}} Y_{ikl} = 1 \quad \forall\, i \in I,\ j \in J \tag{31}$$

 dx.doi.org/10.1021/ie201757t | *Ind. Eng. Chem. Res.* 2012, 51, 3988−4001

Constraint 30 guarantees that slot l of unit k is utilized to process at most one batch. Moreover, eq 31 enforces that, in each stage j, batch i is processed in only one slot of some unit of this stage. This constraint coincides with eq 15. However, it is not redundant for the detailed model formulation. Note that, unlike the problem SP, using this constraint, batch i can be assigned to different slots in all the stages.

Taking into account eq 31, the following constraint must be satisfied for each stage j:

$$\sum_{k \in K_j} \sum_{\substack{l \\ 1 \le l \le L_{kj}}} X_{kl} = |I| \quad \forall\, j \in J \tag{32}$$

Logical relations can be defined among variables X_{kl} and Y_{ikl} that improve the computational performance. In fact, if slot l of unit k is not utilized, then none of the batches is processed in it. Therefore, the following proposition must be true:

$$X_{kl} = 0 \Rightarrow Y_{ikl} = 0 \qquad \forall\, i \in I,\; j \in J,\; k \in K_j,\; 1 \le l \le L_{kj} \tag{33}$$

In the same way, if slot l of unit k is employed, then only one batch is assigned to it. Reciprocally, the opposite conditional is also true. Therefore, the following logical equivalence must be satisfied:

$$X_{kl} = 1 \Leftrightarrow Y_{1kl} = 1 \vee Y_{2kl} = 1 \vee \ldots \vee Y_{|I|kl} = 1 \qquad \forall\, j \in J,\; k \in K_j,\; 1 \le l \le L_{kj} \tag{34}$$

Taking into account eq 30, the disjunction operator used in this proposition is exclusive.

The above propositions are guaranteed by the following constraints:

$$Y_{ikl} \le X_{kl} \quad \forall\, i \in I,\; j \in J,\; k \in K_j,\; 1 \le l \le L_{kj} \tag{35}$$

$$X_{kl} = \sum_{i \in I} Y_{ikl} \quad \forall\, j \in J,\; k \in K_j,\; 1 \le l \le L_{kj} \tag{36}$$

4.2.3. Processing Times of Slots. Equation 8 defined for problem SP is utilized for determining the initial and final processing times of slots. Also, constraints 17 and 18 must be included in the formulation of this model.

4.2.4. Zero-Wait Transfer Policy. To ensure this transfer policy, when a batch is processed in slot l of unit k at stage j and in slot l' of unit k' at stage j+1, the following equation must be satisfied:

$$\mathrm{TF}_{kl} = \mathrm{TI}_{k'l'} \quad \forall\, j,\; j+1 \in J,\; k \in K_j,\; k' \in K_{j+1},\; k/X_{kl} = 1,\; k'/X_{k'l'} = 1,\; 1 \le l \le L_{kj},\; 1 \le l' \le L_{k'j+1} \tag{37}$$

As this constraint must be only satisfied when a batch is assigned to those units and slots, then the condition can be expressed through constraints of Big-M type, as

$$\mathrm{TF}_{kl} - \mathrm{TI}_{k'l'} \ge M_4(Y_{ikl} - 1) + M_4(Y_{ik'l'} - 1) \quad \forall\, j,\; j+1 \in J,\; k \in K_j,\; k' \in K_{j+1},\; 1 \le l \le L_{kj},\; 1 \le l' \le L_{k'j+1} \tag{38a}$$

$$-\mathrm{TF}_{kl} + \mathrm{TI}_{k'l'} \ge M_4(Y_{ikl} - 1) + M_4(Y_{ik'l'} - 1) \quad \forall\, j,\; j+1 \in J,\; k \in K_j,\; k' \in K_{j+1},\; 1 \le l \le L_{kj},\; 1 \le l' \le L_{k'j+1} \tag{38b}$$

where M_4 is a sufficiently large number that relaxes these constraints when the batch processed in slots l and l' at stages j and j+1, respectively, does not utilize unit k in stage j or k' in stage j+1.

4.2.5. Campaign Cycle Time. Taking into account that slots of each unit are utilized in ascending order, the expression for the cycle time of the campaign, CT, is

$$\mathrm{CT} \ge \mathrm{TF}_{kL_{kj}} - \mathrm{TI}_{k1} \quad \forall\, j \in J,\; k \in K_j \tag{39}$$

4.2.6. Additional Constraints. In this formulation, the upper and lower bounds proposed in model SP for the initial and final times, respectively, of slots in each unit of every stage, are also considered. In addition, the lower bound on the cycle time of the campaign, established in eq 25, must be also imposed.

