AIChE Wiley InterScience DISCOVER SOMETHING GREAT

Simultaneous Batching and Scheduling of Single Stage Batch Plants with Parallel Units

Pedro M. Castro
Dept. de Modelação e Simulação de Processos, Instituto Nacional de Engenharia, Tecnologia e Inovação, 1649-038 Lisboa, Portugal

Muge Erdirik-Dogan and Ignacio E. Grossmann
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

DOI 10.1002/aic.11364
Published online November 21, 2007 in Wiley InterScience (www.interscience.wiley.com).

This article deals with the optimal short-term scheduling of single stage batch plants with sequence-dependent changeovers together with the optimal selection of the number of batches to produce. The novelty of the article is that instead of following the traditional approach of considering one processing task per batch, all batches of the product are now aggregated into a single task. Integer variables that hold the number of batches to produce are used to characterize these aggregated tasks. Two conceptually different continuous-time models are proposed. They rely on either multiple time grids or global precedence sequencing variables for event representation and generate a mixed integer linear program. The new formulations are compared to a traditional resource-task network multiple time grid approach as well as to a recent bounding model with immediate precedence sequencing variables. The results for several example problems show the new multiple time grid formulation as the best overall performer. When compared to the traditional approach, one order of magnitude savings in computational effort are achieved due to the need of fewer event points to get to the global optimal solutions.
 AIChE J, 54: 183–193, 2008
Keywords: optimization, mathematical modeling, mixed-integer programming

Introduction

Optimization models for batch scheduling can be classified based on four main aspects[1]: time representation, material balances, event representation, and objective function. Time representation can be considered the most important issue and optimization approaches can be classified into discrete and continuous- time formulations. The latter have received much of the attention of the process systems engineering community in the last decade, and a recent comparative study involving the five most important formulations can be found in Ref. 2.

In terms of material balances, the handling of batches and batch sizes gives rise to two types of model categories. Models based on unified frameworks for process representation like the state-task network (STN)[3] or the resource-task network (RTN)[4] can simultaneously deal with the optimal set of batches (number and size), the allocation and sequencing of manufacturing resources, and the timing of the processing tasks. Alternatively, there are models that assume that the number of batches of each size is known in advance, which can be regarded as one of the approaches for detailed production scheduling, widely used in industry, which decomposes the whole problem into two stages, batching and batch scheduling. Although they can address much larger practical

This article contains supplementary material available via the Internet at http://www.interscience.wiley.com/jpages/1547-5905/suppmat.

Correspondence concerning this article should be addressed to P. M. Castro at pedro.castro@ineti.pt.

problems, they are restricted to processes comprising sequential product recipes.

For event representation, models can rely on single[5–7] or multiple, unit-specific,[8,9] time grids with a prespecified number of event points or, alternatively, on immediate[10] or global precedence[11,12] relationships. In time grid-based models, the higher the number of tasks to execute, the larger the number of event points required to find global optimal solutions. Since most model entities, i.e. variables and constraints, feature one or more time indices, there is a clear incentive to develop models requiring fewer event points in the hope of making them computationally more efficient. A well known example results from shifting from single to multiple-time grid-based models.[2,8] Since time-grid-based models require an iterative search procedure over the number of event points composing the grid to find the global optimal solution, there is the additional disadvantage of solving a larger number of problems. On the other hand, models based on precedence relationships need to be solved only once.

Finally, the objective function can be one of different measures of the quality of the solution, where the criteria selected for the optimization usually has a direct effect on the computational performance. In addition, some objective functions can be very hard to implement for some event representations.

This article builds on recent work by the authors,[8] who have compared different event representation models for the scheduling of multistage batch plants with sequence-dependent changeovers. However, instead of considering that the number of batches for the production of a given product is known in advance, we solve the simultaneous batching and scheduling problem for single stage plants. While traditional STN/RTN approaches can implicitly determine the number of batches by the number of processing tasks that are executed, global optimal solutions may be impossible to reach due to the requirement of too many event points. Precedence-based models can tackle the problem if one defines one processing order for every possible batch of a product. However, in such a case, a large number of binary sequencing variables may be needed that might also compromise problem tractability. The approach that we propose in this article is significantly more efficient and takes advantage of the well-known ability of continuous-time formulations to handle variable duration tasks. Now, the number of batches of a product to produce is defined as explicit integer variables. We can then consider a single aggregated task to account for all the batches of a product that will need to be executed on a certain equipment unit. The total duration of an aggregated task will account for the time required to produce all selected batches of the product, plus the total changeover time between dissimilar batches of the same product, as well as the required changeover to the next product in the sequence.

The most significant development of this article concerns a new RTN-based multiple time grid continuous-time formulation. When weighed against a closely related, traditional RTN approach, fewer processing tasks will be generally needed and since each requires a single time interval, fewer event points will be needed to achieve global optimal solutions. We also widen the scope of the global precedence model of Harjunkoski and Grossmann[11] to consider a single order/activity per product regardless on the number of batches. A major adjustment makes the continuous variables for the orders ending times to be unit-dependent, to enable different batches of a product to be produced in multiple units and hence allow for more flexible production plans.

The two new models are compared to a multiple time grid, traditional RTN approach, and also to a very recent bounding model.[13] The latter relies on immediate precedence sequencing variables and does not use timing variables; so, it can be viewed as a less constrained version of a pure scheduling model. Computational studies are performed for a couple of objective functions, revenue maximization and makespan minimization, to establish the model's performance. In particular, their most serious limitations are identified and briefly explained. Note that when minimizing makespan, the batching problem reduces to finding out how to split the total number of batches over the parallel units in a flexible environment.

Motivation and Problem Definition

The problem that we will address in this article is inspired by a real world application of a specialty chemicals and plastic manufacturing business. This has become a highly competitive and an unpredictable industry due to the introduction of new products, migration of products from specialty grade to commodity, pressures to reduce costs and inventories. Therefore, assessing the accurate production capacity and increasing plant utilization can provide a competitive advantage.

In such industries, changeover times are sometimes required to switch the production from one product to another. If changeovers are sequence-dependent, then the utilization of plant capacity will depend on the sequence in which products are produced on the units. These changeover times can considerably reduce the capacity available for production particularly if their magnitudes are in the order of the batch times. Hence, there is a clear incentive to develop scheduling models that can account for sequence-dependent changeovers efficiently.

In this article, we consider the optimal short-term scheduling of single stage batch plants together with the selection of the optimal number of batches. Given are a set I of products to be produced in a set M of parallel identical/nonidentical batch equipment units. Both the duration $p_{i,m}$ and batch size $b_{i,m}$ of product i in unit m are known and assumed to be fixed. Given also are the duration of the required changeover times between the products, $cl_{i,i',m}$ and the product demand Δ_i. Two alternative objective functions will be considered: (a) the maximization of the sales revenue over a fixed time horizon, H, where the demand will typically not be met for all the products; (b) the minimization of the makespan required to meet the product demand. For the former objective, the products selling prices, v_i, are needed.

To allow for maximum plant flexibility, we do not restrict all batches of a given product to be produced in a single equipment unit. Thus, there will be cases where the optimal solution comprises the production of batches in parallel units (see Figure 1). Note in Figure 1 that nonzero changeovers between different batches of the same product (e.g. I1-I1 in M1, I3-I3 and I4-I4 in M2), i.e. $cl_{i,i,m} \neq 0$, can be handled, although it must be said that this type of changeovers is less frequent than those involving different products.

