Simultaneous Batching and Scheduling of Single Stage Batch Plants

Pedro M. Castro,[*,†] *Muge Erdirik-Dogan*[‡] *and Ignacio E. Grossmann*[‡]

[†] Departamento de Modelação e Simulação de Processos, Instituto Nacional de Engenharia, Tecnologia e Inovação, 1649-038 Lisboa, Portugal

[‡] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

Abstract

This paper presents a new mixed integer linear program (MILP) for the optimal short-term scheduling of single stage batch plants with sequence dependent changeovers and optimal selection of the number of batches to produce. It is a continuous-time formulation employing multiple time grids that is based on the resource-task network (RTN) process representation. The main novelty is that aggregated processing and changeover tasks are considered that account for the time required to produce all batches of the product, plus the changeover time to the next product in the sequence. When compared to the traditional approach of considering a single processing task per batch, fewer event points are needed, which results in significantly lower computational effort as illustrated through the solution of several example problems. The new formulation is further compared to a continuous-time model with global precedence sequencing variables to a bounding model with immediate precedence sequencing variables and to a constraint programming model.

[*] To whom correspondence should be addressed. Tel.: +351-210924643. Fax: +351-217167016. E-mail: pedro.castro@ineti.pt

1. 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 Shaik et al.[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 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 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 pre-specified 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 in order 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 paper builds on recent work by the authors[8], who have compared different event representation models for the batch scheduling of multistage batch plants with sequence dependent changeovers. However, instead of considering that the number of batches for the production of a given order is known in advance, we solve the simultaneous batching and scheduling problem for single stage plants. A new RTN-based multiple time grid continuous-time formulation is proposed that considers the number of batches of a product to produce as explicit integer variables. It features aggregated tasks that include all the required batches of a particular order, where only one will need to be executed in a particular equipment unit. Each 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. In this way, we take advantage of the well known ability of continuous-time formulations to handle variable duration tasks. When compared to the traditional STN/RTN approach that implicitly determines the number of batches by the number of processing tasks that are executed, fewer tasks will be generally needed and since each requires one time interval, fewer event points will be required to achieve global optimal solutions.

To provide a better evaluation of its performance, the new approach is compared to a continuous-time model with global precedence sequencing variables[11] and to a constraint programming[18] model, besides to a multiple time grid, traditional RTN approach. A bounding model[14] using immediate precedence sequencing variables that does not use timing variables and can thus be viewed as a less constrained version of a pure scheduling model is also part of the comparison. Starting with a scenario of total production flexibility that is translated into the ability of producing different batches of the same product in multiple units, the formulations are simplified to the case where all batches of a product are produced in the same unit. This study is conducted for two alternative objective functions, revenue maximization and makespan minimization in order to increase our knowledge of the strengths and drawbacks of each model. 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, whereas for production of all batches of a product in a single unit we get a pure scheduling model.

2. Motivation and problem definition

The problem that we will address in this paper 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 paper, the optimal short-term scheduling of single stage batch plants is considered 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/non-identical batch equipment units. An example involving five products (A-E) and two reactors (R1-R2) is given in Figure 1. 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.

In order 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 2). Nevertheless, we do consider such constraint as a special case and study its impact on model simplification, quality of the solution and computational effort. Note in Figure 2 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.

3. Multiple Time Grid Continuous Models

The two continuous-time formulations considered in this section rely on the Resource-Task Network representation[4] and employ multiple time grids, one for each equipment unit. They also use four-index binary variables linked to the execution of combined processing and changeover tasks, so they are conceptually similar to formulation CT4I developed by Castro et al.[8]. However, that formulation assumed the execution of just one processing task per product, with the amount of material processed (resulting from one or more batches) being implicit on the task duration. In order to deal with material amounts and also with variable duration tasks, the new formulations have used insights from the single time grid, short-term scheduling formulation of Castro et al.[5].

The two new formulations differ conceptually on the definition of a processing task. The traditional way[1] 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. We will be calling this the implicit batching approach and refer to it as CT-IB (see Figure 2). CT-IB can be seen as a minor upgrade from CT4I[8], concerning the generality of single stage problems that can be handled.

Time grid based continuous-time formulations[5-7,9] can handle variable duration tasks without significantly altering the complexity of the model. In the second 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 novel approach 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 3.

The underlying time grid is given in Figure 4. 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 Figure 2 and Figure 3, one can see that CT-IB requires a total of 5 event points while CT-EB requires just 4. 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 where more event points typically means worse performances, such a small example perfectly 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.

3.1. Implicit Batching Approach (CT-IB)

The implicit batching continuous-time formulation (CT-IB) features two sets of binary variables. Extent variables, $N_{i,i',m,t}$, identify the execution of the processing task, starting at event point t, required to produce product i in unit m followed by the changeover task that makes the unit ready to process product i' immediately after. The second set is linked to the initial condition of unit m, determined through variables $C^0_{i,m}$, which can be fixed if the initial state is known. All continuous variables are nonnegative and include the excess resource variables, $C_{i,m,t}$, that 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 time of event point t belonging to time grid m and MS is the makespan.

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 4) only tasks with the same product index can be performed since there are no slots left to process any more tasks (see also Figure 2). Set I_m includes the orders that can be processed in unit m, those that have a nonzero duration, see eq 3.

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

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

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

Regarding the model constraints we start with the excess resource balances, which can be viewed as multiperiod material balance expressions where the excess amount at point t is equal to that at point t-1 adjusted by the amounts produced/consumed by all tasks starting or ending at t. In eq 4 the initial state variables (first term on the right-hand side) only appear in constraints related to the first event point. For unit m, condition i is produced by all tasks (i',i) processed in m and starting at t-1 and is consumed by tasks (i,i'), also processed in unit m, starting at t. Eq 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.

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

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

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

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 2).

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

Whenever the objective is makespan minimization, the timing variables need to be related to the makespan (MS), which must be greater than the ending time of all tasks. Eq 8 ensures that the makespan is greater than the absolute time of event point t plus the duration of all combined tasks starting at or after that event point, for all equipment units.

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

According to Figure 4, 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.

$$T_{1,m} = 0\ \forall m\in M \tag{9}$$

$$T_{t,m} \leq H\ \forall m\in M, t\in T \tag{10}$$

Finally, we conclude with the demand constraint that depends on the objective function. For profit maximization, the production 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 ≥ is used instead.

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

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

3.1.1. Remarks

Although CT-IB considers tasks with known batch sizes and fixed duration, it is straightforward[5] to adapt it in order to consider tasks with variable batch sizes and a duration that are dependent on the amount of material that is processed (if the relation is made linear the model remains as a MILP). CT-IB will then allow different processing instances of the same product to handle different batch sizes, which will increase significantly the variety of production amounts that can be obtained. At the same time, the total duration of the tasks will also belong to a continuous rather than to a discrete domain so the model can use more efficiently the available time horizon. CT-IB can thus be more 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 non-time 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.