The optimal value of the objective function of model SP, $\mathrm{CT}^{\mathrm{SP}}$, represents an appropriate upper bound for the cycle time of model DP presented in this section. Then

$$\mathrm{CT} \le \mathrm{CT}^{\mathrm{SP}} \tag{40}$$

In the Results and Discussion section, the gap between the optimal values of both problems will be studied.

4.2.7. Objective Function. The goal of model DP is the minimization of the cycle time of the campaign defined by constraint 39. Then, the MILP formulation of model DP consists of minimizing the same objective function that model SP, subject to constraints 8, 17, 18, 21, 22, 27−32, 35, 36, 38a, 38b, 39, and 40.

This model allows determination of the assignment of batches to units in each stage, the production sequence on each unit, and initial and final processing times for the batches that compose the campaign in each processing unit, to minimizing the production cycle time.

5. RESULTS AND DISCUSSION

In this section, different examples are studied to illustrate the use of the proposed approach. The first example deals with a case in which the optimal solutions of models that integrate the proposed solution strategy are equal. Subsequently, a simple motivating example for showing the importance of minimizing the cycle time of the campaign is developed. Finally, the proposed strategy is evaluated solving some examples taken from the literature, and a comparative table for the computational performance is presented. All the examples were implemented and solved in GAMS[24] in an Intel Core i7, 2.8 GHz processor. The CPLEX solver was employed for solving the MILP problems, with a 0% optimality gap. The number of continuous and binary variables and constraints strongly depend on the total number of units, the number of batches of the campaign, and the number slots postulated for each unit.

5.1. Example 1. In this example, the batch plant consists of three stages with two identical parallel units operating out-of-phase on stages 1 and 3, and four units on stage 2, and it produces products A, B, C, and D, as is illustrated in Figure 5. The production campaign involves two batches of product A and D, one batch of product B, and three batches of product C; that is, a total of |I |= 8 batches must be processed on all stages, and the production sequence on each of them must be

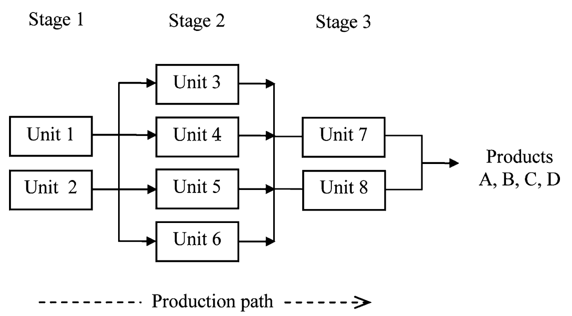

Figure 5. Plant structure for example 1.

Table 1. Processing Times of Batches for Example 1

batches	processing time: t_{ij} (h)		
	stage 1	stage 2	stage 3
I	k = 1, 2	k = 3, 4, 5, 6	k = 7, 8
1, 2 (A)	14	25	7
3 (B)	16	18	5
4, 5, 6 (C)	12	15	4
7, 8 (D)	10	20	5

determined. Data on processing times of batches are shown in Table 1.

The objective function values, constraints, and variables number, and computational statistics for both models are summarized in Table 2. Also the resolution of model DP without considering the upper bound introduced by eq 40 is included to asses the proposed strategy. This last constraint is introduced as a bound and the equations number is not modified respect to model DP.

As was mentioned earlier in the paper, the optimization criteria used for the proposed models is the minimization of the campaign cycle time. To determine this value, the difference between the final time of the last slot assigned and the initial time of the first slot assigned to each unit k of the plant must be calculated. Then, the maximum of these differences is the cycle time of the campaign. The assessment of this value is difficult for model SP because the first and last slots effectively used in each unit must be known. The first slot used is obtained through eq 22. According to eqs 7, 17, and 18, for each unit k, initial and final times of an empty slot coincide with the final time of the last slot previously used, if this exist. Therefore, the final time of the last proposed slot is considered as the final time of the last slot effectively used. Table 3 lists the initial and final processing times of model SP for each proposed slot in each unit at different stages.

In Table 3, bold intervals correspond to slots effectively used for processing a batch. As it has been already mentioned, the final and initial times of each empty slot coincide with the final time of the last previous slot used in the unit. For example, on unit 2 of stage 1, $TF_{21} = TI_{22} = TF_{22}$, $TF_{23} = TI_{24} = TF_{24}$ and $TF_{26} = TI_{27} = TF_{27}$.

For model DP, as slots of each unit are consecutively used in ascending order, only the last slot used in every unit is unknown. As it has been pointed in this work, the slots of zero length take place at the end of each unit. Then, according to the model formulation the cycle time of each unit is obtained from the difference between the final time of the last slot proposed and the initial time of the first slot proposed.

