A Novel Continuous-Time MILP Formulation for Multipurpose Batch Plants. 1. Short-Term Scheduling

T. Majozi and (Frank) X. X. Zhu*

Department of Process Integration, University of Manchester Institute of Science and Technology, P.O. Box 88, Manchester M60 1QD, United Kingdom

In this paper, a new concept, the so-called state sequence network representation, is presented. This representation is based on states only, eliminating the use of tasks and units. Using this new representation as a basis, a novel continuous-time MILP formulation for short-term scheduling of multipurpose batch plants is developed. The presented formulation can readily be extended to intermediate due date scenarios. Also presented in this paper is a new model for the duration constraints that explores degrees of freedom that are intrinsic in batch operations. These are due to operator intervention, raw material purity, and different catalyst types. Time points are used to denote the use or production of a particular state. This formulation leads to the least number of binary variables and much better results when compared to other continuous-time formulations published in the literature. The reduced number of binary variables is a result of considering states only, thereby eliminating binary variables corresponding to tasks and units. This method has been applied to literature examples and industrial problems which show significant improvement in reducing the number of binary variables and, hence, CPU times. The last section of this paper introduces the concept of units aggregation in reducing the binary dimension of large-scale problems. This makes it possible for the method to solve large-scale industrial problems. In the second paper of this series (Zhu, X. X.; Majozi, T. *Ind. Eng. Chem. Res.* **2001**, *40*, 5621−5634), it will be demonstrated how this formulation is applied in the context of integrated planning and scheduling.

1. Introduction

The general scheduling problem entails the determination of the optimal sequence of events using available resources. The systematic formulation for this problem was initially given by Sparrow et al.[1] as an MINLP problem. In this formulation, the processing time for each batch is formulated as a function of the batch size, and the overall run length of each product over the time horizon of interest depends on the number of batches processed to achieve the production requirement. The lack of adequate solution procedures to guarantee global optimality of this formulation at the time stimulated extensive research in this area. Grossmann and Sargent[2] posed this problem as a geometric program and proved that the solution is global using the Kuhn−Tucker conditions. They also demonstrated that this problem could be solved as a relaxed subproblem by disregarding the discreteness of equipment sizes. Knopf et al.[3] considered a more general case with semicontinuous units. The problem was also posed as a geometric program but was solved in the convex primal form as suggested by Fattlar et al.[4] using reduced gradient code OPT.[5] Ravemark and Rippin[6] applied the same formulation as Sparrow et al.[1] did to multiproduct plants and used logarithmic transformation to ensure convexity of the MINLP.

There have also been a variety of heuristic procedures developed to handle specific types of multiproduct and multipurpose batch processes.[7−9] The main drawback of heuristic procedures is in its failure to guarantee global optimality. After the MINLP formulation and the heuristic-based procedures as mentioned above, Kondili et al.[10] developed an MILP formulation for the scheduling of mutipurpose batch plants using a state task network (STN) representation. In this formulation, the time horizon was discretized into uniform time intervals that coincided with the beginning and/or end of a particular task. The major drawback of this formulation is that accuracy increases with the number of time intervals. For some problems, this number can be very large, resulting in an explosive binary dimension of the problem which might not be adequately handled by existing procedures. To overcome this problem, Shah et al.[11] proposed solution procedures to reduce the computational effort of the MILPs resulting from this formulation. However, by virtue of its inherent time restriction, this procedure is bound to lead to suboptimal results. As a result, recent research in the area of batch process scheduling has focused on developing continuous-time approaches that do not restrict time distribution over the scheduling time horizon.

Zhang and Sargent[12,13] proposed a continuous-time formulation that is based on a concept of resource task network (RTN)[14] for batch process scheduling. Using this concept, the overall production facility is modeled as a collection of tasks and resources with the condition that some of the resources are consumed and some are formed. The RTN can be interpreted as an advanced form of the STN in the sense that, other than feeds, intermediates, and products, resources include energy, manpower, storage, and transportation facilities. Moreover, tasks are not merely operations wherein states are converted from one form to another but also include transportation, cleaning, and storage. In this formulation, the overall scheduling time horizon was demarcated into time intervals of unknown lengths, and the

* Corresponding author. E-mail: F.Zhu@umist.ac.uk.

10.1021/ie0005452 CCC: $20.00
Published on Web 11/14/2001

boundaries of each time interval coincided with the start and/or finish of a particular task(s). The overall mathematical formulation led to an MINLP which was then linearized to yield an MILP. However, as intrinsic in all linearization techniques, this exploded the dimension of the overall problem, rendering it difficult to handle by the use of conventional MILP solvers. This procedure is very similar to that proposed by Schilling and Pantelides,[15] which used the concept of time points and time slots. Their formulation yielded an MINLP, with nonliearities arising from duration and resource balance constraints. These were then linearized using the Glover[16] transformation to yield an MILP, which was solved using a novel branch and bound algorithm.[15] It is worth mentioning that all of these formulations base duration constraints on the batch size.