On more technical issues, because of eq 7, tasks will tend to be executed from the last to the first time interval, similarly to CT3I[8] and contrary to CT4I[8]. Eq 13 can be added to enforce unit availability to decrease from start to finish with the exception of the last event point, where all equipment units become available. The idea is to reduce the number of degenerate solutions and facilitate the search procedure. However, adding more constraints also makes the mathematical problem more complex. Since computational studies have shown the latter effect to be more important than the former, eq 13 was not included.

$$\sum_{i \in I_m} (C_{i,m,t} - C_{i,m,t-1}) \leq 0 \; \forall m \in M, t \in]1,|T|[\tag{13}$$

3.2. Explicit Batching Approach (CT-EB)

The explicit batching continuous-time formulation (CT-EB) requires two sets of variables to characterize an aggregated task. It shares the binary extent variables $N_{i,i',m,t}$ with CT-IB while also using the continuous extent variables $\xi_{i,m,t}$, which 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. All other CT-IB variables are also employed by CT-EB. Regarding the model constraints, CT-EB uses eqs. 4-6, 9-10 and a few more sets other than those that are very similar to the ones in CT-IB. We will be focusing only on the features that are different.

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 14). 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 15.

$$\sum_{t \in T} \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} \leq 1 \; \forall m \in M, i \in I_m \tag{14}$$

$$I_{i',m,t} = \{ i \in I_m : (t \neq |T|-1 \wedge i \neq i') \vee (t = |T|-1 \wedge i = i') \} \; \forall m \in M, i' \in I_m, t \in T, t \neq |T| \tag{15}$$

The maximization of sales revenue objective function is given by eq 16, 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 17. Eq 18 places an upper bound on the integer variables. The ceiling function has been used due to the fact that the total production of i cannot exceed its demand (eq 19) for the sales revenue objective. For makespan minimization, the equality is used instead (eq 20).

$$\max \sum_{\substack{t \in T \\ t \neq |T|}} \sum_{m \in M} \sum_{i \in I_m} v_i \cdot \xi_{i,m,t} \tag{16}$$

$$\sum_{\substack{t \in T \\ t \neq |T|}} \xi_{i,m,t} = b_{i,m} \cdot Z_{i,m} \; \forall m \in M, i \in I_m \tag{17}$$

$$\overline{Z}_{i,m} = \lfloor \Delta_i / b_{i,m} \rfloor \forall m \in M, i \in I_m \tag{18}$$

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

$$\sum_{\substack{t \in T \\ t \neq |T|}} \sum_{\substack{m \in M \\ i \in I_m}} \xi_{i,m,t} = \Delta_i \; \forall i \in I \tag{20}$$

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 19-20, where the upper bound on the amount produced is the product demand, and the lower bound is the batch size.

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

$$\xi_{i,m,t} \geq b_{i,m} \cdot \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} \quad \forall m \in M, i \in I_m, t \in T, t \neq |T| \tag{22}$$

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 only for tasks not executed in the last time interval. Figure 5 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 eq 23. Finally, eq 24 is the equivalent of eq 8, and is only to be used for makespan minimization.

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

$$MS \geq T_{t,m} + \sum_{\substack{t' \in T \\ t' \geq t \\ t' \neq |T|}} \sum_{i \in I_m} \left[\frac{\xi_{i,m,t'}}{b_{i,m}} \cdot (p_{i,m} + cl_{i,i,m}) + \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t'}}} N_{i,i',m,t'} \cdot \left(cl_{i,i',m}\Big|_{t \neq |T|-1} - cl_{i,i,m} \right) \right] \forall m \in M, t \in [1, |T|[\tag{24}$$

3.2.1. Remarks

An important simplification of the problem is achieved if all batches of a product are allocated to a single equipment unit. This choice has the advantage of allowing the consideration of simpler mathematical formulations that can effectively handle larger problem sizes. The obvious disadvantage is that the plant equipment units are not used to their full potential, as can be seen in Figure 6, where the three batches of product I3 can no longer be produced in the given time horizon (compare to Figure 3). As a consequence, the revenue has decreased and the idle time of unit M1 has increased.

For the sales revenue objective function, the explicit batching, with products allocated to a single unit (CT-EB-SU), continuous-time formulation, is very similar to CT-EB. A single adjustment is possible in terms of model variables whereas a couple of constraints can be made tighter. Integer variables $Z_{i,m}$ are replaced by integer variables W_i since at most one unit will be involved in the production of i, and their upper bounds can be determined through eq 25. The number of batches of i are now equal to the total amount produced divided by the corresponding batch size, see eq 26 that replaces eq 17. Eq 27 then ensures that i is not produced in more than one equipment unit (it replaces eq 14).

$$\overline{W_i} = \left\lfloor \max_{\substack{m \in M \\ i \in I_m}} \Delta_i / b_{i,m} \right\rfloor \forall i \in I \tag{25}$$

$$\sum_{\substack{m \in M \\ i \in I_m}} \sum_{\substack{t \in T \\ t \neq |T|}} \xi_{i,m,t} / b_{i,m} = W_i \ \forall i \in I \tag{26}$$

$$\sum_{t \in T} \sum_{m \in M} \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} \leq 1 \ \forall i \in I \tag{27}$$

When minimizing makespan, CT-EB-SU becomes very similar to the implicit batching approach CT-IB, although the domain of variables $N_{i,i',m,t}$ is the one of CT-EB, given by eq 15. No integer variables are needed since the number of batches that are required to achieve a certain production can be determined a priori. This information is then incorporated into the processing time data, which naturally must include all required changeovers but the last, considered in the corresponding timing constraint (eq 7). The new processing times are calculated through eq 28 and the demand constraints (eq 12) can be dropped from the formulation. Finally, eq 29 ensures that each product is processed exactly once.

$$p_{i,m} = (p_{i,m} + cl_{i,i,m}) \cdot \lceil \Delta_i / b_{i,m} \rceil - cl_{i,i,m} \ \forall m \in M, i \in I_m \tag{28}$$

$$\sum_{t \in T} \sum_{m \in M} \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} = 1 \ \forall i \in I \tag{29}$$

4. Continuous-Time Model with Global Precedence Sequencing Variables (CT-SV)

Continuous-time scheduling formulations for single/multistage processes do not need to rely on an explicit time grid. The notion of global precedence can be used to relate processing tasks of different products and hence avoid the drawback of requiring the specification of the total number of event points before solving the problem, which may compromise the quality of the solution. The continuous-time model with global precedence sequencing variables (CT-SV) shown next, needs to be solved only once to find the global optimal solution to the problem (if solved to zero optimality gap).