Table 4 lists the initial and final processing times of model DP for each proposed slot in each unit at different stages. Also, bold intervals correspond to slots effectively used for processing batches.

For both models, CT = max{52 − 2, 50 − 0, 72 − 22, 57 − 16, 68 − 34, 55 − 10, 77 − 30, 73 − 55} = max{50, 50, 50, 41, 34, 45, 47, 18} = 50. Then, the cycle time of the campaign for models SP and DP is the same, namely CT^{SP} = CT = 50 h, and it is achieved in units 1, 2, and 3. The optimal solution of model DP is attained considering as upper bound for the CT variable the optimal value of model SP. The optimal production sequence obtained in each batch unit for the different stages is illustrated in the Gantt chart of Figure 6. The last row of Table 2 shows the solution of model DP without considering the upper bound for CT. The computational time is greater than the previous models. Therefore, in this case, the resolution strategy in two phases has a better performance.

This example shows that there are cases where the optimal solution of the simplified model coincides with the optimal solution of the detailed scheduling model. Consequently, taking into account the NP complexity of scheduling problems, SP may be considered as a good heuristic that allows computing a good approximated solution of the exact scheduling problem.

5.2. Example 2. In this example, the importance of minimizing the cycle time of the campaign is compared with the objective usually used in most scheduling formulations, namely makespan minimization. Also, the proposed solution strategy as opposed to directly solving the detailed scheduling problem is assessed from the computational point of view.

The considered plant consists of four stages with two identical units operating out-of-phase on stages 1 and 3, and one unit on the remaining stages, as is illustrated in Figure 7. Products A, B, C, and D are manufactured following the same sequence of stages, and the number of batches in the campaign is two for products A, B, and C, and one for product D. Then, a total of $|I|$= 7 batches must be scheduled on each stage. Data on processing times of batches are shown in Table 5.

Table 6 lists the objective function values, model sizes and computational statistics for the formulations that integrate the proposed solution strategy, as well as for model DP when the optimal value of the objective function from model SP is not considered.

The optimal objective values of models SP and DP are not equal. The cycle time of the campaign for the simplified model is CT^{SP} = 33 h, whereas for the detailed model is CT = 29 h. As can be noted in Table 6, the optimal value of the objective function obtained from the first model provides a tighter upper

Table 2. Model Sizes and Computational Statistics for Example 1

model	objective function	constraints	variables		nodes	CPU time (s)
			binary	continuous		
simplified (SP)	50	1401	120	562	673	7.60
detailed (DP)	50	4807	306	70	350	5.46
DP without considering SP	50	4807	306	70	1513	24.97

dx.doi.org/10.1021/ie201757t | Ind. Eng. Chem. Res. 2012, 51, 3988−4001

Table 3. Final and Initial Processing Times of Model SP for All Slots Proposed on Different Units for Example 1

		slots, l							
		1	2	3	4	5	6	7	8
stages	units, k	$TI_{k1}-TF_{k1}$	$TI_{k2}-TF_{k2}$	$TI_{k3}-TF_{k3}$	$TI_{k4}-TF_{k4}$	$TI_{k5}-TF_{k5}$	$TI_{k6}-TF_{k6}$	$TI_{k7}-TF_{k7}$	$TI_{k8}-TF_{k8}$
1	1	0−0	0−0	**2−16**	**16−30**	30−30	30−30	**30−42**	**42−52**
	2	**0−10**	**10−22**	22−22	22−22	**22−34**	**34−50**	50−50	
2	3	0−0	**22−37**	37−37	37−37	37−37	37−37	37−37	**52−72**
	4	0−0	0−0	**16−41**	41−41	41−41	41−41	**42−57**	
	5	0−0	0−0	0−0	0−0	**34−49**	**50−68**		
	6	**10−30**	30−30	30−30	**30−55**	55−55			
3	7	**30−35**	**37−41**	**41−48**	48−48	**49−53**	53−53	**57−61**	**72−77**
	8	0−0	0−0	0−0	**55−62**	62−62	**68−73**	73−73	

Table 4. Final and Initial Processing Times of Model DP for All Slots Proposed on Different Units for Example 1

		slots: l						
		1	2	3	4	5	6	7
stages	units k	$TI_{k1}-TF_{k1}$	$TI_{k2}-TF_{k2}$	$TI_{k3}-TF_{k3}$	$TI_{k4}-TF_{k4}$	$TI_{k5}-TF_{k5}$	$TI_{k6}-TF_{k6}$	$TI_{k7}-TF_{k7}$
1	1	**2−16**	**16−30**	**30−42**	**42−52**	52−52	52−52	52−52
	2	**0−10**	**10−22**	**22−34**	**34−50**			
2	3	**22−37**	**52−72**	72−72	72−72	72−72		
	4	**16−41**	**42−57**	57−57				
	5	**34−49**	**50−68**					
	6	**10−30**	**30−55**					
3	7	**30−35**	**37−41**	**41−48**	**49−53**	**57−61**	**72−77**	77−77
	8	**55−62**	**68−73**	73−73	73−73			