Mockus and Reklaitis[17] introduced the concept of time periods and events in formulating a continuous-time short-term scheduling model for multipurpose/multiproduct batch plants using the STN representation. In this formulation, events represent the beginning and/or end of a task(s), and time periods represent the time length between two consecutive events. The problem is formulated as an MINLP problem, with nonlinearities arising in the material balance, capacity, and resource allocation constraints, rendering the model nonconvex. This model was later linearized to yield a linearly constrained MINLP that was solved using a modified outer approximation algorithm. Global optimality could not be guaranteed, and poor computational results were obtained when applying this model to literature problems.

The common feature of all of the formulations described above is the assignment of a single binary variable to describe units (j) and tasks (i) at any point in time t, say y_{ijt}. This eventually leads to a large number of binary variables, with a dimension of $i \times j \times t$. Ierapetritou and Floudas[18] presented a continuous-time formulation for short-term scheduling of multipurpose batch processes that was aimed at addressing this problem. Their procedure used the STN representation and was based on the concept of event points. An event point represented the beginning and/or end of a task at a particular point within the time horizon. The optimal number of event points was determined using an iterative procedure. The main feature of this formulation is the separation of units and task events by assigning specific binary variables to units $yv(j,n)$ and tasks $wv(i,n)$. This leads to a much smaller number of binary variables for processes with several tasks and units, which makes this formulation more attractive for industrial applications. The overall scheduling model was formulated as an MILP model. However, this formulation has two major drawbacks. First, in situations where stages involve several units, it initially predicts a relatively larger number of binary variables, which are later reduced by exploiting the one-to-one correspondence between units and tasks. In a situation where more than one task takes place in a given unit, this reduction cannot be achieved. Moreover, for large problems the reduction of binary variables might not be straightforward. The second drawback is the modeling of duration constraints as a function of the batch size. This implicitly imposes restrictions on time and eventually results in suboptimal results.

It is these observations that inspired the development of the method presented in this paper. Using the procedure presented in this paper, one is assured of the least number of binary variables without using the variable reduction procedure. This is achieved by adopting a representation of batch processes, the so-called state sequence network (SSN). Only states are considered in this network, thereby eliminating the need for task and unit binary variables required by the STN representation. This makes it possible to define a single variable $y(s,p)$ throughout the formulation. Two examples from literature and an industrial case study demonstrate the application of this methodology.

It is worth noting that all of the procedures mentioned above do not consider intermediate orders of products; i.e., the demand has to be met at the end of the time horizon. Pinto and Grossmann,[19,20] Karimi and McDonald,[21] and Ierapetritou et al.[22] have addressed the problems with intermediate orders.

2. Problem Statement

The scheduling problem that is considered in this paper can be stated as follows. Given (i) the production recipe for each product, including mean processing times in each unit operation, (ii) the available units and their capacities, (iii) the maximum storage capacity for each material, and (iv) the time horizon of interest, determine (i) the optimal schedule for tasks within the time horizon of interest, (ii) the amount of material processed in each unit at any particular point in time within the time horizon, and (iii) the amount delivered to customers over the entire time horizon.

3. State Sequence Network (SSN)

Figure 1 illustrates the building blocks of the SSN. Figure 1a shows the transition from state s to state s'. This implies that there has to be a unit operation between these two because of the change in states. Figure 1b shows the mixing of different states to yield a new state, e.g., raw materials entering a reactor to yield a product. Figure 1c illustrates a splitting/separation unit with state s as the input and states s' and s'' as outputs.

To facilitate understanding of this concept, a literature problem[18] is presented in Figure 2.

For comparison, the SSN and STN corresponding to this example are shown in parts a and b of Figure 3, respectively.

The SSN for this example takes its simplest form, because there is no mixing or splitting of independent states. In the later examples, it will be demonstrated how the SSN is constructed if there is mixing or splitting of states. In the SSN representation, only states are considered while tasks and units are implicitly incorporated. This representation is developed by realizing that (i) the capacity of a unit in which a particular state is used or produced sets an upper limit on the amount of state used or produced by the corresponding task, (ii) the presence of a particular state in an operation corresponds to the existence of a corresponding task, and (iii) the usage of state s (input) corresponds to the production of state s' (output). Time points, as introduced by Schilling and Pantelides,[15] are used in this formulation. This is similar to the event point concept used by Ierapetritou and Floudas,[18] except that there are fundamental differences in the formulation of sequence and duration constraints. Figure 4 shows how the time points are defined in the time horizon of interest.