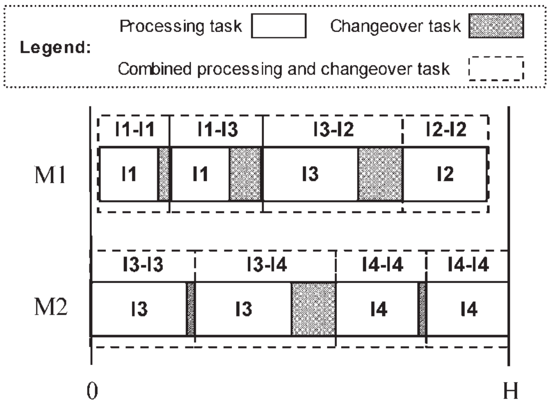

Figure 1. Implicit batching approach.

Required number of batches for a particular product determined by the number of instances executed of the corresponding processing task (e.g. 2 batches for product I1, 1 for I2, 3 for I3 and 2 for I4).

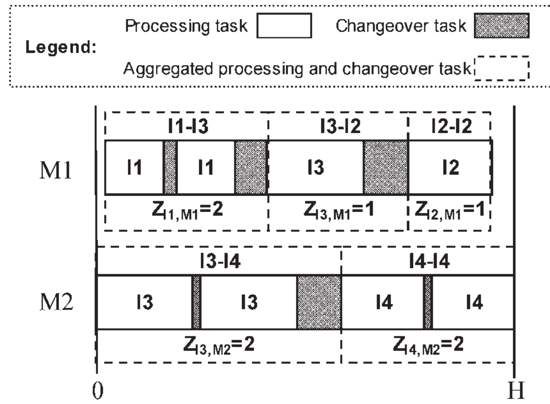

Figure 2. Explicit batching approach.

Required number of batches for a particular product is defined as an integer model variable that affects the duration of the corresponding aggregated task.

Multiple Time Grid Continuous Models

The first model category in terms of event representation uses multiple, unit specific, time grids. We use formulation CT4I[8] as a representative of traditional[1] STN/RTN scheduling formulations and propose a new Resource-Task Network-based[4] batching and scheduling approach. The two formulations differ conceptually on the definition of a processing task. The traditional way is to define a processing task as the activity to process one batch of a particular product. If the model decides that more than a single batch of a product is required, it will execute a few instances of the corresponding task in the given time horizon, with the number of batches being equal to the number of instances of the task that are carried out. This will be called the implicit batching approach and referred to as CT-IB (see Figure 1).

Time grid-based continuous-time formulations[5–7,9] can handle variable duration tasks without significantly altering the complexity of the model. In the new approach, we take advantage of this fact and consider all processing instances of the same product that are executed in the same unit as a single aggregated task. The number of batches to produce on a particular unit will be defined as integer model variables, $Z_{i,m}$, and will affect the duration of the aggregated task. As a consequence, this is named the explicit batching approach, referred to as CT-EB. In CT-EB, the aggregated task linked to product i accounts for the aggregated processing time, the changeover time between different batches of the same product (as many times as the number of batches allocated to the unit minus one) and the final changeover time (if required) that prepares the unit for the subsequent production of another product. These features are illustrated in Figure 2.

The underlying time grid is given in Figure 3. All $|M|$ time grids feature the same number of event points, defined in set T. They also share the same origin and the fact that H acts as an upper bound on the time span. However, all other event points are free to vary between those two limits and there is no relation whatsoever between the timing variables of event points belonging to different grids. An important property inherited from CT4I[8] is that a single time interval (slot) is enough for the execution of any task. Thus, the higher the number of tasks to execute, the larger the number of event points required to find the global optimal solution, which like in all other time grid based continuous-time models needs to be iteratively estimated.[1,2] With that in mind, from a comparison between Figures 1 and 2, one can see that CT-IB requires a total of five event points, while CT-EB requires just four. Since the number of event points required to solve a problem to global optimality is being used[2] as a performance metric for continuous-time formulations, such a small example illustrates the motivation for the development of CT-EB. Overall, CT-EB can be seen as a major upgrade from CT-IB concerning efficiency, but not generality since it cannot handle certain features that can appear in practice but are not part of the problem under consideration as will be discussed later on.

A few variables are common to both formulations. The four-index binary extent variables $N_{i,i',m,t}$ identify the execution of the combined/aggregated processing and changeover task. More specifically, they locate the starting event point (t) of the processing task of product i in unit m followed by the changeover task that makes the unit ready to process product i' immediately after. Binary variables $C^0_{i,m}$ account for the initial condition of unit m, which can be fixed if the initial state is known. The excess resource variables $C_{i,m,t}$ are of the continuous type and, like the following, are all nonnegative. They indicate the availability of unit m at a condition that enables it to process product i at event point t. These variables could also be defined as binary variables, but it is not necessary since the model constraints ensure that $C_{i,m,t} = \{0,1\}$. It is important to note that the sum over i is linked to the availability of the unit for which variables $R_{m,t}$ could be used explicitly.[8] Finally, $T_{t,m}$ represents the absolute

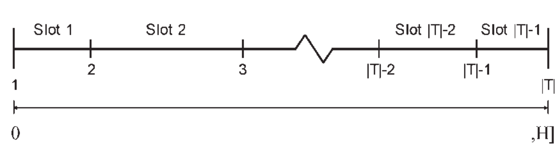

Figure 3. Basic continuous-time grid (one for each equipment unit).

time of event point t belonging to time grid m and MS is the makespan.

Review of traditional, implicit batching approach (CT-IB)

The model constraints of the implicit batching continuous-time formulation (CT-IB) are given below. Other than makespan minimization, we consider the objective of the maximization of the sales revenue, Eq. 1, where the total production of product i is achieved by multiplying the batch size by the number of batches, which in turn is equal to the number of tasks that are executed. In Eq. 1, the domain of the binary variables is given by set $I_{i',m,t}$, defined through Eq. 2. Note that in the last time interval (tasks starting at $t = |T| - 1$, see Figure 3) only tasks with the same product index can be performed, since there are no slots left to process any more tasks (see also Figure 1). Set I_m includes the orders that can be processed in unit m, those that have a nonzero duration, see Eq. 3.

$$\max \sum_{t \in T} \sum_{m \in M} \sum_{i' \in I} \sum_{i \in I_{i',m,t}} v_i \cdot b_{i,m} \cdot N_{i,i',m,t} \quad (1)$$

$$I_{i',m,t} = \{ i \in I_m : t \neq |T| - 1 \vee i = i' \} \quad \forall m \in M, i' \in I_m, t \in T, t \neq |T| \quad (2)$$

$$I_m = \{ i \in I : p_{i,m} > 0 \} \quad \forall m \in M \quad (3)$$

Equation 4 is the excess resource balance. Equation 5 ensures that there is but one initial condition for each equipment unit. There is exactly one equipment unit of type m; so the maximum availability at any point in time is equal to one, Eq. 6. The timing constraints relate the duration of a particular time interval to the duration of the combined task taking place, which is calculated by multiplying the corresponding binary extent variable by the sum of the processing plus changeover time, see Eq. 7. Note that we need to remove the changeover time of tasks executed in the last time interval to ensure that whenever there are products with $cl_{i,i,m} \neq 0$, we end with a processing task, which is reasonable since we do not know what lies ahead (see Figure 1). Because of this, tasks will tend to be executed from the last to the first time interval, similarly to CT3I[8] and contrary to CT4I[8]. Whenever the objective is makespan minimization, Eq. 8 ensures that the makespan (MS) is greater than the ending time of all tasks. According to Figure 3, the time of the first event point must be set to zero (Eq. 9), while the time horizon acts as the upper bound for all points, see Eq. 10.