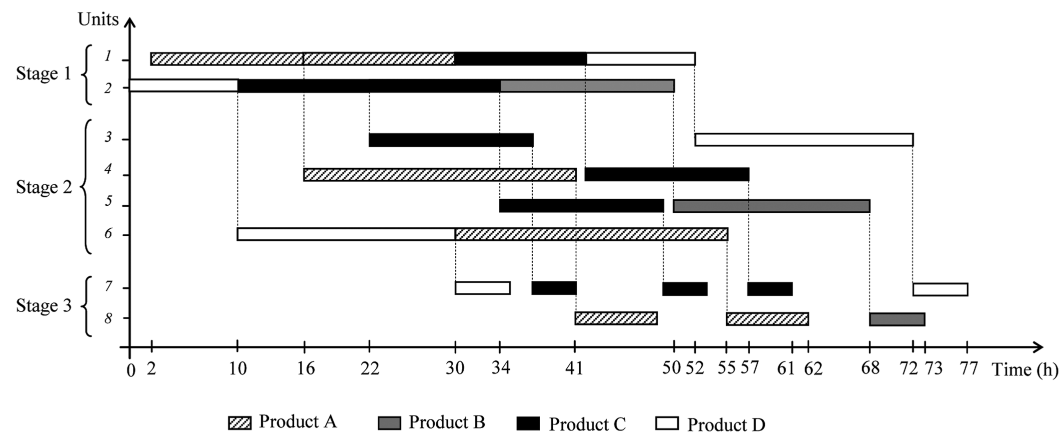

Figure 6. Optimal production schedule for the campaign of example 1.

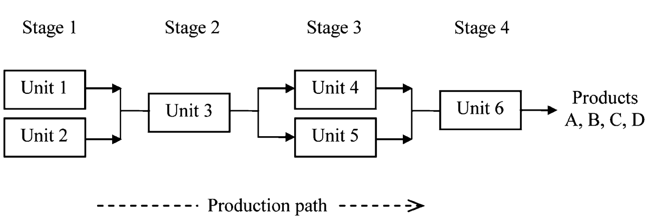

Figure 7. Plant structure for example 2.

bound on the objective function of the detailed model and permits solving it in a reasonable computing time. As well as the previous one, this example shows the significant advantage of the proposed solution strategy to achieve the exact scheduling of the batches that integrate the production campaign. The last row of Table 6 shows a considerable increase in the computation time when model DP is solved without considering the upper bound on the cycle time.

Table 5. Proccesing Times of Batches for Example 2

	processing time: t_{ij} (h)			
batches	stage 1	stage 2	stage 3	stage 4
i	k = 1, 2	k = 3	k = 4, 5	k = 6
1, 2 (A)	3	2	6	5
3, 4 (B)	6	2	2	2
5, 6 (C)	6	1	10	5
7 (D)	10	2	6	5

For each model, the optimal production sequence obtained for each batch unit at different stages is illustrated in Figure 8. Since stages 2 and 4 have only one unit, it is easy to compare the production sequence on them. In Figure 8b, at stage 2, the batch sequencing is D−C−B−A−B−A−C, while at stage 4 the batch sequencing is D−B−C−B−A−A−C. Therefore, the preordering constraint in the assignment of batches, introduced

Table 6. Model Sizes and Computational Statistics for Example 2

model	objective function	constraints	variables		nodes	CPU time (s)
			binary	continuous		
simplified (SP)	33	957	89	362	2390	5.54
detailed (DP)	29	3177	256	66	11114	45.01
DP without considering SP	29	3177	256	66	132217	660.56