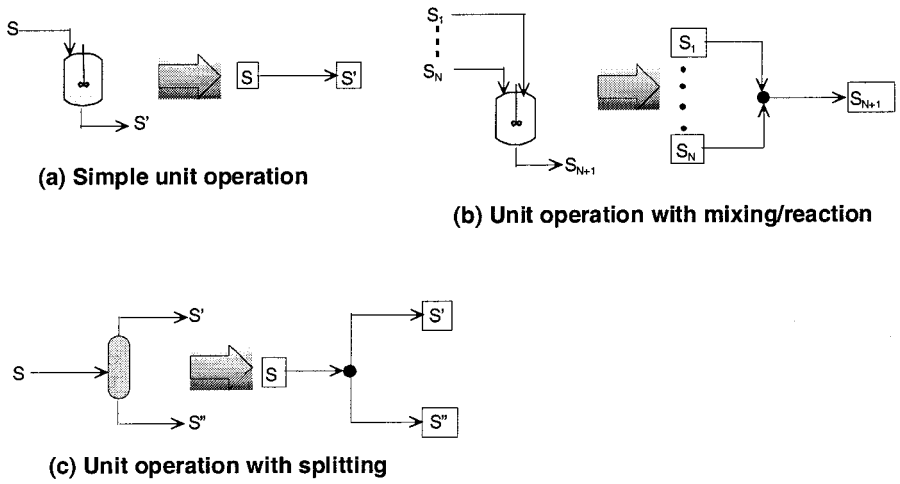


Figure 1. Building blocks for the SSN.

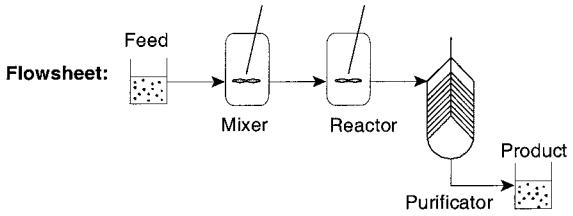


Figure 2. Plant flowsheet for the literature example.

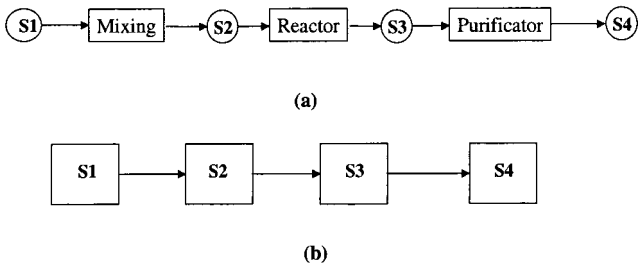


Figure 3. STN (a) and SSN (b) networks for the literature example.

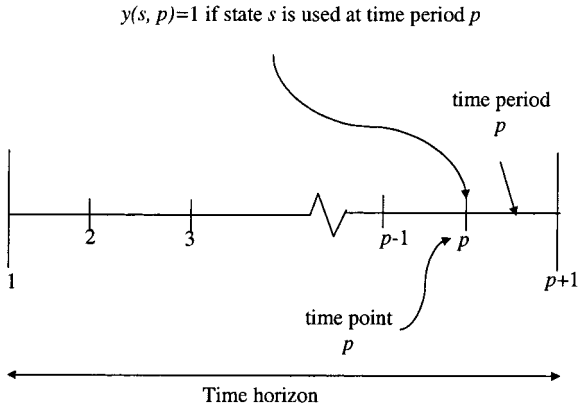


Figure 4. Description of time points.

In their formulation, Ierapetritou and Floudas[18] separated task and unit events by assigning corresponding binary variables to tasks [$wv(i,n)$] and units [$yv(j,n)$], respectively. This led to an overall number of binary variables of $n(N_i + N_j)$, where N_i and N_j are the numbers of tasks and units, respectively. As a result, the example in Figure 2 gave $n(3 + 3)$ binary variables, because it consists of three tasks and three units. Exploiting the one-to-one correspondence between tasks and units further reduced this number of binary variables to $3n$. This is only possible for processes in which only one task is performed in one unit. If a unit can perform more than one task, as in the examples given later in this paper, this reduction of binary variables is not possible. However, in the proposed new formulation, the only binary variables involved [$y(s,p)$] are those corresponding to states. $y(s,p)$ equals 1 if state s is used at time point p and zero otherwise. It should also be noted that the usage of state s at time point p corresponds to the production of state s' at time period $p + 1$, where s and s' are the input and output states of a particular operation, respectively. In the event where many states are used simultaneously, which is frequently encountered in batch reactors and blending operations, only one state is assigned the binary variable $y(s,p)$ because all of these states are used at the same time point. The state that has been assigned the binary variable is termed an effective state. Therefore, the resulting number of binary variables from this formulation is EP, where E is the number of effective states involved in the process and P is the total number of time points used in the formulation. In the given example (Figure 2), this formulation results in $3P$ binary variables, as shown in section 3. This is due to the fact that states s_1, s_2, and s_3 are the effective states. However, in using aggregation models that are only applicable to in-phase operation, as presented in the last section of the paper, the resulting number of binary variables is always SP, where S is the number of processing stages involved in the process. Each processing stage can have more than one processing unit (processor). It should, therefore, be noticed that the aggregation model is the simplification of the general model using the effective states. Also, the choice of effective states is not unique. However, once the choice of effective states has been made, it should remain consistent throughout the formulation.