Model CT-SV builds on the work of Harjunkoski and Grossmann[11] and employs one sequencing variable 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*. The novelty of CT-SV is that it considers multiple batches for each product instead of a single one. Like for CT-EB, integer variables $Z_{i,m}$ give the number of batches of *i* processed in unit *m*. The full processing task for product *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 7. Another adjustment[11] is that the ending time of the full 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 30 represents the maximization of the revenue from product sales. Eq 31 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. Eqs 32-33 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 34. Eq 35 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 58 (see 5). Eq 36 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 37.

$$\max \sum_{m\in M}\sum_{i\in I_m} v_i \cdot b_{i,m} \cdot Z_{i,m} \tag{30}$$

$$Tfm_{i,m} \geq Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m} \ \forall m \in M, i \in I_m \tag{31}$$

$$\begin{aligned} &Tfm_{i',m} \geq Tfm_{i,m} + Z_{i',m} \cdot (p_{i',m} + cl_{i',i',m}) - cl_{i',i',m} \cdot Y_{i',m} + cl_{i,i',m} \cdot X_{i,i'} - H \cdot (3 - X_{i,i'} - Y_{i,m} - Y_{i',m}) \\ &\forall m \in M, i, i' \in I_m, i' > i \end{aligned} \tag{32}$$

$$\begin{aligned} &Tfm_{i,m} \geq Tfm_{i',m} + Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m} + cl_{i',i,m} \cdot (1 - X_{i,i'}) - H \cdot (2 + X_{i,i'} - Y_{i,m} - Y_{i',m}) \\ &\forall m \in M, i, i' \in I_m, i' > i \end{aligned} \tag{33}$$

$$Tfm_{i,m} \leq H \ \forall m \in M, i \in I_m \tag{34}$$

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] \leq H \ \forall m \in M \tag{35}$$

$$Z_{i,m} \leq \lceil \Delta_i / b_{i,m} \rceil \cdot Y_{i,m} \ \forall m \in M, i \in I_m \tag{36}$$

$$\sum_{\substack{m \in M \\ i \in I_m}} b_{i,m} \cdot Z_{i,m} \leq \Delta_i \ \forall m \in M, i \in I \tag{37}$$

When minimizing makespan, besides turning eq 37 into an equality or the opposite inequality, and replacing eq 35 with the tighter eq 38, one additional set of constraints is required. Eq 39 ensures that the makespan, *MS*, is greater than the ending time of all tasks.

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] \leq MS \ \forall m \in M \tag{38}$$

$$MS \geq Tfm_{i,m} \ \forall m \in M, i \in I_m \tag{39}$$

4.1. 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 7.2 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.

Simplifying the problem by assuming that a product is allocated to a single equipment unit significantly increases the performance of the model, as will be seen in section 6. In terms of variables, the unit index can be dropped from the timing variables in model CT-SV-SU, which become Tf_i. Regarding the constraints, a new set is required to ensure allocation of each product to at most a single equipment unit, eq 40. In addition, minor adjustments are needed for some of the previous sets due to the use of different timing variables. More specifically, for the revenue objective function, eqs 31-34 are replaced by eqs 41-44. Eqs 35-36 remain the same while eq 18 acts as the demand constraint, instead of eq 37.

$$\sum_{\substack{m\in M\\ i\in I_m}} Y_{i,m} \leq 1 \ \forall i \in I \tag{40}$$

$$Tf_i \geq \sum_{\substack{m\in M\\ i\in I_m}} [Z_{i,m}\cdot(p_{i,m}+cl_{i,i,m}) - cl_{i,i,m}\cdot Y_{i,m}] \ \forall i \in I \tag{41}$$

$$\begin{gathered} Tf_{i'} \geq Tf_i + Z_{i',m}\cdot(p_{i',m}+cl_{i',i',m}) - cl_{i',i',m}\cdot Y_{i',m} + cl_{i,i',m}\cdot X_{i,i'} - H\cdot(3 - X_{i,i'} - Y_{i,m} - Y_{i',m}) \\ \forall m \in M, i,i' \in I_m, i' > i \end{gathered} \tag{42}$$

$$\begin{gathered} Tf_i \geq Tf_{i'} + Z_{i,m}\cdot(p_{i,m}+cl_{i,i,m}) - cl_{i,i,m}\cdot Y_{i,m} + cl_{i',i,m}\cdot(1 - X_{i,i'}) - H\cdot(2 + X_{i,i'} - Y_{i,m} - Y_{i',m}) \\ \forall m \in M, i,i' \in I_m, i' > i \end{gathered} \tag{43}$$

$$Tf_i \leq H \ \forall i \in I \tag{44}$$

For makespan minimization and similarly to CT-EB-SU, we can use eq 28 to define the processing time of the full processing tasks. As a consequence, the timing constraints are now simpler (see eqs 45-49). We then ensure through eq 50 that each product needs to be allocated to exactly one unit.

$$Tf_i \geq \sum_{\substack{m\in M\\ i\in I_m}} Y_{i,m}\cdot p_{i,m} \ \forall i \in I \tag{45}$$

$$Tf_{i'} \geq Tf_i + p_{i',m} + cl_{i,i',m}\cdot X_{i,i'} - H\cdot(3 - X_{i,i'} - Y_{i,m} - Y_{i',m}) \ \forall m \in M, i,i' \in I_m, i' > i \tag{46}$$

$$Tf_i \geq Tf_{i'} + p_{i,m} + cl_{i',i,m}\cdot(1 - X_{i,i'}) - H\cdot(2 + X_{i,i'} - Y_{i,m} - Y_{i',m}) \ \forall m \in M, i,i' \in I_m, i' > i \tag{47}$$

$$\sum_{i\in I_m} Y_{i,m}\cdot p_{i,m} \leq MS \ \forall m \in M \tag{48}$$

$$MS \geq Tf_i \ \forall i \in I \tag{49}$$

$$\sum_{\substack{m\in M\\ i\in I_m}} Y_{i,m} = 1 \ \forall i \in I \tag{50}$$

5. Model of Erdirik-Dogan and Grossmann[14] with Immediate Precedence Sequencing Variables (E-D&G)

Erdirik-Dogan and Grossmann[14] 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[15]. Immediate precedence sequencing variables are used and the sequencing constraints can be regarded as a relaxation of the traveling salesman problem[16].

The E-D&G model has the potential drawback of generating solutions featuring subcycles even though for asymmetric sequence-dependent changeovers the likelihood is 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 section 7. 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. General subtour elimination constraints of the type given in Birewar & Grossmann[17], do not compromise optimality but increase the size of the problem greatly if the number of products is high. The alternative used in this paper, is to introduce these constraints for only the set of products involved in the various subcycles. In such 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 & Grossmann[14], 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 $ZZP_{i,i',m}$ identify if the link between i and i' in unit m is to be broken (see Figure 8). The E-D&G model constraints are given next together with a brief explanation.

$$Y_{i,m} = \sum_{i' \in I_m} ZP_{i,i',m} \ \forall m \in M, i \in I_m \tag{51}$$

$$Y_{i,m} = \sum_{i' \in I_m} ZP_{i',i,m} \ \forall m \in M, i \in I_m \tag{52}$$

$$\sum_{i \in I_m} \sum_{i' \in I_m} ZZP_{i,i',m} = 1 \ \forall m \in M \tag{53}$$

$$ZZP_{i,i',m} \le ZP_{i,i',m} \ \forall m \in M, i \in I_m, i' \in I_m \tag{54}$$

$$Y_{i,m} \ge ZP_{i,i,m} \ \forall m \in M, i \in I_m \tag{55}$$

$$ZP_{i,i,m} + Y_{i',m} \le 1 \ \forall m \in M, i \in I_m, i' \in I_m, i \ne i' \tag{56}$$

$$ZP_{i,i,m} \ge Y_{i,m} - \sum_{\substack{i' \in I_m \\ i' \ne i}} Y_{i',m} \ \forall m \in M, i \in I_m \tag{57}$$

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] + \sum_{i \in I_m} \sum_{i' \in I_m} cl_{i,i',m} \cdot (ZP_{i,i',m} - ZZP_{i,i',m}) \le H \ \forall m \in M \tag{58}$$

$$Z_{i,m} \ge Y_{i,m} \ \forall m \in M, i \in I_m \tag{59}$$

Eqs 51-52 ensure that if product *i* is assigned to unit *m* there is exactly one transition to/from product *i'* in that unit. Eq 53 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 54. 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 55-57). Eq 58 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 35 due to the use of immediate instead of global precedence sequencing variables. Eq 59 guarantees that only in cases where product *i* is assigned to unit *m*, can there be batches of *i* be produced in *m*. The remaining sets of constraints, eqs 36-37 together with the revenue maximization objective function, eq 30, are shared with CT-SV.