$$C_{i,m,t} = C^0_{i,m}\Big|_{t=1} + C_{i,m,t-1}\Big|_{t \neq 1} + \sum_{i' \in I_{i,m,t-1}} N_{i',i,m,t-1} - \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} \quad \forall m \in M, i \in I_m, t \in T \quad (4)$$

$$\sum_{i \in I_m} C^0_{i,m} = 1 \quad \forall m \in M \quad (5)$$

$$C_{i,m,t} \leq 1 \quad \forall m \in M, i \in I_m, t \in T \quad (6)$$

$$T_{t+1,m} - T_{t,m} \geq \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} \left[N_{i,i',m,t} \cdot \left(p_{i,m} + cl_{i,i',m}\Big|_{t \neq |T|-1} \right) \right] \quad \forall m \in M, t \in T, t \neq |T| \quad (7)$$

$$\mathrm{MS} \geq T_{t,m} + \sum_{\substack{t' \in T \\ t' \geq t \\ t' \neq |T|}} \sum_{i' \in I_m} \sum_{i \in I_{i',m,t'}} \left[N_{i,i',m,t'} \cdot \left(p_{i,m} + cl_{i,i',m}\Big|_{t \neq |T|-1} \right) \right] \quad \forall m \in M, t \in T, t \neq |T| \quad (8)$$

$$T_{1,m} = 0 \quad \forall m \in M \quad (9)$$

$$T_{t,m} \leq H \quad \forall m \in M, t \in T \quad (10)$$

The demand constraints, which are objective function-dependent, complete the model. For profit maximization, the total amount produced of i must not exceed its demand (Eq. 11), while for makespan minimization we want to meet the exact demand for all products (Eq. 12). Naturally, the latter assumes that the given demands are multiples of the batch sizes otherwise $\geq$ is used instead.

$$\sum_{m \in M} \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} b_{i,m} \cdot N_{i,i',m,t} \leq \Delta_i \quad \forall i \in I \quad (11)$$

$$\sum_{m \in M} \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} b_{i,m} \cdot N_{i,i',m,t} = \Delta_i \quad \forall i \in I \quad (12)$$

Remarks. Although CT-IB considers tasks with known batch sizes and fixed duration, it is straightforward[5] to adapt it to consider tasks with variable batch sizes and durations that are dependent on the amount of material that is processed (if the relation is made linear the model remains a MILP). CT-IB will then allow different processing instances of the same product to handle different batch sizes, which will increase the variety of production amounts that can be obtained, and use more efficiently the available time horizon. CT-IB can thus be considered a more flexible and general formulation than CT-EB.

Concerning the definition of other objective functions, it is important to emphasize that the binary variables $N_{i,i',m,t}$ can be used to account for other nontime-related contributions. For example, fixed and variable (dependent on the amount processed) operating costs can be merged into a single parameter, since the batch sizes are fixed, and eventually further incorporated into parameter v_i. Changeover costs are also straightforward to implement, since the summation of such binary variables over t gives the number of changeovers from i to i' in unit m.

New Explicit Batching Approach (CT-EB)

The explicit batching continuous-time formulation (CT-EB) requires an additional set of variables to characterize an aggregated task. Continuous extent variables $\xi_{i,m,t}$ give the amount of product i produced in unit m at time interval t. In addition, CT-EB uses the integer variables $Z_{i,m}$ to determine the number of batches of i produced in unit m, as already mentioned. CT-EB shares Eqs. 4–6, 9, and 10 with CT-IB and features constraints that are very similar to some used by the latter. We will be highlighting only the most important differences.

The first thing to note is that the domain of variables $N_{i,i',m,t}$ has changed. Like before, tasks with different order indices cannot be executed in the last time slot. However, now at most one aggregated task of product i will be executed in unit m, since one task can handle multiple batches (see Eq. 13). Therefore, tasks with the same order index can be restricted to the last time slot. Set $I_{i',m,t}$ is thus given by Eq. 14.

$$\sum_{t\in T}\sum_{\substack{i'\in I_m\\ i\in I_{i',m,t}}} N_{i,i',m,t} \leq 1 \quad \forall m\in M, i\in I_m \tag{13}$$

$$I_{i',m,t} = \{i\in I_m : (t\neq |T|-1 \wedge i\neq i') \vee (t=|T|-1 \wedge i=i')\} \quad \forall m\in M, i'\in I_m, t\in T, t\neq |T| \tag{14}$$

The maximization of sales revenue objective function is given by Eq. 15, where the total production of product i is now accounted for through the continuous extent variables. These must equal the batch size times the number of batches of the product on that unit, Eq. 16. Equation 17 places an upper bound ($\overline{Z}_{i,m}$) on the integer variables. The floor function has been used due to the fact that the total production of i cannot exceed its demand (Eq. 18) for the sales revenue objective. For makespan minimization, the equality is used instead (Eq. 19).

$$\max \sum_{\substack{t\in T\\ t\neq |T|}}\sum_{m\in M}\sum_{i\in I_m} v_i\cdot \xi_{i,m,t} \tag{15}$$

$$\sum_{\substack{t\in T\\ t\neq |T|}} \xi_{i,m,t} = b_{i,m}\cdot Z_{i,m} \quad \forall m\in M, i\in I_m \tag{16}$$

$$\overline{Z}_{i,m} = \lfloor \Delta_i / b_{i,m} \rfloor \quad \forall m\in M, i\in I_m \tag{17}$$

$$\sum_{\substack{t\in T\\ t\neq |T|}}\sum_{\substack{m\in M\\ i\in I_m}} \xi_{i,m,t} \leq \Delta_i \quad \forall i\in I \tag{18}$$

$$\sum_{\substack{t\in T\\ t\neq |T|}}\sum_{\substack{m\in M\\ i\in I_m}} \xi_{i,m,t} = \Delta_i \quad \forall i\in I \tag{19}$$

Material i can only be produced if one of its aggregated tasks is active at that point. The binary and continuous extent variables are related through Eqs. 20 and 21, where the upper bound on the amount produced is the product demand, and the lower bound is the batch size.

$$\xi_{i,m,t} \leq \Delta_i \cdot \sum_{\substack{i'\in I_m\\ i\in I_{i',m,t}}} N_{i,i',m,t} \quad \forall m\in M, i\in I_m, t\in T, t\neq |T| \tag{20}$$

$$\xi_{i,m,t} \geq b_{i,m} \cdot \sum_{\substack{i'\in I_m\\ i\in I_{i',m,t}}} N_{i,i',m,t} \quad \forall m\in M, i\in I_m, t\in T, t\neq |T| \tag{21}$$

The timing constraints are more complex than before. For each aggregated task, we need to account for the total processing task, total changeover time between different batches of the same product and changeover time for the following product; the latter is only for tasks not executed in the last time interval. Figure 4 illustrates how the procedure works for a simple example and relates the multiplying terms to the relevant model variables. The general constraint is given in

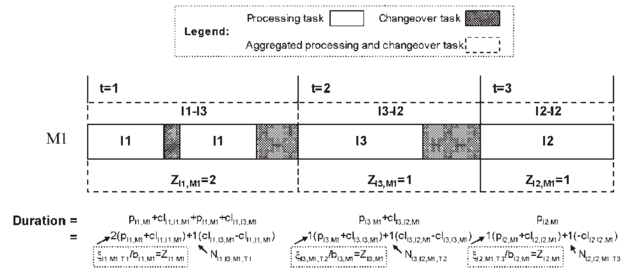

Figure 4. Calculation of the duration of aggregated tasks for explicit batching approach.