Another distinct feature of the presentation made in this paper is the exploration of the degrees of freedom that are intrinsic in batch chemical processes. This is achieved by formulating the duration constraints such that batch time is not dependent on the batch size but influenced by external factors such as catalyst type, raw material purity, and human intervention. Most of the formulations presented in the literature have overlooked this feature and are strictly associated the variation of

processing time to variation in batch sizes, which is certainly not a true reflection of what happens in batch operations. This also imposes restrictions on time, because it implicitly implies that the quantities processed in each unit operation are dependent on the length of the time horizon of interest. Nevertheless, a formulation of duration constraints which associates batch time to batch sizes is also presented in the new model in order to compare results to those appearing in the literature. Using this formulation of duration constraints, results similar to those obtained using recent methods were obtained with the least number of binary variables. Much better results were obtained by using the formulation of duration constraints that incorporates intrinsic degrees of freedom. This is due to the fact that this formulation does not impose restrictions on time. All of the results presented in this paper were obtained using GAMS 2.25/OSL in a 333 MHz AMD K6-2 processor, unless otherwise stated.

4. Mathematical Model

Sets

$S_{in,j} = \{s_{in,j} | s_{in,j}$ is an input state to unit $j\}$
$S_{out,j} = \{s_{out,j} | s_{out,j}$ is an output state to unit $j\}$
$S = \{s | s$ is any state$\} = S_{in,j} \cup S_{out,j}$
$S^*_{in,j} = \{s^*_{in,j} | s^*_{in,j}$ is an effective state$\} \subseteq S_{in,j}$
$P = \{p | p$ is a time point$\}$
$J = \{j | j$ is a unit$\}$

Variables

$t_p(s,p)$ = time at which state s is produced at time point p, $s \in S_{out,j}$
$t_u(s,p)$ = time at which state s is used at time point p, $s \in S_{in,j}$
$q_s(s,p)$ = amount of state s stored at time point p
$m_p(s,p)$ = amount of state s produced at time point p, $s \in S_{out,j}$
$m_u(s,p)$ = amount of state s used at time point p, $s \in S_{in,j}$
$y(s,p)$ = binary variable associated with usage of state s at time point p, $s \in S^*_{in,j}$
$t_a(s)$ = actual processing time for state s, $s \in S_{in,j}$
$d(s,p)$ = amount of state s delivered to customers at time point p, $s \in S_{out,j}$
$\rho^+(s), \rho^-(s)$ = slack variables for variation in processing time due to added degrees of freedom, $s \in S_{in,j}$

Parameters

V_j = capacity of a particular unit j
H = time horizon of interest
$\tau(s_{in,j},p)$ = mean processing time for a state
$Q_s^0(s)$ = initial amount of state s stored
$v(s_{in,j},p)$ = allowed percentage time variation for processing a state

Capacity Constraints.

$$V_j^{min} y(s^*_{in,j},p) \le \sum_{s \in S_{in,j}} m_u(s,p) \le V_j^{max} y(s^*_{in,j},p), \quad \forall\, j \in J,\ p \in P \tag{1}$$

This constraint implies that the total amount of all of the states consumed $[\sum m_u(s,p)]$ at time point p is limited by the capacity of the unit which consumes the states (V_j). The max and min superscripts denote the maximum and the minimum capacities. According to constraint (1), states will be consumed in a particular unit j if the corresponding effective state is used at time point p.

Material Balances.

$$\sum_{s \in S_{in,j}} m_u(s,p-1) = \sum_{s \in S_{out,j}} m_p(s,p), \quad \forall\, p \in P,\ j \in J \tag{2}$$

$$q_s(s,p_1) = Q_s^0(s) - m_u(s,p_1), \quad s \ne \text{product},\ p_1 = \text{starting point} \tag{3}$$

$$q_s(s,p) = q_s(s,p-1) - m_u(s,p), \quad s = \text{feed},\ \forall\, p \in P,\ p > p_1 \tag{4}$$

$$q_s(s,p) = q_s(s,p-1) + m_p(s,p) - m_u(s,p), \quad s \ne \text{product, feed},\ \forall\, p \in P,\ p > p_1 \tag{5}$$

$$q_s(s,p_1) = Q_s^0(s) - d(s,p_1), \quad s = \text{product},\ p_1 = \text{starting point} \tag{6}$$

$$q_s(s,p) = q_s(s,p-1) + m_p(s,p) - d(s,p), \quad s = \text{product, byproduct},\ \forall\, p \in P,\ p > p_1 \tag{7}$$