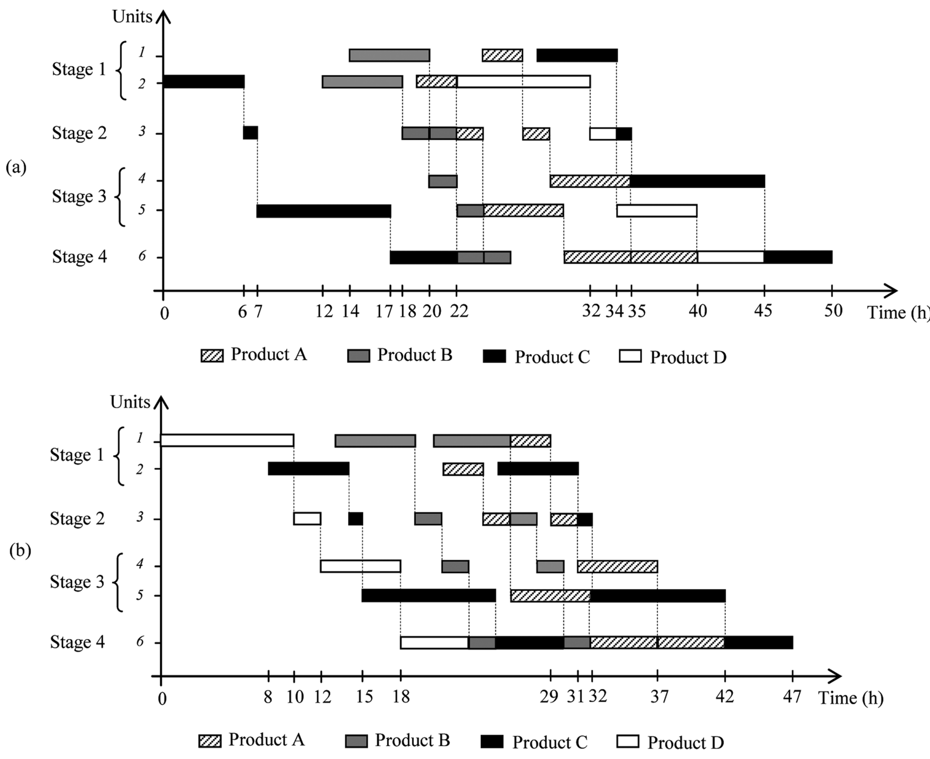

Figure 8. Gantt chart of the production campaign for example 2: (a) simplified model; (b) detailed model.

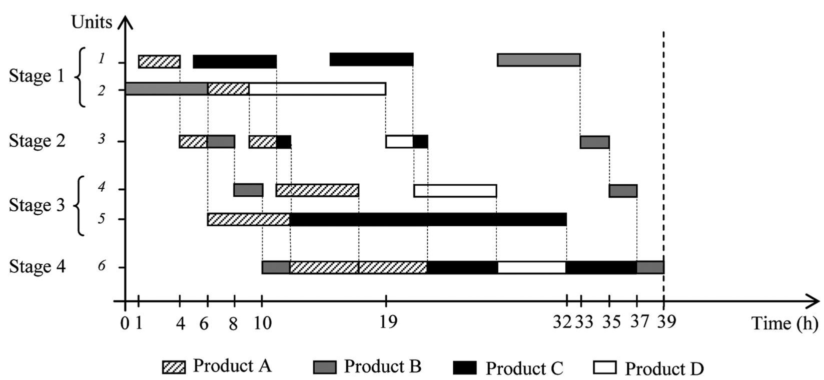

Figure 9. Gantt chart of the production campaign for example 2, considering makespan minimization.

in the simplified model to diminish the search space, is not valid for the optimal solution. Thus, the solution of model SP is suboptimal, but it introduces an appropriate upper bound to achieve a better performance, as it is shown in Table 6.

As previously mentioned, the proposed objective function is not the optimization criterion usually addressed in scheduling problems, namely, makespan minimization. It determines the minimum production time required to produce a given number of batches. Model DP has been adapted incorporating this new objective function. Also, constraints about cycle time have been removed. If this performance measure is utilized for this example with the adapted formulation, the optimal solution obtained for the detailed model corresponds to a makespan of 39 h as is shown in the Gantt chart in Figure 9.

From the solution shown in Figure 9, for each stage, the start and final processing times for the batches that compose the campaign are obtained. Consequently, the cycle time of the campaign is calculated as max{32, 19, 31, 29, 26, 29}; that is, CT = 32 h, that it is achieved in the first unit of the first stage.

Taking into account that operation mode using campaigns is appropriate to plan the production of plants operating under stable demand patterns over long time horizons, the selection of the cycle time minimization as objective function can be justified. To compare the performance of both objective