When minimizing makespan, and besides the previously mentioned changes to eq 37, one just needs to replace eq 58 by eq 60.

$$\sum_{i \in I_m} [Z_{i,m} \cdot (p_{i,m} + cl_{i,i,m}) - cl_{i,i,m} \cdot Y_{i,m}] + \sum_{i \in I_m} \sum_{i' \in I_m} cl_{i,i',m} \cdot (ZP_{i,i',m} - ZZP_{i,i',m}) \le MS \ \forall m \in M \tag{60}$$

Whenever the optimal solution features subtours, the model needs to be re-solved with the subtour elimination constraints (eq 61). 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. Eq 61 is applied for every pair of subtours belonging to the set of active systems, $AcS_{it,m}$, which is defined by eq 62. The elements of sets $IS_{s,it,m}$ and $AcS_{it,m}$ are given in section 7.3 for one example.

$$\sum_{i \in IS_{s,it,m}} \sum_{i' \in IS_{s',it,m}} ZP_{i,i',m} \geq 1 \; \forall it \in IT, m \in M, (s,s') \in AcS_{it,m} \tag{61}$$

$$AcS_{it,m} = \{(s,s') \; s,s' \in S_{it}, s < s' : \sum_{i \in IS_{s,it,m}} 1 > 1 \wedge \sum_{i \in IS_{s',it,m}} 1 > 1\} \; \forall it \in IT, m \in M \tag{62}$$

5.1. Remarks

With immediate precedence sequencing variables it is straightforward to account for transitions costs in the objective function.

To ensure that a given product is allocated to a single unit, model E-D&G-SU can be used. When compared to the base models, eq 40 replaces eq 37 for revenue maximization, while for makespan minimization eq 50 needs to be added.

6. Constraint Programming Model (CP-SU)

Constraint programming[13], originally developed to solve feasibility problems, can also solve optimization problems. Constraint programming (CP) algorithms are very efficient for some classes of problems, among which scheduling is a prominent one. The CP formulations shown here are based on ILOG's OPL Studio modeling language[18]. Contrary to the previous MILP models, CP scheduling models do not rely explicitly on time grids neither on sequencing constraints. Instead, task (activity) sequencing is implicit through starting and ending times in the equipment unit where they have been allocated. Nevertheless, CP scheduling models seem to rely on global precedence sequencing variables since one such model exhibited the same limitations as a global precedence sequencing variable model in the solution of a single stage problem with sequence dependent changeovers[8]. CP scheduling models are also similar to RTN models in the sense that both have as their two main components activities (tasks) and resources. Activities and resources, which can be of three different types (equipment units correspond to unary resources), are the special constructs of the model. These are then linked to global constraints (e.g. requires) to build up the scheduling model. For a brief description of such OPL Studio's entities and their possible use for solving scheduling problems of higher structural complexity, the reader is directed to the work of Maravelias and Grossmann[19].

The constraint programming model considered assumes that a particular product is allocated to a single equipment unit (CP-SU). A more general model, able of handling the scenario of total plant flexibility, was also developed but it exhibited a very low performance and for this reason it is not shown here. The activities of model CP-SU concern the execution of product *i* and are subject to a transition time that is accessed through the corresponding element in the transition matrix, see eq 63. The transition matrix is given by the parameter *chgover*[Units,Products,Products] (equivalent to $cl_{i,i',m}$), with each row being associated to a particular equipment unit (eq 64). Each activity can be executed in one of *m* units, so these machines are alternative resources from the activity standpoint, see eq 65. Integer variables W_i indicate the number of batches of product *i* to produce (upper bound given by eq 25), while set Q_i identifies the unit allocated to product *i*: $Q_i \in M$, $\forall i \in I$. The amount of *i* that is produced will be given by A_i, which is bounded by the product demand..

Activity DO[*i* in Products] transitionType *i* (63)

UnaryResource unit[*m* in Units] (*chgover*[*m*]) (64)

AlternativeResources Machines[unit] (*chgover*[*m*]) (65)

Concerning the model constraints, eq 66 is the objective function for revenue maximization. Eq 67 relates the amount of *i* produced to the number of batches executed and to the batch size of the unit where the product is executed. Notice the use of variable Q_i as an index (to determine the relevant unit), an important feature of CP models. Eq 68 states that each activity requires one of the available equipment units, while eq 69 determines the activity's duration (activities are equivalent to the full processing tasks in Figure 7). Notice the inequality sign, which is needed because the duration of an activity must be nonnegative. It is also important to mention that all activities will be performed even though no batches of product *i* are produced (such activities will have a zero duration). Furthermore, zero duration activities are not subject to changeovers so it is exactly as if those activities were not being performed. Finally, eq 70 states that selecting unit *m* from the set of alternative machines to process product *i*, is equivalent to allocating *i* to unit *m*.

$$\max \sum_{i\in I} v[i]\cdot A[i] \tag{66}$$

$$A[i] = W[i]\cdot b[i,Q[i]]\ \forall i\in I \tag{67}$$

$$\text{DO}[i]\text{ requires Machines }\forall i\in I \tag{68}$$

$$\text{DO}[i]\text{.duration} \geq p[i,Q[i]]\cdot W[i] + (W[i]-1)\cdot chgover[Q[i],i,i]\ \forall i\in I \tag{69}$$

$$\text{activityHasSelectedResource}(\text{DO}[i],\text{Machines},\text{unit}[m]) \Leftrightarrow Q[i]=m\ \forall m\in M, i\in I_m \tag{70}$$

When minimizing makespan the first thing to do is to determine the duration of the full processing task through eq 28. Then, besides eqs 68 and 70, eqs 71-72 are needed.

$$\min\left(\max_{i\in I} \text{DO}[i]\text{.end}\right) \tag{71}$$

$$\text{DO}[i]\text{.duration} = p[i,Q[i]]\ \forall i\in I \tag{72}$$

7. Computational Studies

The performance of the eight 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 as Supporting Information. 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 one hour, or up to the moment the solver ran out of memory. The CP models were in turn implemented and solved in

ILOG's OPL Studio 3.7.1 and identical termination criteria were used. The computer used was a Pentium-4 3.4 GHz processor with 2 GB of RAM, running Windows XP Professional.

The results have been divided into those generated under the option of maximum plant flexibility and those under the restriction of all batches of a product allocated to a single equipment unit. 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 and constraint programming based approaches are not entirely general, for reasons explained in section 7.2, 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.

7.1. Maximum Plant Flexibility

Table 1 and Table 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[14] (E-D&G) are undoubtedly the best approaches. 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 in five cases (entries in the table with a superscript). The worst, was found for P3 (H=120 h) for which the lower bound after 5 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 5 event points is equal to \$2068.58 for a possible maximum of \$2085.38 (upper bound from E-D&G). Due to 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 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 one minute to solve as opposed to 4 hours 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. Due to 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.

There are a couple of results from Table 1 and Table 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.