Eq. 22. Finally, Eq. 23 is the equivalent of Eq. 8 and is only to be used for makespan minimization.

$$T_{t+1,m} - T_{t,m} \geq \sum_{i\in I_m}\left[\frac{\xi_{i,m,t}}{b_{i,m}}\cdot (p_{i,m}+cl_{i,i,m}) + \sum_{\substack{i'\in I_m\\ i\in I_{i',m,t}}} N_{i,i',m,t}\cdot \left(cl_{i,i',m}\big|_{t\neq |T|-1} - cl_{i,i,m}\right)\right] \quad \forall m\in M, t\in T, t\neq |T| \tag{22}$$

$$\mathrm{MS} \geq T_{t,m} + \sum_{\substack{t'\in T\\ t'\geq t\\ t'\neq |T|}}\sum_{i\in I_m}\left[\frac{\xi_{i,m,t'}}{b_{i,m}}\cdot (p_{i,m}+cl_{i,i,m}) + \sum_{\substack{i'\in I_m\\ i\in I_{i',m,t'}}} N_{i,i',m,t'}\cdot \left(cl_{i,i',m}\big|_{t\neq |T|-1} - cl_{i,i,m}\right)\right] \quad \forall m\in M, t\in [1,|T|] \tag{23}$$

Models Based on Precedence Relationships

The second model type uses precedence relationships instead of an explicit time grid to relate processing tasks of different products. The major advantage is that the quality of the solution can no longer be compromised due to an a priori specification concerning the maximum number of events that can take place. In this section, a new general precedence model is proposed and an algorithm[13] relying on an immediate precedence model is reviewed.

New Model with Global Precedence Sequencing Variables (CT-SV)

The new batching and scheduling continuous-time formulation (CT-SV) uses global precedence sequencing variables and needs to be solved only once to find the global optimal solution to the problem. It is related to the one by Harjunkoski and Grossmann[11] but now multiple batches for each product are considered instead of a single one. One sequencing variable is employed for each pair of products ($X_{i,i'}$) to identify if product i ends before i' (with $i' > i$). Variables $Y_{i,m}$, also binary, are used to assign order i to unit m. Like for CT-EB, integer variables $Z_{i,m}$ give the number of batches of i processed in unit m. The full processing length of prod-

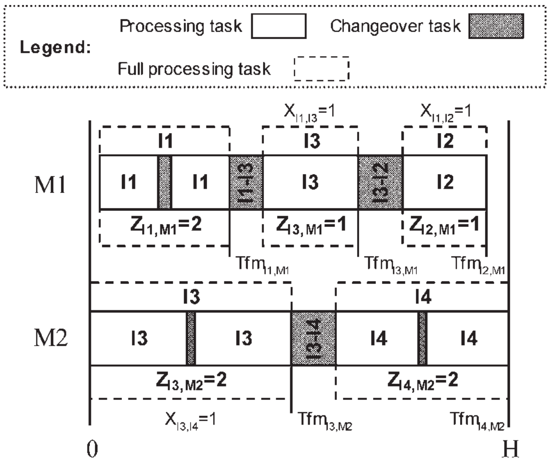

Figure 5. Illustration of global precedence sequencing variables approach showing the values of the most important model variables for this simple example.

uct i will include the total time of the individual processing tasks plus the total changeover time between different batches of i, as is illustrated in Figure 5. The other novel feature is that the ending time of the tasks are now unit-dependent (given by continuous variables $Tfm_{i,m}$), since a particular product may be produced in multiple units.

Concerning the model constraints, Eq. 24 represents the maximization of the revenue from product sales. Equation 25 ensures that the ending time of full processing task i in unit m is greater than the total duration of its contributors. Notice that the first term on the right-hand side accounts for one more changeover (i,i) than it requires and that is the reason why that same time is subtracted in the second term. Equations 26 and 27 are big-M constraints (where the big-M parameter is the time horizon, H) relating the ending times of any two products. The time horizon acts as an upper bound on the timing variables, Eq. 28. Equation 29 ensures that the total duration of the full processing tasks is lower than the time horizon. Note that due to the use of global instead of immediate precedence sequencing variables, changeovers between batches of different products cannot be added to the LHS; so the constraints are not as tight as those given by Eq. 41. Equation 30 states that batches of i can only be allocated to unit m if the product is allocated to that unit. The ceiling instead of the floor function has been used to determine the upper bound, since it might not be possible to meet the exact demand when minimizing the makespan. The demand constraint is given in Eq. 31.

$$\max \sum_{m \in M} \sum_{i \in I_m} v_i \cdot b_{i,m} \cdot Z_{i,m} \tag{24}$$

$$Tfm_{i,m} \geq Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m} \quad \forall m \in M, i \in I_m \tag{25}$$

$$\begin{aligned} Tfm_{i',m} \geq\ & Tfm_{i,m} + Z_{i',m} \cdot (p_{i',m} + cl_{i',i',m}) - cl_{i',i',m} \cdot Y_{i',m} \\ & + cl_{i,i',m} \cdot X_{i,i'} - H \cdot (3 - X_{i,i'} - Y_{i,m} - Y_{i',m}) \\ & \forall m \in M, i, i' \in I_m, i' > i \end{aligned} \tag{26}$$

$$\begin{aligned} Tfm_{i,m} \geq\ & Tfm_{i',m} + Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m} \\ & + cl_{i',i,m} \cdot (1 - X_{i,i'}) - H \cdot (2 + X_{i,i'} - Y_{i,m} - Y_{i',m}) \\ & \forall m \in M, i, i' \in I_m, i' > i \end{aligned} \tag{27}$$

$$Tfm_{i,m} \leq H \quad \forall m \in M, i \in I_m \tag{28}$$

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] \leq H \quad \forall m \in M \tag{29}$$

$$Z_{i,m} \leq \lceil \Delta_i / b_{i,m} \rceil \cdot Y_{i,m} \quad \forall m \in M, i \in I_m \tag{30}$$

$$\sum_{\substack{m \in M \\ i \in I_m}} b_{i,m} \cdot Z_{i,m} \leq \Delta_i \quad \forall m \in M, i \in I \tag{31}$$

When minimizing makespan, besides turning Eq. 31 into an equality or the opposite inequality, and replacing Eq. 29 with the tighter Eq. 32, one additional set of constraints is required. Equation 33 ensures that the makespan, MS, is greater than the ending time of all tasks.

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] \leq \text{MS} \quad \forall m \in M \tag{32}$$

$$\text{MS} \geq Tfm_{i,m} \quad \forall m \in M, i \in I_m \tag{33}$$

Remarks. Model CT-SV implicitly assumes the same global precedence between any two products in all equipment units. However, it is theoretically possible that if two products are processed in more than a single unit, the optimal sequence may change due to different values of $cl_{i,i',m}$ and $cl_{i,i',m'}$ or simply because of the influence of the other products that are processed in those units (see section CT-SV for an example). Cutting off the true optimal solution can be overcome by adding index m to the sequencing variables $X_{i,i'}$, but computational studies have shown a steep decrease in the model's performance.