Constraint (2) is the material balance around a particular unit j. It implies that the sum of the masses for all of the input states used at time point $p-1$ should be equal to the sum of the masses for all of the output states produced at time point p. Constraint (3) states that the amount of state s stored at the first time point is the difference between the amount stored before the beginning of the process and that being utilized at the first time point. Constraint (4) only applies to the feed, because it is the state that is only used in the process. Constraint (5) only applies to intermediates, because they are both produced and used in the process. Constraints (6) and (7) only apply to products and byproducts, because they are the only states that have to be taken out of the process as shown by the terms $d(s,p)$.

Duration Constraints (Batch Time as a Function of Variable Batch Size). In this section the duration constraints are modeled as a function of the batch size, although this effect is not common in practical situations, as mentioned earlier. The following constraints show how this effect is modeled in the proposed approach using the SSN representation.

$$t_p(s_{out,j},p) = t_u(s^*_{in,j},p-1) + a(s^*_{in,j})\, y(s^*_{in,j},p-1) + b(s^*_{in,j}) \sum_{s_{in,j}} m_u(s,p-1), \quad \forall\, j \in J,\ p \in P,\ s_{out,j} \in s_{out,j} \tag{8}$$

$$b(s^*_{in,j}) = \frac{t^{max}(s^*_{in,j}) - t^{min}(s^*_{in,j})}{V_j^{max} - V_j^{min}}, \quad \forall\, j \in J \tag{9}$$

$$a(s^*_{in,j}) = \tau(s^*_{in,j})[1 - v(s^*_{in,j})] = t^{min}(s^*_{in,j}), \quad \forall\, j \in J \tag{10}$$

$$t^{max}(s^*_{in,j}) = \tau(s^*_{in,j})[1 + v(s^*_{in,j})], \quad \forall\, j \in J \tag{11}$$

Equation 10 denotes the minimum processing time for the effective state in the corresponding unit operation. This is, in essence, the minimum residence time of a batch within a unit operation. The parameter for variable batch time is defined by eq 9. This gives the amount of time required to process a unit amount of a batch corresponding to a particular effective state in a corresponding unit operation.

Duration Constraints (Exploring More Degrees of Freedom). In this section the duration constraints

are modeled such that they do not depend on the batch size. This is because, in practice, the processing time variation is not due to variable batch sizes but due to several other reasons. Among them are equipment design and performance, variable raw material purity, different catalyst types, and different operator response times; e.g., one operator might take an hour longer to open steam required for a chemical reaction than the other. All of these factors directly influence the reaction rate, which leads to variable processing times. Any batch operation model that does not consider these intrinsic degrees of freedom is bound to lead to suboptimal results. In this work, the variable processing times have been included in the formulation by using a linearization technique as shown below.

$$t_a(s^*_{in,j}) = \tau(s^*_{in,j}) + \rho^+(s^*_{in,j}) + \rho^-(s^*_{in,j}), \quad \forall\, j \in J \tag{12}$$

$$t_p(s_{out,j},p) = t_u(s^*_{in,j},p-1) + t_a(s^*_{in,j},p-1)y(s^*_{in,j},\, p-1), \quad \forall\, j \in J,\ p \in P,\ s_{out,j} \in S_{out,j} \tag{13}$$

Constraint (12) is the definition of the processing time for a particular batch corresponding to a particular effective state in a corresponding unit operation, where ρ^+ and ρ^- are slack variables to account for variation in processing times due to the degrees of freedom. The bounds on the slack variables are determined by variation in processing times. A positive slack variable implies that the actual processing time is longer than the mean processing time. This could be due to deteriorating equipment, using an old catalyst, poor quality raw materials, or a long operator response time. A negative slack variable means that the actual processing time is shorter than the mean processing time. This could be due to using new catalyst, raw materials of high purity, or efficient operator response time. Therefore, in this formulation, the variation in processing time is not associated with batch sizes. This will become apparent in the examples given later in this paper. Constraint (13) states that the elapsed time between usage of input state(s) at time point $p-1$ and production of output state(s) at time point p in unit j is equal to the processing time of the corresponding batch. This batch can only be processed if a corresponding effective state is used at time point $p-1$. However, eq 13 constitutes bilinear terms, which are linearized using the technique proposed by Glover[16] as follows. The derivation of the following equations is shown in the appendix. McComick[23] showed that this linearization maintains the convexity of the overall formulation, because these constraints form the concave and convex envelopes of the bilinear terms over the given bounds.