7.2. Limitations of CT-SV and CP models

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 9. 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, $Z_{I7,M1}$=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 33, when applied to these two products and unit M1, gives: $Tfm_{I2,M1}$-$Tfm_{I6,M1}$≥56 h (note that $X_{I2,I6}$=0). Overall, of the 16 instances solved for maximum plant flexibility, the global precedence issue was responsible for 6 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. With the problems constrained to production of a particular product in a single unit, CT-SV only failed for P3, H=144 h. This limitation is common to CP, which has global precedence sequencing variables that are implicit in its global constraints.

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 10. 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 mentioned in section 4.1, this can be overcome by disaggregating the sequencing variables over the equipment units, making it possible for $X_{I2,I6,M1}$=1 and $X_{I2,I6,M2}$=0.

7.3. Limitation of model 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 11. 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 62). The six subtour elimination constraints that were included in the second iteration of the algorithm are the following.

$$ZP_{I2,I3,M1} + ZP_{I2,I7,M1} + ZP_{I10,I3,M1} + ZP_{I10,I7,M1} \geq 1 \quad (73)$$

$$ZP_{I2,I5,M1} + ZP_{I2,I6,M1} + ZP_{I10,I5,M1} + ZP_{I10,I6,M1} \geq 1 \quad (74)$$

$$ZP_{I3,I5,M1} + ZP_{I3,I6,M1} + ZP_{I7,I5,M1} + ZP_{I7,I6,M1} \geq 1 \quad (75)$$

$$ZP_{I1,I3,M2} + ZP_{I1,I7,M2} + ZP_{I9,I3,M2} + ZP_{I9,I7,M2} \geq 1 \quad (76)$$

$$ZP_{I1,I4,M2} + ZP_{I1,I8,M2} + ZP_{I9,I4,M2} + ZP_{I9,I8,M2} \geq 1 \quad (77)$$

$$ZP_{I3,I4,M2} + ZP_{I3,I8,M2} + ZP_{I7,I4,M2} + ZP_{I7,I8,M2} \geq 1 \quad (78)$$

The second iteration returns a solution equal to $1272.8, which is lower than the global optimal solution of $1388.88. 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 76 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 73-78, features the following constraints.

$$\begin{aligned}&ZP_{I2,I3,M1} + ZP_{I2,I5,M1} + ZP_{I2,I9,M1} + ZP_{I2,I10,M1} + ZP_{I6,I3,M1} + ZP_{I6,I5,M1} + ZP_{I6,I9,M1} + ZP_{I6,I10,M1} + \\ &ZP_{I7,I3,M1} + ZP_{I7,I5,M1} + ZP_{I7,I9,M1} + ZP_{I7,I10,M1} \geq 1\end{aligned} \quad (79)$$

$$ZP_{I1,I3,M2}+ZP_{I1,I4,M2}+ZP_{I1,I7,M2}+ZP_{I1,I9,M2}+ZP_{I8,I3,M2}+ZP_{I8,I4,M2}+ZP_{I8,I7,M2}+ZP_{I8,I9,M2}\geq 1 \quad (80)$$

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.

7.4. All Batches of a Given Product in the same Unit

The flexibility of the plant is reduced if production of a product is restricted to a single equipment unit. From the modeling point of view, this option has the advantage of leading to simpler mathematical formulations and typically to lower computational efforts. Furthermore it is even possible that the optimal solution to the restricted problem, which is at most as good as the optimal solution to the unrestricted problem, is better than the best one found for the unrestricted problem up to a considerable computational resource limit. The results in Table 5 for CT-SV-SU are evidence of this behavior, when compared to those given in Table 1 for CT-SV. Notably, P2 (H=168 h) and P4 (H=96 h) are solved in just 45 and 52.1 s, respectively, and solutions of \$1329.2 and \$1725.34 are found, while CT-SV can only reach \$1313.8 and \$1715.6 until the moment the solver runs out of memory. Overall, for problems that were solved to global optimality, there were only two exceptions to the rule that the constrained problem is easier to solve than its unconstrained counterpart. The computational effort for P4 (H=144 h) increased by more than two orders of magnitude simply because the complexity of the mathematical problem has increased substantially due to the fact that 144 h is no longer enough to meet all maximum demands (the makespan has increased from 130 to 151 h, see Table 2 and Table 6). For P3 (H=168 h) with CT-SV-SU the difference is not as significant since the new makespan, 168 h, although higher than 167 h, still allows for maximum production to be achieved, with fewer degrees of freedom.

The results in Table 5 also show that CT-EB-SU and CT-SV-SU have similar performances. Recall, however, that CT-EB-SU has the disadvantage of needing an iterative procedure over |T|, while CT-SV-SU only needs to solve each problem once. E-D&G-SU failed to find the optimal solution in all instances of P3 but was able to solve all problems in less than a minute. The constraint programming approach is the worst performer for revenue maximization since it can only solve to optimality the three simplest problems. Problem P2 is already too hard to solve by CP-SU and very poor solutions are frequently obtained after 1 h of computational time.

For makespan minimization, the results in Table 6 show that CP-SU is the best performer by far, able to solve all four problems in less than one second. It is worth to emphasize that under the restriction of production in a single unit and fixed demands, the three models considered in this section are submodels of those given in Castro et al.[8], which can also handle release and due dates and be used in multistage plants. Larger problems involving 15 orders in 5 units were solved to global optimality for makespan minimization in less than one hour of computational time and the constraint programming model was found to be the best performer so the results in this paper are consistent with those findings. It is also interesting to see that for the three formulations, whenever the time horizon is sufficient to meet all product demands, it is a better option to solve for makespan minimization than for revenue maximization (e.g. P3, H=168 h). Furthermore, the solution will typically be better since the units will feature shorter changeovers or be idle for less time, in cases were the makespan is lower than the pre-specified time horizon.

From the above results, is it worth to use more complex models and allow the production process of the plant to be more flexible? Table 7 gives the improvements in the value of the objective function for maximum plant flexibility. For revenue maximization gains were observed in half of the cases, with an average increase of 0.81% and a maximum of 3.86% for P4 (H=120 h). For makespan minimization it was a better approach in all 4 cases, with a far better average of 5.85% and a maximum of 13.91%. Thus, it can only be concluded that the answer to the question is yes.

To end this discussion we should emphasize that we are looking into revenue maximization for a fixed time horizon or makespan minimization for production of all the required batches of the given product orders. For the latter case, the derived schedule will typically be used as the starting point for scheduling the next set of orders that will arrive at the plant. The objective of makespan minimization for the partial schedules attempts to maximize the productivity of the plant by reducing changeovers and idle times. However, it looks at the equipment units as a system and not as components. For maximum plant flexibility, the equipment units will tend to end at about the same time and this is achieved by distributing batches of one or more products through the parallel units.

For example, in Figure 10, M1 and M2 end both at 235 h. If on the other hand, batches of the same product are restricted to be produced in a single unit, fewer changeovers will be required and the productivity of the plant will increase despite the fact that the makespan will be higher. In Figure 12, unit M1 ends at 237 h but M2 ends before that, at 225 h, for an average completion time of 231 h, which is lower than the 235 h obtained for maximum plant flexibility.