Contrary to the time grid formulations, the use of global precedence sequencing variables prevents us from explicitly knowing the required changeover tasks between batches of adjacent products. As a consequence, accounting for transition costs is not possible.

Review of Algorithm of Erdirik-Dogan and Grossmann[13] with Immediate Precedence Sequencing Variables (E-D&G)

Erdirik-Dogan and Grossmann[13] have recently proposed a method for simultaneously determining the number of batches to produce of each product together with their allocation and sequencing on the equipment units. The idea for the sequencing is to generate a cyclic schedule that minimizes the changeover times amongst the assigned products while determining at the same time the optimal sequence by breaking one of the links in the cycle.[14] Immediate precedence sequencing variables are used and the sequencing constraints can be regarded as a relaxation of the traveling salesman problem.[15]

The E-D&G model has the potential drawback of generating solutions featuring subcycles even though for asymmetric sequence-dependent changeovers the likelihood of such is

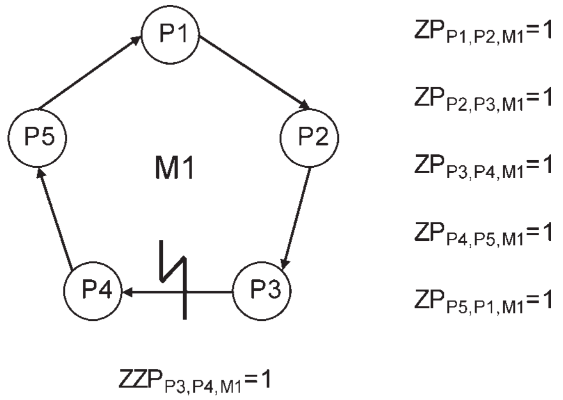

Figure 6. Example of a cyclic schedule and location of the link to be broken.

very small. Whenever subcycles are present, the solution does not correspond to a feasible schedule. Nevertheless, since a less constrained version of the scheduling problem is being solved, the model will yield an upper/lower bound when maximizing revenue/minimizing makespan, which is very tight, as will be seen in the computational studies section. If no subcycles are present, the solution is a global optimal schedule.

Subcycles can be broken by adding subtour elimination constraints and solving the model iteratively until a feasible schedule is found. Rather than performing a branch and bound search with the subtour elimination constraints,[16] a heuristic alternative is to successively add these constraints for the set of products involved in the various subcycles. In this case, at the end of the iterative procedure, the solution will correspond to a lower/upper bound on the revenue/makespan.

When compared to Erdirik-Dogan and Grossmann,[13] the nomenclature of the model has been adapted to the specifics of the problem under consideration. Like in CT-SV, $Y_{i,m}$ are the binary assignment variables and $Z_{i,m}$ are the integer batching variables. Also of the binary type, $ZP_{i,i',m}$ are the immediate precedence sequencing variables indicating that product i precedes i' in unit m, while $ZZP_{i,i',m}$ identifies if the link between i and i' in unit m is to be broken (see Figure 6). The E-D&G model constraints are given next together with a brief explanation.

$$Y_{i,m} = \sum_{i' \in I_m} ZP_{i,i',m} \quad \forall m \in M, i \in I_m \tag{34}$$

$$Y_{i,m} = \sum_{i' \in I_m} ZP_{i',i,m} \quad \forall m \in M, i \in I_m \tag{35}$$

$$\sum_{i \in I_m} \sum_{i' \in I_m} ZZP_{i,i',m} = 1 \quad \forall m \in M \tag{36}$$

$$ZZP_{i,i',m} \leq ZP_{i,i',m} \quad \forall m \in M, i \in I_m, i' \in I_m \tag{37}$$

$$Y_{i,m} \geq ZP_{i,i,m} \quad \forall m \in M, i \in I_m \tag{38}$$

$$ZP_{i,i,m} + Y_{i',m} \leq 1 \quad \forall m \in M, i \in I_m, i' \in I_m, i \neq i' \tag{39}$$

$$ZP_{i,i,m} \geq Y_{i,m} - \sum_{\substack{i' \in I_m \\ i' \neq i}} Y_{i',m} \quad \forall m \in M, i \in I_m \tag{40}$$

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] + \sum_{i \in I_m} \sum_{i' \in I_m} [cl_{i,i',m} \times (ZP_{i,i',m} - ZZP_{i,i',m})] \leq H \quad \forall m \in M \tag{41}$$

$$Z_{i,m} \geq Y_{i,m} \quad \forall m \in M, i \in I_m \tag{42}$$

Equations 34 and 35 ensure that if product i is assigned to unit m there is exactly one transition to/from product i' in that unit. Equation 36 states that exactly one of the links in the optimal cycle can be broken. A link cannot be broken if the corresponding pair is not selected in the cycle, Eq. 37. To avoid schedules consisting of self loops, a self changeover is allowed if and only if that product is the only one assigned to the unit (Eqs. 38–40). Equation 41 ensures that the total processing plus changeover time in unit m does not exceed the time horizon. It is a much tighter constraint than Eq. 29 due to the use of immediate instead of global precedence sequencing variables. Equation 42 guarantees that only in cases where product i is assigned to unit m, can there be batches of i produced in m. The remaining sets of constraints, Eqs. 30 and 31 together with the revenue maximization objective function, Eq. 24, are shared with CT-SV.

When minimizing makespan and besides the previously mentioned changes to Eq. 31, one just needs to replace Eq. 41 by Eq. 43.

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] + \sum_{i \in I_m} \sum_{i' \in I_m} cl_{i,i',m} \cdot (ZP_{i,i',m} - ZZP_{i,i',m}) \leq \text{MS} \ \forall m \in M \tag{43}$$

Whenever the optimal solution features subtours, the model needs to be resolved with the subtour elimination constraints (Eq. 44). In this iterative procedure, set IT includes all previous iterations, S_{it}, the set of subtours of iteration it, and $IS_{s,it,m}$ the set of products involved in subtour s of iteration it in machine m. Equation 44 is applied for every pair of subtours belonging to the set of active systems, $AcS_{it,m}$, which is defined by Eq. 45. The elements of sets $IS_{s,it,m}$ and $AcS_{it,m}$ are given in E-D&G section for one example.

$$\sum_{i \in IS_{s,it,m}} \sum_{i' \in IS_{s',it,m}} ZP_{i,i',m} \geq 1 \quad \forall it \in IT, m \in M, (s,s') \in AcS_{it,m} \tag{44}$$

$$AcS_{it,m} = \{(s,s')s, s' \in S_{it}, s < s' : \sum_{i \in IS_{s,it,m}} 1 > 1 \wedge \sum_{i \in IS_{s',it,m}} 1 > 1\} \quad \forall it \in IT, m \in M \tag{45}$$

Note that with immediate precedence sequencing variables, it is straightforward to account for transition costs in the objective function.