$$t_p(s_{out,j},p) = t_u(s^*_{in,j},p-1) + \tau(s^*_{in,j})\, y(s^*_{in,j},p-1) + \Gamma^+(s^*_{in,j},p-1) + \Gamma^-(s^*_{in,j},p-1), \quad \forall\, p \in P,\ j \in J \tag{14}$$

$$\rho^+(s^*_{in,j},p) - v(s^*_{in,j})\,\tau(s^*_{in,j})[1 - y(s^*_{in,j},p)] \le \Gamma^+(s^*_{in,j},p) \le \rho^+(s^*_{in,j},p), \quad \forall\, j \in J,\ p \in P \tag{15}$$

$$0 \le \Gamma^+(s^*_{in,j},p) \le v(s^*_{in,j})\,\tau(s^*_{in,j})\, y(s^*_{in,j},p), \quad \forall\, j \in J,\ p \in P \tag{16}$$

$$\rho^-(s^*_{in,j},p) \le \Gamma^-(s^*_{in,j},p) \le \rho^-(s^*_{in,j},p) + v(s^*_{in,j})\, t(s^*_{in,j})[1 - y(s^*_{in,j},p)], \quad \forall\, j \in J,\ p \in P \tag{17}$$

$$-v(s^*_{in,j})\,\tau(s^*_{in,j})\, y(s^*_{in,j},p) \le \Gamma^-(s^*_{in,j},p) \le 0, \quad \forall\, j \in J,\ p \in P \tag{18}$$

According to constraints (14)−(18), if state $s^*_{in,j}$ is used at time point p, i.e., $y(s^*_{in,j},p) = 1$, Γ^+ and Γ^- take the values of ρ^+ and ρ^-, respectively. Otherwise, these slack variables reduce to zero.

Sequence Constraints.

$$t_u(s^*_{in,j},p) \ge \sum_{s_{in,j}} \sum_{s_{out,j}} \sum_{p' \le p} [t_p(s_{out,j},p') - t_u(s_{in,j},p'-1)], \quad \forall\, j \in J,\ p \in P,\ s_{out,j} \in S_{out,j},\ s_{in,j} \in S_{in,j} \tag{19}$$

$$t_u(s^*_{in,j},p) \ge t_p(s_{out,j},p), \quad \forall\, j \in J,\ p \in P \tag{20}$$

$$t_u(s_{in,j},p) \ge t_p(s_{out,j'},p), \quad \forall\, j, j' \in J,\ p \in P,\ s_{out,j'} = s_{in,j} \tag{21}$$

Constraints (19) and (20) imply that state s can only be used in a particular unit, at any time point, after all of the previous states have been processed. Constraint (19) is only relevant in situations where more than one task can be conducted in one unit; otherwise, it is redundant in the presence of constraints (20) and (21). Constraint (21) stipulates that a state can only be processed at a particular time point p in a particular unit j after it has been produced from another unit j'. In recycling, j is the same as j'. It is worth noting that constraints (20) and (21) are only applicable to intermediates, because they are the only states that are both produced and used.

Time Horizon Constraints.

$$t_u(s_{in,j},p) \le H, \quad \forall\, s_{in,j} \in S_{in,j},\ p \in P,\ j \in J \tag{22}$$

$$t_p(s_{out,j},p) \le H, \quad \forall\, s_{out,j} \in S_{out,j},\ p \in P,\ j \in J \tag{23}$$

Constraints (22) and (23) respectively stipulate that the usage or production of state should be within the time horizon of interest.

Storage Constraints.

$$q_s(s,p) \le Q^{max}(s), \quad \forall s \in S,\ p \in P \tag{24}$$

Constraint (24) states that the amount of state s stored at each time point cannot exceed the maximum allowed.

Objective Function. The objective function for this formulation is the maximization of product throughput.

$$\text{Maximize} \sum_s \sum_p d(s,p), \quad s = \text{product},\ p \in P \tag{25}$$

5. Literature Examples

First Literature Example.[18] In this section, the above mathematical model is applied to a literature example shown in Figure 2. The SSN representation is given in Figure 3b. Table 1 gives data for this example.

Five time points were used. Using fewer time points leads to a suboptimal solution of 50, and using more time points did not improve the solution.

Capacity Constraints.

State s_1

$$m_u(s_1,p) \le 100y(s_1,p), \quad \forall\, p \in P$$

State s_2

$$m_u(s_2,p) \le 75y(s_2,p), \quad \forall\, p \in P$$

State s_3

$$m_u(s_3,p) \le 50y(s_3,p), \quad \forall\, p \in P$$

Material Balances.