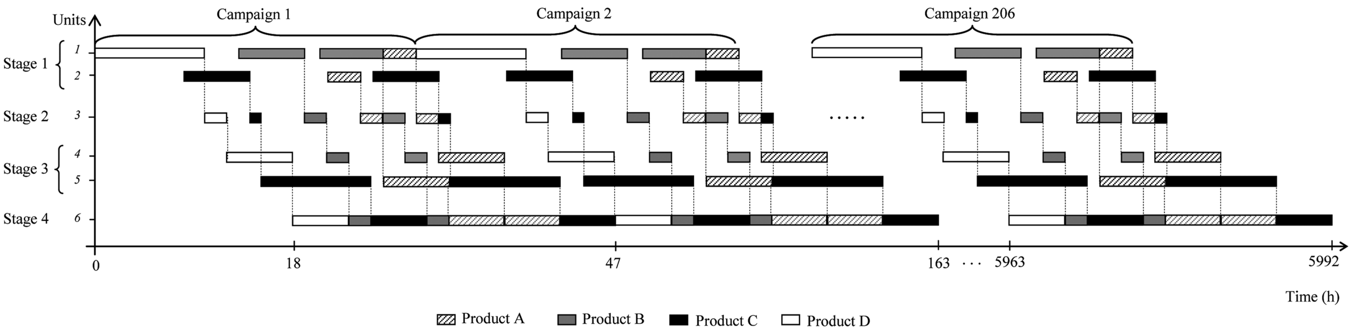

Figure 10. Optimal production schedule for the campaign of example 2 over a time horizon of 6000 h. Objective function: minimizing cycle time.

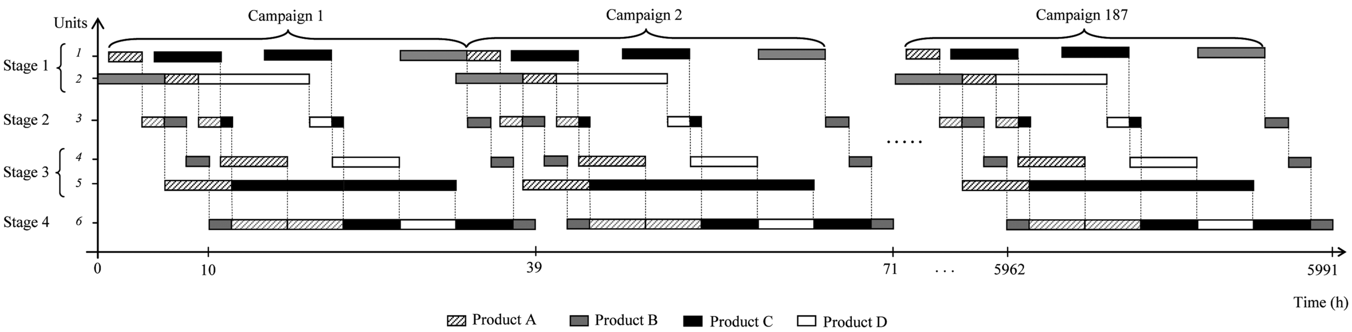

Figure 11. Optimal production schedule for the campaign of example 2 over a time horizon of 6000 h. Objective function: minimizing makespan.

Table 7. Processing Times t_{ij} (h) of Batches for Examples I9, I24, I26, and I29 from Liu and Karimi[17]

	I9		I24		I26		I29		
	stage 1	stage 2	stage 1	stage 2	stage 1	stage 2	stage 1	stage 2	stage 3
i	k = 1, 2	k = 3, 4	k = 1−3	k = 4, 5	k = 1−3	k = 4, 5	k = 1, 2	k = 3−7	k = 8−10
1	11	22	11	7	16	32	26	97	34
2	10	24	10	19	35	44	25	79	24
3	28	23	28	8	53	33	33	105	22
4	14	20	14	10	39	20	29	55	45
5	11	21	11	17	16	42	16	57	21
6	21	17	21	15	26	40	26	65	42
7	24	16	17	11	42	21	42	106	51
8			14	7	19	17			
9					27	37			

functions, it is assumed that the plant operates during a time horizon equal to 6000 h, where the same campaign is cyclically repeated over this horizon, with an overlapping operation mode. Figures 10 and 11 illustrate the detailed schedules for the campaigns obtained with both considered objective functions: cycle time minimization and makespan minimization, respectively.

For the first case, the campaign is repeated 206 times, whereas for the second, it is repeated 187 times. When the cycle time minimization is considered, the total idle time is 16.4% lower than that obtained for the makespan minimization case. In particular, the total idle times are reduced by approximately 59%, 6%, 10%, 44%, and 97% for units 1, 3, 4, 5, and 6, respectively, while for unit 2, the idle time is increased by 27%. Hence, in general, the equipment utilization is increased.

Moreover, taking into account that in the first case the maximum number of times that the campaign is repeated is higher than in the second case, the production is increased approximately 10% using the same equipment in the same time horizon. In conclusion, the overlapping between campaigns is tighter when the criterion of cycle time minimization is used.[8−11,25]

5.3. Solution Strategy Assessment. The previous examples have shown that there are cases where the optimal solutions of both models coincide. There are others where the solution of the simplified model is different but provides a good upper bound to solve the detailed scheduling model with reduced computational burden.

To assess the performance of the proposed strategy, new examples taken from Liu and Karimi[17] were used as test problems. Data of processing times of batches for each example are shown in Table 7.