Table 1. Overview of Computational Performance (CPUs) for Maximum Plant Flexibility and Revenue Maximization[a]

			Model			
Problem	H (h)	Optimum ($)	CT-IB	CT-EB	CT-SV	E-D&G
P1 (|I| = 5, |M| = 2)	120	908.7	2.95	1.19	21.9	**0.42**
	144	1036.5	6.17	0.84	7.03	**0.36**
	168	1149.3	4.72	0.55	2.8	**0.42**
P2 (|I| = 8, |M| = 2)	144	1202.8	18.2	1.5	40859[f]	**1.12**
	168	1329.2	465	1.73	11705[g]	**1.09**
	192	1464.4	48.5	**0.52**	38209[h]	0.75
P3 (|I| = 10, |M| = 2)	120	1218.5	723	**285**	11842[i]	5.23[n]
	144	1388.88	853	**49.1**	13850[j]	1.89[o]
	168	1544.88	14386	**0.19**	32.8	1.67[p]
P4 (|I| = 10, |M| = 4)	96	1742.14	3600[c]	2292	9286[k]	**33.6**
	120	2068.58[b]	3600[d]	**14669**	9990[l]	470[q]
	144	2196.68	3600[e]	**79.4**	3600[m]	246[r]

[a]FTP = fewer event points were used (|T| value within brackets) than those required to find the optimal solution. BPS = best possible solution at the time of termination. LB = first solution features subtours so it provides a lower bound on the optimal solution (value given). MRL = maximum resource limit exceeded. NIT = number of iterations. OM = solver ran out of memory. SO = suboptimal solution returned. UB = first solution features subtours so it provides an upper bound on the optimal solution (value given).
[b]May not be the global optimal solution.
[c]MRL, SO = 1751.3, FTP (|T| = 7).
[d]MRL, SO = 2028.0, FTP (|T| = 8).
[e]MRL, SO = 2183.1, FTP (|T| = 8).
[f]SO = 1196.8, although solver solved to optimality (special case).
[g]OM, SO = 1313.8, BPS = 1375.4.
[h]OM, BPS = 1505.5.
[i]MRL, SO = 1212.4, BPS = 1336.9.
[j]OM, SO = 1336.9, BPS = 1492.9.
[k]OM, SO = 1715.6, BPS = 1820.3.
[l]OM, SO = 2037.7, BPS = 2165.3.
[m]MRL, SO = 2148.98, BPS = 2196.68.
[n]UB = 1264.9, SO = 1029.52, NIT = 5.
[o]UB = 1429.28, SO = 1261.68, NIT = 3.
[p]UB = 1544.88, SO = 1492.88, NIT = 3.
[q]UB = 2085.38, SO = 2032.80, NIT = 4.
[r]NIT = 5.

Computational Studies

The performance of the four alternative approaches is evaluated in this section through the solution of four example problems ranging from 5 products in 2 units to 10 products in 4 units. For revenue maximization, each problem is solved for three different values of the time horizon, H. Naturally, the higher the time span the higher the revenue, due to the production of an higher number of batches. For makespan minimization with the continuous-time formulations, we need to specify a value of H that is sufficiently large to ensure full production. We have used 1.5 times the highest value of the maximum revenue tests. Overall, a total of 128 computer runs were made. The data from the example problems are adapted from data taken from a real industrial plant and are given in Supplementary Material. It is important to emphasize that, despite the fact that all the examples feature $cl_{i,i,m}=0\ \forall m \in M, i \in I_m$, we have confirmed with other examples that all the approaches are able to handle $cl_{i,i,m} \neq 0$.

The continuous-time mathematical formulations were implemented and solved in GAMS 22.2 using CPLEX 10.0.1 as the MILP solver. All the problems were solved to optimality (relative tolerance equal to 1E-6) or up to a maximum resource limit of typically 1 h, or up to the moment the solver ran out of memory. The computer used was a Pentium-4 3.4 GHz processor with 2 GB of RAM, running Windows XP Professional.

Tables showing the computational effort for the several formulations identify the best performer in bold and give the value of the global optimal solution or in alternative the best known solution. Since the global precedence sequencing variables-based approach is not entirely general, for reasons explained in section CT-SV, we rely on CT-EB or E-D&G to identify the global optimum. With CT-EB, this is assumed to be found whenever no improvement in the optimal value is observed following a single increment in the number of event points |T|. With E-D&G we know that a solution is a global optimum whenever no subtours are observed in the first iteration.

Tables 1 and 2 show the results for the objectives of revenue maximization and makespan minimization, respectively. The new explicit batching formulation (CT-EB) and the approach of Erdirik-Dogan & Grossmann[13] (E-D&G) are

Table 2. Overview of Computational Performance (CPUs) for Maximum Plant Flexibility and Makespan Minimization

		Model			
Problem	Optimum (h)	CT-IB	CT-EB	CT-SV	E-D&G
P1	171	5.34	0.53	1.55	**0.42**
P2	235	38.6	4.92	4080[c]	**0.64**
P3	167	364	**10.9**	3600[d]	1.75[f]
P4	130[a]	9400[b]	9645	64500[e]	**34.4**[g]

[a]May not be the global optimal solution.
[b]MRL, SO = 153, FTP (|T| = 8).
[c]SO = 236, although solver solved to optimality (special case).
[d]MRL, SO = 168, BPS = 152.5.
[e]MRL, SO = 133, BPS = 130.8.
[f]LB = 161, SO = 175, NIT = 4.
[g] LB = 129, NIT = 2.

Table 3. Detailed Computational Statistics for Problem P3 ($H = 120$)

	Model			
	CT-IB	CT-EB	CT-SV	E-D&G*
|T|	11	7		
Discrete variables	1840	960	85	440
Single variables	2083	1235	106	441
Constraints	253	445	233	515
RMIP	1336.88	1284.67	1346.06	1346.06
Obj	1218.5	1218.5	1212.42	1264.9
CPU	723	285	11842†	1.14
Nodes	171,407	130,436	18,092,900	351

*Results for first iteration only.
†OM, BPS = 1336.9.

undoubtedly the best ones. It is also apparent that the new approach should be preferred in the more complex problems, for which the latter typically generates a first solution with subtours, that when eliminated, lead to the degradation of the solution. This behavior was observed in five cases (entries in the last column of Table 1 with a superscript). The worst was found for P3 ($H = 120$ h) for which the lower bound after five iterations was equal to \$1029.52, 15.5% lower than the global optimal solution. However, it is also true that E-D&G is less demanding computationally so it may have the edge when trading-off between quality of the solution and computational effort. Problem P4 (makespan minimization) is the best example of this. With both CT-EB and E-D&G, we were able to find a solution of 130 h, but the former was 280 times slower. Furthermore, E-D&G provides us with the important additional information that the global optimal solution is at most 129 h. The other problem for which the global optimal solution is still unknown is P4 ($H = 120$ h), for which the optimal solution returned by CT-EB for five event points is equal to \$2068.58 for a possible maximum of \$2085.38 (upper bound from E-D&G). Because of the already high computational effort, we did not solve the problem for a higher number of event points, which would widen the feasible region.

The implicit batching CT-IB approach is a worse performer than CT-EB by typically one order of magnitude, an expected behavior since CT-IB requires a larger number of discrete variables and, most of the times, also exhibits a larger integrality gap (see Table 3 for detailed computational statistics). Contributing to the number of discrete variables are the number of binary and integer variables of the model. CT-EB uses exactly the same set of binary variables as CT-IB ($N_{i,i',m,t}$ being the most important) plus the integer variables $Z_{i,m}$. However, the number of binary extent variables in CT-IB will be significantly higher due to the need to use time grids with more event points (i.e. index t has a wider range) to find the exact same solution as CT-EB, as can be seen in Table 4.