Unit mass balances: Mixer

$$m_p(s_2,p) = m_u(s_1,p-1), \quad \forall\, p \in P$$

Unit mass balances: Reactor

$$m_p(s_3,p) = m_u(s_2,p-1), \quad \forall\, p \in P$$

Unit mass balances: Purificator

$$m_p(s_4,p) = m_u(s_3,p-1), \quad \forall\, p \in P$$

State s_1

$$q_s(s_1,p_1) = Q_s^0(s_1) - m_u(s_1,p_1)$$

$$q_s(s_1,p) = q_s(s_1,p-1) - m_u(s_1,p), \quad \forall\, p \in P,\ p > p_1$$

State s_2

$$q_s(s_2,p_1) = Q_s^0(s_2) - m_u(s_2,p_1)$$

$$q_s(s_2,p) = q_s(s_2,p-1) + m_p(s_2,p) - m_u(s_1,p), \quad \forall\, p \in P,\ p > p_1$$

State s_3

$$q_s(s_3,p_1) = Q_s^0(s_3) - m_u(s_3,p_1)$$

$$q_s(s_3,p) = q_s(s_3,p-1) + m_p(s_3,p) - m_u(s_3,p), \quad \forall\, p \in P,\ p > p_1$$

State s_4

$$q_s(s_4,p_1) = Q_s^0(s_4) - d(s_4,p_1)$$

$$q_s(s_4,p) = q_s(s_4,p-1) + m_p(s_4,p) - d(s_4,p), \quad \forall\, p \in P,\ p > p_1$$

Duration Constraints (Batch Time as a Function of the Batch Size).

$$t_p(s_2,p) = t_u(s_1,p-1) + 3y(s_1,p-1) + 0.03m_u(s_1,p-1), \quad \forall\, p \in P$$

$$t_p(s_3,p) = t_u(s_2,p-1) + 2y(s_2,p-1) + 0.0267m_u(s_2,p-1), \quad \forall\, p \in P$$

$$t_p(s_4,p) = t_u(s_3,p-1) + y(s_3,p-1) + 0.02m_u(s_3,p-1), \quad \forall\, p \in P$$

The coefficients for $m_u(s,p)$ and $y(s,p)$ are defined by constraints (9) and (10), respectively.

Table 1. Data for the Literature Example[18]

unit	capacity	suitability	mean processing time (h)
1	100	mixing	4.5
2	75	reaction	3.0
3	50	purification	1.5

state	storage capacity	initial amount	price
1	unlimited	unlimited	0.0
2	100	0.0	0.0
3	100	0.0	0.0
4	unlimited	0.0	1.0

Duration Constraints (Exploring More Degrees of Freedom).

$$t_p(s_2,p) = t_u(s_1,p-1) + 4.5y(s_1,p-1) + \Gamma^+(s_1,p-1) + \Gamma^-(s_1,p-1), \quad \forall\, p \in P$$

$$t_p(s_3,p) = t_u(s_2,p-1) + 3.0y(s_2,p-1) + \Gamma^+(s_2,p-1) + \Gamma^-(s_2,p-1), \quad \forall\, p \in P$$

$$t_p(s_4,p) = t_u(s_3,p-1) + 1.5y(s_3,p-1) + \Gamma^+(s_3,p-1) + \Gamma^-(s_3,p-1), \quad \forall\, p \in P$$

The slack variables Γ^+ and Γ^- are defined by constraints (15)−(18) with $v(s) = 1/3$ because the time variation allowed in this example is 33%.

Sequence Constraints. Because each unit can only perform one task, constraint (19) is redundant.

States s_1 and s_2

$$t_u(s_1,p) \ge t_p(s_2,p), \quad \forall\, p \in P$$

States s_2 and s_3

$$t_u(s_2,p) \ge t_p(s_2,p), \quad \forall\, p \in P$$

$$t_u(s_2,p) \ge t_p(s_3,p), \quad \forall\, p \in P$$

States s_3 and s_4

$$t_u(s_3,p) \ge t_p(s_3,p), \quad \forall\, p \in P$$

$$t_u(s_3,p) \ge t_p(s_4,p), \quad \forall\, p \in P$$

These correspond to constraints (20) and (21) given in the mathematical model.

Time Horizon Constraints.

State s_1

$$t_u(s_1,p) \le 12, \quad \forall\, p \in P$$

State s_2

$$t_u(s_2,p) \le 12, \quad t_u(s_2,p) \le 12\ , \quad \forall\, p \in P$$

State s_3

$$t_u(s_3,p) \le 12, \quad t_u(s_3,p) \le 12, \quad \forall\, p \in P$$

State s_4

$$t_p(s_4,p) \le 12, \quad \forall\, p \in P$$

Storage Constraints.

$$q_s(s_2,p) \le 100, \quad \forall\, p \in P$$

$$q_s(s_3,p) \le 100, \quad \forall\, p \in P$$

Objective Function.

$$\text{Maximize } R = \sum_p d(s_4,p), \quad \forall\, p \in P$$

In this formulation, the only binary variables involved are $y(s_1,p)$, $y(s_2,p)$, and $y(s_3,p)$ corresponding to states s_1, s_2, and s_3, respectively. Therefore, the total number of binary variables is $3p$.