8. Conclusions

This paper has presented a new continuous-time formulation for the optimal short-term scheduling of single stage multiproduct plants, where the number of batches of a product to produce in a particular unit are explicit integer variables. All individual processing tasks, one per batch, are included in a single aggregated task per product, whereas in the traditional approach, each batch corresponds to a single task. Since each task occupies exactly one time interval, fewer tasks leads to time grids consisting of fewer event points and, consequently, a lower number of model variables and constraints.

The performance of the new formulation 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 multiple time grid implicit batching approach, by a continuous-time model with global precedence sequencing variables, by a model with immediate precedence sequencing variables that does not determine the timing of events, and by a constraint programming model. The new formulation emerged overall as the best performer for the scenario of maximum plant flexibility, where different batches of the same product can be produced in different units. The model with immediate precedence sequencing variables is the fastest but it is not a general scheduling model in the sense that it assumes a single cyclic schedule in each unit, which can be broken, but 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. When compared to the implicit batching approach, the computational effort of the new formulation was typically one order of magnitude lower, which in practice indicates that the new formulation can tackle larger problems.

The other goal of the paper has been to study the effect of considering a less flexible production mode on model simplification. Restricting all batches of a product to a single unit allows for a reduction in the number of variables and/or constraints and/or the use of tighter constraints, which make the model simpler and generally faster. The drawback is that worse solutions than those obtained for the more flexible plant may result. It was particularly interesting to find out that the models reacted differently, with the model with global precedence sequencing variables approaching the performance of the new approach for revenue maximization, and the constraint programming model overcoming the other two for makespan minimization.

Supporting Information Available: Tables of data in GAMS format for problems P1-P4.

References

(1) Méndez, C.A.; Cerdá, J.; Grossmann, I.E.; Harjunkoski, I.; Fahl, M. State-of-the-art Review of Optimization Methods for Short-Term Scheduling of Batch Processes. Comput. Chem. Eng. **2006**, *30*, 913.

(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.

(3) Kondili, E.; Pantelides, C.C.; Sargent, R. A General Algorithm for Short-Term Scheduling of Batch Operations I. MILP Formulation. Comput. Chem. Eng. **1993**, *17*, 211.

(4) Pantelides, C.C. Unified Frameworks for the Optimal Process Planning and Scheduling. In *Proceedings of the Second Conference on Foundations of Computer Aided Operations*; Cache Publications: New York, 1994; pp 253.

(5) Castro, P.M.; Barbosa-Póvoa, A.P.; Matos, H.A.; Novais, A.Q. Simple Continuous-time Formulation for Short-Term Scheduling of Batch and Continuous Processes. *Ind. Eng. Chem. Res.* **2004**, *43*, 105.

(6) Maravelias, C.T.; Grossmann, I.E. New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.* **2003**, *42*, 3056.

(7) Sundaramoorthy, A.; Karimi, I.A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem. Eng. Sci.* **2005**, *60*, 2679.

(8) Castro, P.M.; Grossmann, I.E.; Novais, A.Q. Two New Continuous-Time Models for the Scheduling of Multistage Batch Plants with Sequence Dependent Changeovers. *Ind. Eng. Chem. Res.* **2006**, *45*, 6210.

(9) Janak, S.L.; Lin, X.; Floudas, C.A. Enhanced Continuous-Time Unit-Specific Event-Based Formulation for Short-Term Scheduling of Multipurpose Batch Processes: Resource Constraints and Mixed Storage Policies. *Ind. Eng. Chem. Res.* **2004**, *43*, 2516.

(10) Gupta, S.; Karimi, I.A. An Improved MILP Formulation for Scheduling Multiproduct, Multistage Batch Plants. Ind. Eng. Chem. Res. **2003**, *42*, 2365.

(11) Harjunkoski, I.; Grossmann, I. Decomposition Techniques for Multistage Scheduling Problems using Mixed-integer and Constraint Programming Methods. Comput. Chem. Eng. **2002**, *26*, 1533.

(12) Méndez, C.A.; Henning, G.P.; Cerdá, J. An MILP Continuous-time Approach to Short-term Scheduling of Resource Constrained Multistage Flowshop Batch Facilities. Comput. Chem. Eng. **2001**, *25*, 701.

(13) Hentenryck, P. V. Constraint Satisfaction in Logic Programming; MIT Press: Cambridge, MA, 1989.

(14) Erdirik-Dogan, M.; Grossmann, I.E. Optimal Production Planning Models for Parallel Batch Reactors with Sequence-dependent Changeovers. To appear in AIChE Journal, 2007.

(15) Birewar, D.B.; Grossmann I.E. Simultaneous Production Planning and Scheduling in Multiproduct Batch Plants. *Ind. Eng. Chem. Res.* **1990**, *29*, 570.

(16) Nemhauser, G.; Wolsey, L. Integer and Combinatorial Optimization; John Wiley & Sons: New York, 1988.

(17) Birewar, D. B.; Grossmann I. E. Efficient Optimization Algorithms for Zero-Wait Scheduling of Multiproduct Batch Plants. *Ind. Eng. Chem. Res.* **1989**, *28*, 1333-1345.

(18) Hentenryck, P. V. The OPL Optimization Programming Language; MIT Press: Cambridge, MA, 1999.

(19) Maravelias, C.T.; Grossmann, I.E. A hybrid MILP/CP Decomposition Approach for the Continuous-Time Scheduling of Multipurpose Batch Plants. Comput. Chem. Eng. **2004**, *28*, 1921.

List of Caption for Tables

Table 1. Overview of computational performance (CPU s) for maximum plant flexibility and revenue maximization [a].

Table 2. Overview of computational performance (CPU s) for maximum plant flexibility and makespan minimization.

Table 3. Detailed computational statistics for problem P3 (H=120)

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.

Table 5. Overview of computational performance (CPU s) for products in a single unit and revenue maximization.

Table 6. Overview of computational performance (CPU s) for products in a single unit and makespan minimization.

Table 7. Improvement in the value of the objective function for maximum plant flexibility.

List of Captions for Figures

Figure 1. Example of a single stage multiproduct plant

Figure 2. 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 3. 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.

Figure 4. Basic continuous-time grid (one for each equipment unit).

Figure 5. Calculation of the duration of aggregated tasks for explicit batching approach.

Figure 6. Explicit batching approach with products allocated to a single unit.

Figure 7. Illustration of global precedence sequencing variables approach showing the values of the most important model variables for this simple example.

Figure 8. Example of a cyclic schedule and location of the link to be broken

Figure 9. Part of the optimal schedule for example P4 (H=96).

Figure 10. Optimal schedule for P2 (makespan minimization).

Figure 11. Solution of E-D&G for P3 (H=144 h), first iteration.

Figure 12. Optimal schedule for P2 (makespan minimization) with all batches of a given product in the same unit.

List of Tables

Table 1. Overview of computational performance (CPU s) 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.

Table 2. Overview of computational performance (CPU s) 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[b]
|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[a]	1.14
nodes	171407	130436	18092900	351

[a] OM, BPS=1336.9. [b] Results for first iteration only.