At the bottom of the performance table we find CT-SV, which fails to find the optimal solution in the majority of the cases. It is worth noting that when compared to CT-IB, CT-SV requires about one-tenth the number of discrete variables, has a similar integrality gap but has a worse performance most of the times. The most notable exception to the rule is P3 ($H = 168$ h) that takes less than 1 min to solve as opposed to 4 h by CT-IB. However, this is a special problem with zero integrality gap due to the fact that the time horizon allows for all product demands to be met (makespan is 167 h, see Table 2). When compared to E-D&G, Table 3 shows that the use of global instead of immediate precedence sequencing variables leads to a significantly smaller number of variables and constraints. Continuous-time models with global precedence sequencing variables are known[8] to be able to find very good solutions fast, but also not being that good for proving optimality and CT-SV is no exception. Because of the smaller number of discrete variables, many nodes can be searched per second, but since the branch-and-bound tree rapidly explodes (i.e. the solver runs out of memory given sufficient time, see for example P4, $H = 120$ h), it can only be concluded that the model is not as tight as time grid-based models where each assignment brings the feasible region of the LP and the MILP closer together.

It is worth mentioning that if one were to restrict to process each product in a single unit, the proposed formulations can be reduced to simpler models that are solved more effectively. Table 5 gives the comparison in the value of the objective function for maximum plant flexibility vs. the case

Table 4. Number of Event Points (|T|) Used to Solve the Problem by the Implicit (CT-IB) and Explicit Batching Continuous-Time Formulations (CT-EB) for Maximum Plant Flexibility

		Revenue Maximization		Makespan Minimization	
Problem	H (h)	CT-IB	CT-EB	CT-IB	CT-EB
P1	120	9	4		
	144	10	4		
	168	11	4	12	4
P2	144	11	5		
	168	13	5		
	192	14	5	16	7
P3	120	11	7		
	144	12	7		
	168	13	7	14	7
P4	96	7*	5		
	120	8*	5		
	144	8*	4	8[a]	5

*Finding the optimal solution requires even more event points

Table 5. Improvement in the Value of the Objective Function for Maximum Plant Flexibility

	Revenue Maximization		Makespan Minimization
Problem	H (h)	% Increase	(% Decrease)
P1	120	0	8.06
	144	0	–
	168	1.32	–
P2	144	0.50	0.84
	168	0	–
	192	0	–
P3	120	0.50	0.60
	144	0	–
	168	0	–
P4	96	0.97	13.91
	120	3.86	–
	144	2.60	–

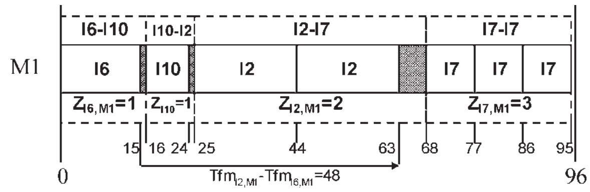

Figure 7. Part of the optimal schedule for example P4 (H = 96).

of a single unit. For revenue maximization, decreases for the single unit case were observed in half of the cases, with an average decrease of 0.81% and a maximum of 3.86% for P4 (H = 120 h). For makespan minimization, the single unit case was worse in all four cases, with an average of 5.85% less and a maximum of 13.91% less.

Model Limitations

There are a couple of results from Tables 1 and 2 that are very important but are hidden. CT-SV was able to solve P2 (for H = 144 h and makespan minimization) to optimality, but it returned a suboptimal solution thus highlighting the fact that CT-SV is not as general as the multiple time grid continuous-time models. In fact, CT-SV can fail for two different reasons, which will now be explained.

CT-SV

Castro et al.[8] reported that continuous-time models using global precedence sequencing variables can cutoff the real optimal solution from the feasible space in problems involving sequence-dependent changeovers. To explain this behavior, let us use the optimal solution of P4 (H = 96 h) and the schedule for unit M1, which features a I6-I10-I2-I7 sequence, see Figure 7. The relevant processing and changeover times (h) are $p_{I6,M1} = 15$, $p_{I10,M1} = 8$, $p_{I2,M1} = 19$, $p_{I7,M1} = 9$, $cl_{I6,I10,M1} = 1$, $cl_{I6,I2,M1} = 18$, $cl_{I10,I2,M1} = 1$, and $cl_{I2,I7,M1} = 5$. The optimal number of batches are in turn $Z_{I6,M1} = 1$, $Z_{I10,M1} = 1$, $Z_{I2,M1} = 2$, and $Z_{I7,M1} = 3$. It can be seen that the difference between the ending times of products I2 and I6 is equal to 48 h. However, this solution cannot be generated by CT-SV simply because Eq. 27, when applied to these two products and unit M1, gives $Tfm_{I2,M1} - Tfm_{I6,M1} \geq 56$ h (note that $X_{I2,I6} = 0$). Overall, of the 16 instances solved, the global precedence issue was responsible for six failures, which were identified after using the optimal values from CT-EB to fix the values of the integer variables $Z_{i,m}$ and then realize that either an infeasible solution (for revenue maximization) or a worse solution (for makespan minimization) was returned.

The second failure for CT-SV was observed with P2 for makespan minimization. The optimal schedule obtained by both CT-IB and CT-EB with MS = 235 h is shown in Figure 8. Notice that in M1, I2 precedes I6, which corresponds to $X_{I2,I6} = 1$, while in M2 it is product I6 that globally precedes I2, which corresponds to $X_{I2,I6} = 0$. Obviously, this cannot happen. Nevertheless, as previously mentioned, this can be overcome by disaggregating the sequencing variables over

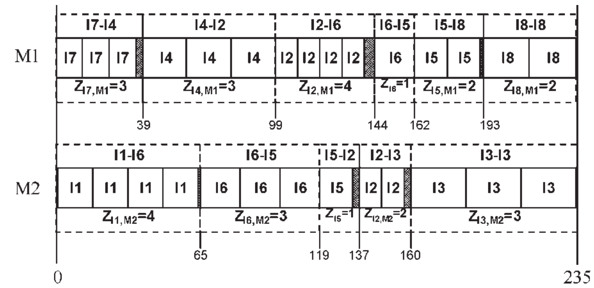

Figure 8. Optimal schedule for P2 (makespan minimization).

the equipment units, making it possible for $X_{I2,I6,M1} = 1$ and $X_{I2,I6,M2} = 0$.

E-D&G

The E-D&G model can generate solutions featuring subcycles that cannot be translated into a feasible schedule. To illustrate this issue, let us consider the solution from the first iteration of P3 (H = 144 h), which like all other problems, has asymmetric changeover time values. Six products are allocated to each machine and there are six subtours of two products, S_{IT1} = {S1, S2,. . .,S6}, see Figure 9. The correspondence of products to subtour, iteration, and unit is then used to generate sets $IS_{s,it,m}$, which become: $IS_{S1,IT1,M1}$ = {I2, I10}; $IS_{S2,IT1,M1}$ = {I3, I7}; $IS_{S3,IT1,M1}$ = {I5, I6}; $IS_{S4,IT1,M2}$ = {I1, I9}; $IS_{S5,IT1,M2}$ = {I3, I7}; $IS_{S6,IT1,M2}$ = {I4, I8}. For each machine, there are three systems to consider for the subtour elimination constraints, $AcS_{IT1,M1/M2}$ = {(S1,S2), (S1,S3), (S2,S3)} (see Eq. 45). The six subtour elimination constraints that were included in the second iteration of the algorithm are the following.