Liu and Karimi[17] addressed short-term scheduling of multistage, multiproduct batch plants with ZW interstage policy, where the plant structure is fixed and the number of batches processed in the plant is known. They considered different scheduling objectives such as makespan, tardiness, earliness, and just-in-time. These authors solved the above examples through two types of MILP formulations where the optimization criterion for both cases is the makespan minimization. The first approach uses a continuous-time, slot-based representation (MZSL), while the second one uses a continuous-time, sequence-based representation (MZSQ2).

Table 8. Model Sizes and Computational Statistics for Examples Taken from Liu and Karimi[17]

	variables					
model	binary	continuous	constraints	no. examined nodes	objective function (h)	CPU time (s)
example I9						
MZSL	147	176	465	40661	85[a]	81.9
SP	75	236	591	2126	74	0.39
DP	144	38	1427	169	74	0.70
DP without considering SP	144	38	1427	4543	74	9.14
example I24						
MZSL	192	225	587	244841	58[a]	802
SP	100	362	865	1600	47	7.70
DP	198	46	2315	1657	47	9.04
DP without considering SP	198	46	2315	20754	47	59.71
example I26						
MZSL	243	280	733	39582	159[a]	234
SP	122	453	1062	38	143	0.68
DP	260	54	3505		143	0.62
DP without considering SP	260	54	3505	958	143	12.83
example I29						
MZSL	192	225	569	202448	179[a]	294
SP	105	506	1298	2113	134	21.18
DP	216	56	2554	536	134	4.53
DP without considering SP	216	56	2554	4790	134	50.09

[a]Objective function: Min Makespan.

The examples presented in Table 7 are solved using the solution strategy proposed in this paper, considering cycle time minimization as objective function, which present advantage when the plant operates in campaign-mode during long time horizons, as it was previously mentioned.

Taking into account the results of models SP and DP from Table 8, it can be noted that the optimal solutions for these models are the same for each example. Therefore, the simplified model SP can be considered a good heuristic for solving the scheduling problem of batches that compose a production campaign in a multistage, multiproduct batch plant.

Also, Table 8 lists model sizes and computational statistics for the detailed scheduling model DP when the optimal value of the objective function obtained from simplified model SP is not considered. As it can be noted, the proposed strategy shows a decrease in the total computational effort to achieve the exact scheduling of the batches, when it is compared with the solution of the model DP without considering the solution of model SP. Although the problem sizes are similar, the computational burden of model MZSL of Liu and Karimi[17] is significantly bigger than the proposed approach. This comparison is mainly based on the number of examined nodes, taking into account the different computers used.

6. CONCLUSIONS

In this work the scheduling of a multistage multiproduct batch plant with identical parallel units, operating in a campaign-mode, was tackled. Under stable conditions, demands can be accurately forecasted during a relatively long time horizon (e.g., several weeks or months). Then, more efficient production management can be accomplished if cyclic scheduling is used. In general, previous approaches focused on short-term scheduling and the formulations for plant operation using campaigns are few.

Two asynchronous slot-based continuous-time formulations that composed the proposed solution strategy were addressed: (i) a simplified model where preordering constraints were imposed in order to obtain an approximate solution of the scheduling problem in reasonable computing time, and (ii) a detailed model where preordering constraints are avoided and the optimal cycle time of the simplified model was used as upper bound of the cycle time of this model in order to reduce the search space. In addition, a novel expression was formulated for the number of proposed slots for each unit of each stage. Without loss of generality, this formula significantly reduces the number of postulated slots in each unit compared with formulations proposed in previous works.

The proposed models and the solution strategy were illustrated through three examples. In the first one, the optimal solution of the simplified model coincided with the optimal solution of the detailed model, and in many implemented instances, this was the case. Therefore, the simplified model represents a good heuristic for the scheduling problem of multistage multiproduct batch plants. In the second example, the optimal solution of both simplified and detailed model did not coincide. In this case, the upper bound for the cycle time imposed in the detailed model really affects the model performance, reducing significantly the computing time. When this bound is not used, the resolution time is increased by twelve times. From the computational point of view, the proposed solution strategy is very efficient in providing a general solution of the scheduling problem. Finally, the model was applied to other examples found in the literature.

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: +54 342 4534451. Fax: +54 342 4553439. E-mail: mmontagna@santafeconicet.gov.ar.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors want to thank the financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) under project PIP 1817 from Argentina.

■ NOMENCLATURE

Indices

i = batch.

j = stage.

k = unit.

k_{first}^{j} = first unit of stage j.

l = slot.

$\tilde{l}_k$ = first slot effectively used in unit k for processing one batch.

Sets

I = set of batches.

J = set of stages.

K = set of units.

K_j = identical batch units that operate in parallel and out-of-phase in stage j.

Parameters

L_{kj} = number of slots postulated for unit k of stage j.