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[a]	5		
	120	8[a]	5		
	144	8[a]	4	8[a]	5

[a] Finding the optimal solution requires even more event points

Table 5. Overview of computational performance (CPU s) for products in a single unit and revenue maximization.

			model			
problem	H (h)	optimum ($)	CT-EB-SU	CT-SV-SU	E-D&G-SU	CP-SU
P1	120	908.7	0.67	**0.16**	0.33	0.7
	144	1036.5	0.92	**0.12**	0.33	89.2
	168	1134.3	0.86	**0.06**	0.44	0.59
P2	144	1196.8	1.17	9.05	**0.67**	3600[e]
	168	1329.2	1.81	45	**0.75**	3600[f]
	192	1464.4	0.58	9.55	**0.55**	3600[g]
P3	120	1212.42	**24.7**	34695	4.61[b]	3600[h]
	144	1388.88	**8.5**	17156[a]	3.5[c]	3600[i]
	168	1544.88	**5.64**	64.3	1.28[d]	3600[j]
P4	96	1725.34	68.6	52.1	**7.95**	3600[k]
	120	1991.72	1294	**20**	34.3	3600[l]
	144	2141.03	18309	**28**	40.8	3600[m]

[a] SO=1368.9, although solver solved to optimality (special case). [b] UB=1238.48, SO=1140.02, NIT=3. [c] UB=1420.9, SO=1336.88, NIT=2. [d] SO=1440.88, NIT=2. [e] MRL, SO=1169.8. [f] MRL, SO=436.9. [g] MRL, SO=876.1. [h] MRL, SO=735.6. [i] MRL, SO=858.1. [j] MRL, SO=858. [k] MRL, SO=453.5. [l] MRL, SO=577.7. [m] MRL, SO=654.9

Table 6. Overview of computational performance (CPU s) for products in a single unit and makespan minimization.

		model			
problem	optimum (h)	CT-EB-SU	CT-SV-SU	E-D&G-SU	CP-SU
P1	186	0.2	0.06	0.54	**0.01**
P2	237	0.34	0.36	0.53	**0.05**
P3	168	4.88	11.7	1.86[a]	**0.64**
P4	151	184	1.95	68.1	**0.80**

[a] LB=166, SO=176, NIT=3.

Table 7. 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	-

List of Figures

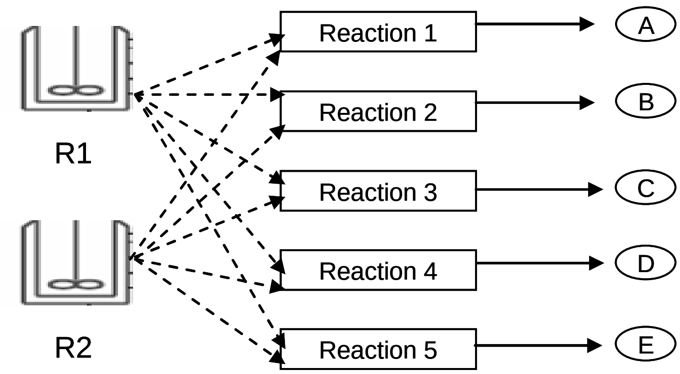

Figure 1. Example of a single stage multiproduct plant

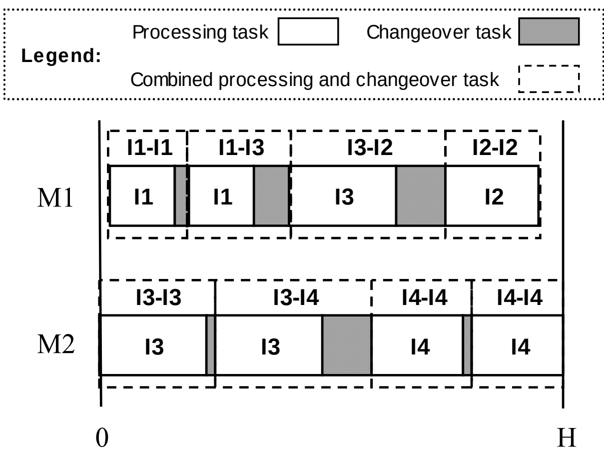

Figure 2. Implicit batching approach. Required number of batches for a particular product determined by the number of instances executed of the corresponding processing task (e.g. 2 batches for product I1, 1 for I2, 3 for I3 and 2 for I4).

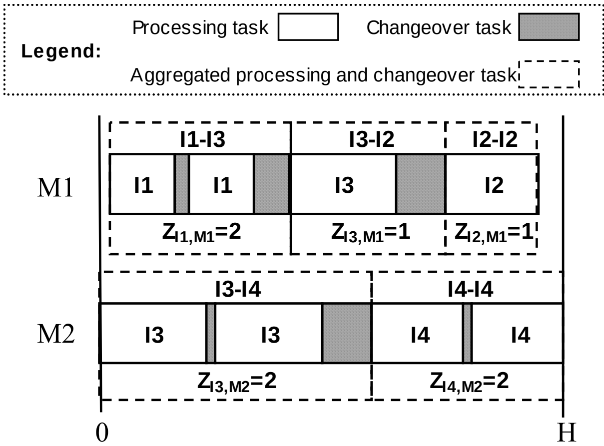

Figure 3. Explicit batching approach. Required number of batches for a particular product is defined as an integer model variable that affects the duration of the corresponding aggregated task.

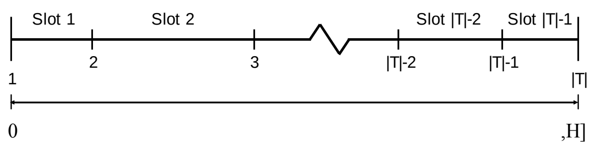

Figure 4. Basic continuous-time grid (one for each equipment unit).

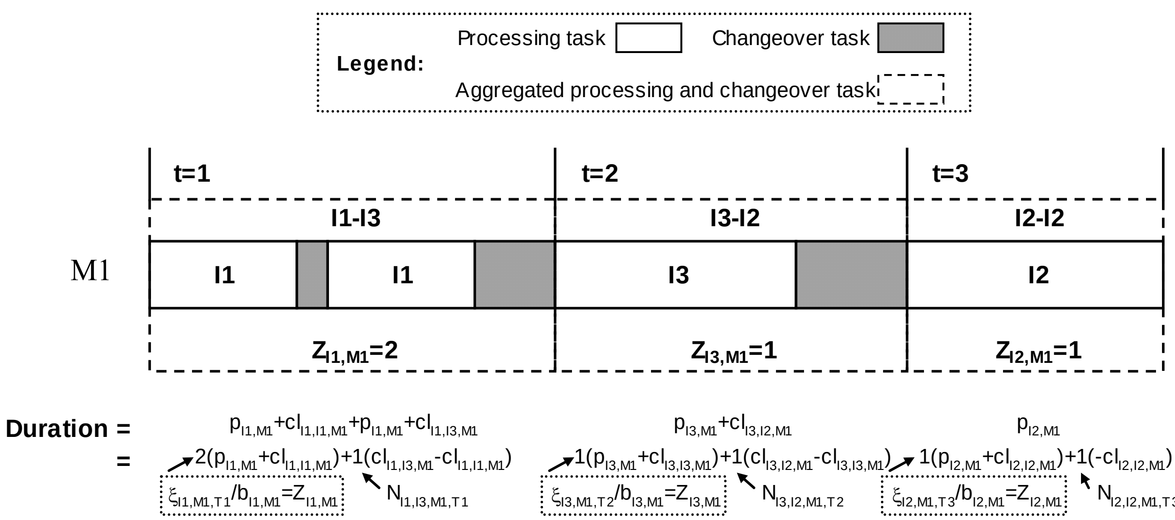

Figure 5. Calculation of the duration of aggregated tasks for explicit batching approach.