$$ZP_{I2,I3,M1} + ZP_{I2,I7,M1} + ZP_{I10,I3,M1} + ZP_{I10,I7,M1} \geq 1 \quad (46)$$

$$ZP_{I2,I5,M1} + ZP_{I2,I6,M1} + ZP_{I10,I5,M1} + ZP_{I10,I6,M1} \geq 1 \quad (47)$$

$$ZP_{I3,I5,M1} + ZP_{I3,I6,M1} + ZP_{I7,I5,M1} + ZP_{I7,I6,M1} \geq 1 \quad (48)$$

$$ZP_{I1,I3,M2} + ZP_{I1,I7,M2} + ZP_{I9,I3,M2} + ZP_{I9,I7,M2} \geq 1 \quad (49)$$

$$ZP_{I1,I4,M2} + ZP_{I1,I8,M2} + ZP_{I9,I4,M2} + ZP_{I9,I8,M2} \geq 1 \quad (50)$$

$$ZP_{I3,I4,M2} + ZP_{I3,I8,M2} + ZP_{I7,I4,M2} + ZP_{I7,I8,M2} \geq 1 \quad (51)$$

The second iteration returns a solution equal to \$1272.8, which is lower than the global optimal solution of \$1388.88.

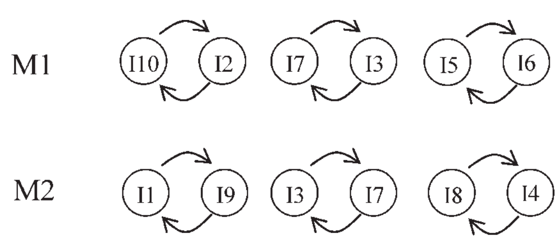

Figure 9. Solution of E-D&G for P3 (H = 144 h), first iteration.

This tells us that one of the above constraints has led to the elimination of the optimal solution from the feasible region. The optimal sequence of production is equal to I8-I7-I2-I10-I1 in M1 and I5-I6-I4-I9-I3-I7 in M2; so a single binary variable, $ZP_{I9,I3,M2}$, differs from zero and only Eq. 49 does not cutoff the optimal solution. The solution from the second iteration features two subtours in M1 and two other in M2: $IS_{S7,IT2,M1} = \{I2, I6, I7\}$; $IS_{S8,IT2,M1} = \{I3, I5, I9, I10\}$; $IS_{S9,IT2,M2} = \{I1,I8\}$; $IS_{S10,IT2,M2} = \{I3, I4, I7, I9\}$. Thus, the third iteration, besides Eqs. 46–51, features the following constraints:

$$ZP_{I2,I3,M1} + ZP_{I2,I5,M1} + ZP_{I2,I9,M1} + ZP_{I2,I10,M1} + ZP_{I6,I3,M1} + ZP_{I6,I5,M1} + ZP_{I6,I9,M1} + ZP_{I6,I10,M1} + ZP_{I7,I3,M1} + ZP_{I7,I5,M1} + ZP_{I7,I9,M1} + ZP_{I7,I10,M1} \geq 1 \quad (52)$$

$$ZP_{I1,I3,M2} + ZP_{I1,I4,M2} + ZP_{I1,I7,M2} + ZP_{I1,I9,M2} + ZP_{I8,I3,M2} + ZP_{I8,I4,M2} + ZP_{I8,I7,M2} + ZP_{I8,I9,M2} \geq 1 \quad (53)$$

The final solution, \$1261.68, corresponds to a sequence of production equal to I2-I6-I7-I10-I3-I5 in M1 and I1-I3-I7-I4-I9-I8, which clearly respects all the subtour elimination constraints.

Conclusions

This article has taken a fresh look at the optimal short-term scheduling of single stage multiproduct plants in cases involving multiple product batches. Two new models have been presented that include all individual processing tasks into a single aggregated task per product, whereas in the traditional approach, each batch corresponds to a single task. The number of batches of a product to produce is modeled as explicit integer variables that affect the duration of the task. Defining fewer tasks has the advantage of generating smaller models, which can generally be solved faster.

The performance of the new formulations has been illustrated through the solution of 12 example problems for the objective of revenue maximization and 4 for the objective of makespan minimization. The same problems were solved by a traditional scheduling approach as well as by a very recent bounding method relying on a model with immediate precedence sequencing variables. The new multiple time grid formulation emerged as the overall best performer. When compared to the related traditional approach, the computational effort was typically one order of magnitude lower, which in practice indicates that the new formulation can tackle larger problems. Nevertheless, it is important to emphasize that this conclusion is for single stage problems only. The traditional approach, on the other hand, can handle more general cases. The bounding method was found to be the fastest but it is not completely appropriate for short-term scheduling, since it assumes a single cyclic schedule in each unit. While this can be broken, two or more cyclic schedules per unit may result. In such cases, subtour elimination constraints can be added and the problem solved iteratively to find a feasible schedule at the likely expense of removing the global optimal solution from the feasible space.

Literature Cited

1. Méndez CA, Cerdá J, Grossmann IE, 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.
2. Shaik M, Janak S, Floudas C. Continuous-time models for short-term scheduling of multipurpose batch plants: a comparative study. *Ind Eng Chem Res* 2006;45:6190–6209.
3. Kondili E, Pantelides CC, Sargent R. A general algorithm for short-term scheduling of batch operations. I. MILP formulation. *Comput Chem Eng* 1993;17:211–227.
4. Pantelides CC. Unified frameworks for the optimal process planning and scheduling. In: *Proceedings of the Second Conference on Foundations of Computer Aided Operations*; New York: Cache Publications, 1994:253.
5. Castro PM, Barbosa-Póvoa AP, Matos HA, Novais AQ. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind Eng Chem Res* 2004;43:105–118.
6. Maravelias CT, Grossmann IE. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind Eng Chem Res* 2003;42:3056–3074.
7. Sundaramoorthy A, Karimi IA. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem Eng Sci* 2005;60:2679–2702.
8. Castro PM, Grossmann IE, Novais AQ. Two new continuous-time models for the scheduling of multistage batch plants with sequence dependent changeovers. *Ind Eng Chem Res* 2006;45:6210–6226.
9. Janak SL, Lin X, Floudas CA. 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.
10. Gupta S, Karimi IA. An improved MILP formulation for scheduling multiproduct, multistage batch plants. *Ind Eng Chem Res* 2003;42:2365–2380.
11. Harjunkoski I, Grossmann I. Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Comput Chem Eng* 2002;26:1533–1552.
12. Méndez CA, Henning GP, Cerdá J. An MILP continuous-time approach to short-term scheduling of resource constrained multistage flowshop batch facilities. *Comput Chem Eng* 2001;25:701–771.
13. Erdirik-Dogan M, Grossmann IE. Planning models for parallel batch reactors with sequence-dependent changeovers. *AIChE J* 2007;53:2284–2300.
14. Birewar DB, Grossmann IE. Simultaneous production planning and scheduling in multiproduct batch plants. *Ind Eng Chem Res* 1990;29:570–580.
15. Nemhauser G, Wolsey L. *Integer and Combinatorial Optimization*. New York: Wiley, 1988.
16. Birewar DB, Grossmann IE. Efficient optimization algorithms for zero-wait scheduling of multiproduct batch plants. *Ind Eng Chem Res* 1989;28:1333–1345.

Manuscript received Jun. 4, 2007, and revision received Sept. 28, 2007.