M_n = parameter used for constraint of Big-M type, where n = 1, 2, 3, 4.

N_j = cardinality of K_j.

t_{ij} = processing time of batch i at stage j.

Binary Variables

X_{kl} = indicates if slot l of unit k is employed.

Y_{ikl} = indicates if batch i is assigned to slot l of unit k (defined for model DP).

Z_{il} = indicates if batch i is assigned to slot l.

Continuous Variables

CT = cycle time of the campaign.

TF_{kl} = final processing time of slot l in unit k.

TI_{kl} = initial processing time of slot l in unit k.

Y_{ikl} = continuous variable that assumes value 1 if batch i is assigned to slot l of unit k, and otherwise it take value 0 (defined for model SP).

■ REFERENCES

(1) Pinto, J. M.; Grossmann, I. E. Assignments and sequencing models of the scheduling of process systems. *Ann. Oper. Res.* **1998**, *81*, 433−466.

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

(3) Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Comput. Chem. Eng.* **2004**, *28*, 2109−2129.

(4) Floudas, C.; Lin, X. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Ann. Oper. Res.* **2005**, *139*, 131−162.

(5) 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* (6−7), 913−946.

(6) Pan, M.; Li, X.; Qian, Y. Continuous-time approaches for short-term scheduling of network batch processes: Small-scale and medium-scale problems. *Chem. Eng. Res. Des.* **2009**, *87*, 1037−1058.

(7) Mouret, S.; Grossmann, I. E.; Pestiaux, P. Time representations and mathematical models for process scheduling problems. *Comput. Chem. Eng.* **2011**, *35*, 1038−1063.

(8) Birewar, D.; Grossmann, I. E. Incorporating scheduling in the optimal design of multiproduct batch plants. *Comput. Chem. Eng.* **1989**, *13*, 141−161.

(9) Shah, N.; Pantelides, C.; Sargent, R. Optimal periodic scheduling of multipurpose batch plants. *Ann. Oper. Res.* **1993**, *42*, 193−228.

(10) Schilling, G.; Pantelides, C. Optimal periodic scheduling of multipurpose plants. *Comput. Chem. Eng.* **1999**, *23*, 635−655.

(11) Castro, P.; Barbosa-Póvoa, A.; Matos, H. Optimal periodic scheduling of batch plants using RTN-based discrete and continuous-time formulations: A case study approach. *Ind. Eng. Chem. Res.* **2003**, *42*, 3346−3360.

(12) Wu, D.; Ierapetritou, M. Cyclic short-term scheduling of multiproduct batch plants using continuous-time representation. *Comput. Chem. Eng.* **2004**, *28*, 2271−2286.

(13) Castro, P.; Novais, A. Optimal equipment allocation for high plant flexibility: An industrial case study. *Ind. Eng. Chem. Res.* **2008**, *47*, 2742−2761.

(14) Castro, P.; Novais, A. Short-term scheduling of multistage batch plants with unlimited intermediate storage. *Ind. Eng. Chem. Res.* **2008**, *47*, 6126−6139.

(15) Shaik, M. A.; Floudas, C. A. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. *Comput. Chem. Eng.* **2008**, *32*, 260−274.

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

(17) Liu, Y.; Karimi, I. A. Scheduling multistage batch plants with parallel units and no interstage storage. *Comput. Chem. Eng.* **2008**, *32*, 671−693.

(18) 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.

(19) Chen, C.; Liu, C.; Feng, X.; Shao, H. Optimal short-term scheduling of multiproduct single-stage batch plants with parallel lines. *Ind. Eng. Chem. Res.* **2002**, *41*, 1249−1260.

(20) Lim, M.; Karimi, I. A. Resource-constrained scheduling of parallel production lines using asynchronous slots. *Ind. Eng. Chem. Res.* **2003**, *42*, 6832−6842.

(21) Liu, Y.; Karimi, I. A. Novel continuous-time formulations for scheduling multi-stage batch plants with identical parallel units. *Comput. Chem. Eng.* **2007**, *31*, 1671−1693.

(22) Ierapetritou, M. G.; Floudas, C. A. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind. Eng. Chem. Res.* **1998**, *37*, 4341−4359.

(23) Li, J.; Floudas, C. A. Optimal event point determination for short-term scheduling of multipurpose batch plants via unit-specific event-based continuous-time approaches. *Ind. Eng. Chem. Res.* **2010**, *49*, 7446−7469.

(24) Brooke, A.; Kendrick, D.; Meeraus, A.; Raman, R. *GAMS, A User's Guide*; GAMS Development Corporation: Washington, DC, 2005.

(25) Trautmann, N.; Schwindt, C. A cyclic approach to large-scale short-term planning in chemical batch production. *J. Sched.* **2009**, *12*, 595−606.