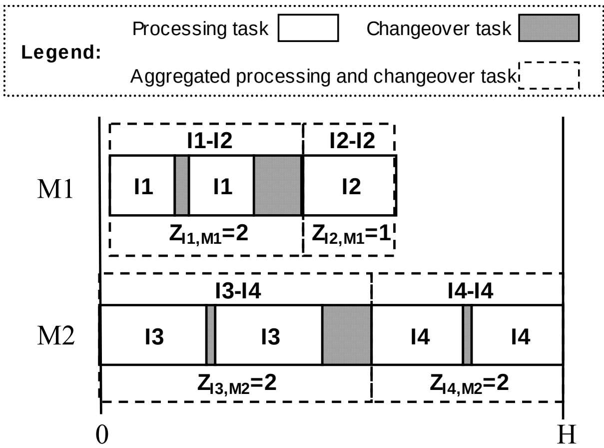

Figure 6. Explicit batching approach with products allocated to a single unit.

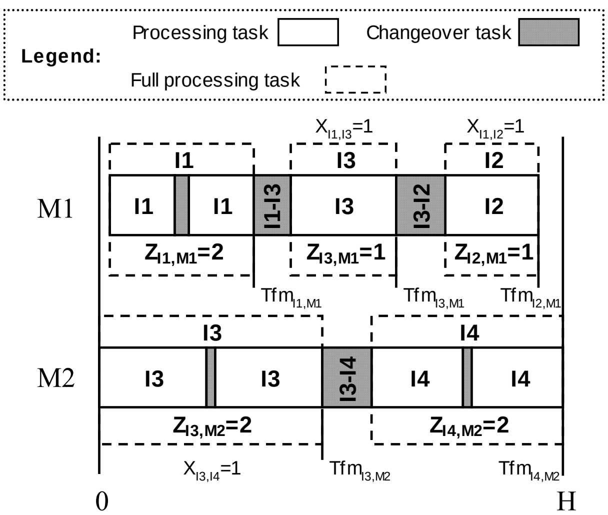

Figure 7. Illustration of global precedence sequencing variables approach showing the values of the most important model variables for this simple example.

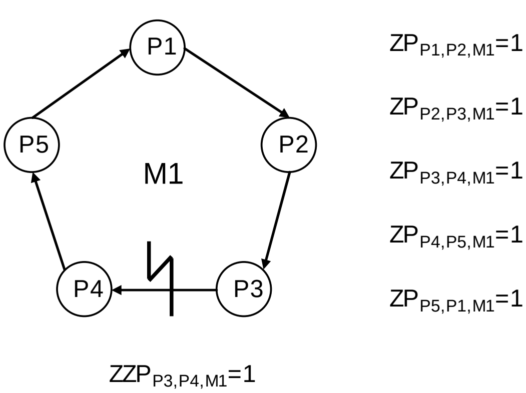

Figure 8. Example of a cyclic schedule and location of the link to be broken

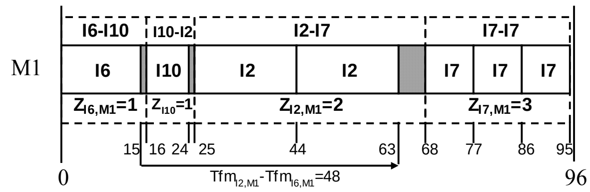

Figure 9. Part of the optimal schedule for example P4 (H=96).

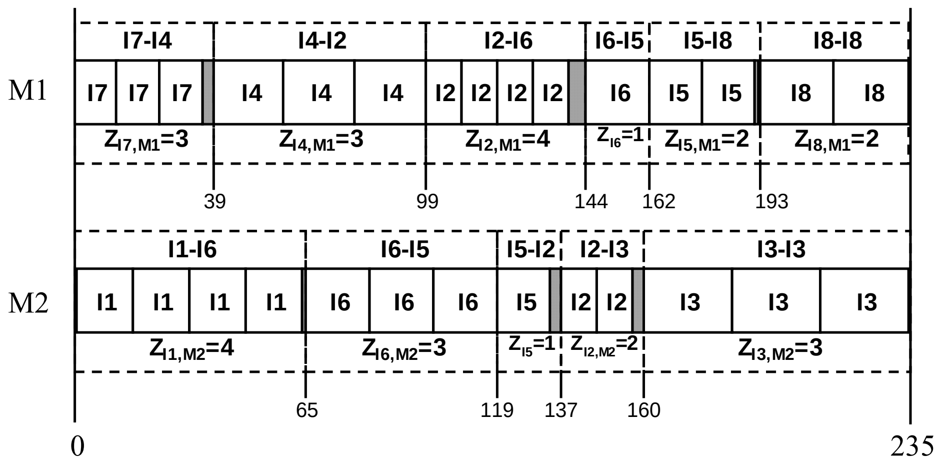

Figure 10. Optimal schedule for P2 (makespan minimization).

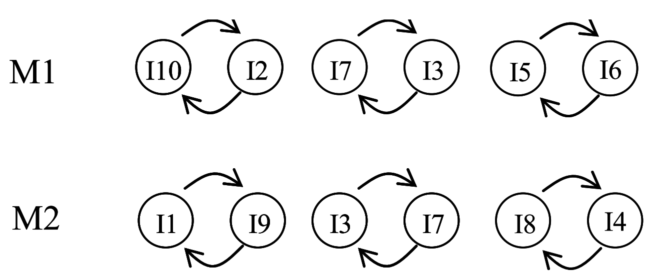

Figure 11. Solution of E-D&G for P3 (H=144 h), first iteration.

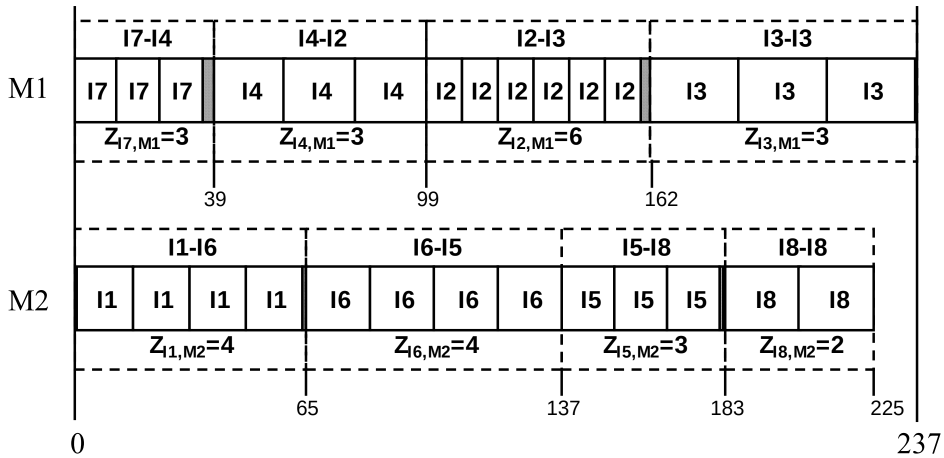

Figure 12. Optimal schedule for P2 (makespan minimization) with all batches of a given product in the same unit.