Computational Results. Five time points and a 12 h time horizon were used for this example. The results from this proposed method and from the methods proposed by Ierapetritou and Floudas,[18] Zhang,[13] and Schilling and Pantelides[15] are shown in Table 2.

The results in the second and third columns were obtained using GAMS 2.25/OSL in a 333 MHz AMD K6-2 processor. The results in the fourth column were reproduced based on the method by Ierapetritou and Floudas[18] using the same solver in the same computer as above, while those in the fifth and sixth columns were taken directly from Ierapetritou and Floudas.[18] The results appearing in the third, fourth, fifth, and sixth columns of Table 2 were obtained by modeling the duration constraints as a function of the batch size variation. The proposed approach gives an objective value of 71.473 and requires only 15 binary variables, compared to 48 and 46 binary variables required in approaches proposed by Zhang[13] and Schilling and Pantelides,[15] respectively. The formulation by Ierapetritou and Floudas[18] initially consisted of 30 binary variables that were later reduced to 15 by exploiting one-to-one correspondence of units and tasks. It is also worth noting that the proposed formulation requires the least number of constraints (i.e., 159) and shortest CPU time (0.391 s). Using four time points led to an objective value of 50, and using more time points did not improve the objective function. The Gantt chart for the proposed approach is shown in Figure 5.

In accordance with the formulation, Figure 5 shows that the smaller the batch, the shorter the processing time in the same unit. The objective value of 71.518 reported by Ierapetritou and Floudas[18] was due to rounding off of the parameter for the batch size variation. The same value is obtained in this formulation if a similar rounding off is used. Table 3 shows the values of the binary variables at different time points.

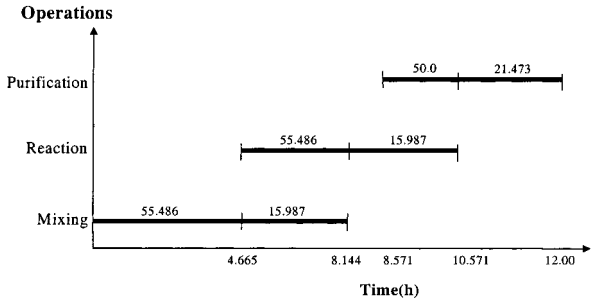


Figure 5. Gantt chart for the literature example: proposed approach 2.

Because the last time point, i.e., p_5, corresponds to the end of the time horizon, all of the binary variables are zero because no state can be used.

The results given in the second column of Table 2 were obtained by modeling duration constraints independent of the batch size as aforementioned. This formulation leads to only 15 binary variables and gives revenue of 100 with an integrality gap of 0%. This is 39.82% more than the revenue obtained by modeling duration constraints as a function of the batch size. A zero integrality gap implies that the relaxed objective is feasible in the given problem. This is also a reflection of best equipment utilization. The larger number of constraints in the new approach, i.e., 243 instead of 159, is due to the linearization of bilinear terms that arise in the duration constraints. It should be noted, however, that the increase in the number of equations does not increase the number of binary variables. Therefore, the difficulty in solving the problem is not increased, as indicated by the relatively shorter CPU time, viz. 0.332 s compared to 0.391 s. The Gantt chart for this example is shown in Figure 6.

It should be noted in Figure 6 that the batch size has no effect on the processing time. The batches in the reaction unit take the same length of processing time, i.e., 3 h, although they are 75 and 25 capacity units, respectively. In this problem, the slack variables catering for operation uncertainties were zero. This implies that, within a 12 h time horizon, the mean processing times set an upper bound on the duration of each operation. Both proposed formulations converged within a tolerance of 10^{-6}.

Comparison of the results from Figures 5 and 6 shows that using the batch size in modeling the duration constraints leads to suboptimal results, due to inherent time restriction. In Figure 5, the batch that is in the purification stage between 10.57 and 12.00 h had to be 21.473 capacity units in order to fit the time horizon. A batch of bigger capacity would require the extension of the time horizon. However, in practice, any

Table 2. Results for the First Literature Example[a]

			formulation of		
	proposed approach 1	proposed approach 2	Ierapetritou and Floudas[18]	Zhang[13]	Schilling and Pantelides[15]
NTP	5	5	5	7	6
NC	243	159	177	263	220
NV	161	101	101	187	157
NB	15	15	30 (15)	48	46
MILP solution	100	71.473	71.518	71.45	71.47
relaxed objective	100	100	100	149.99	170.79
CPU time (s)	0.332	0.391	0.602	21.9	N/A

[a] NTP = number of time points; NC = number of constraints; NV = total number of variables; NB = number of binary variables.

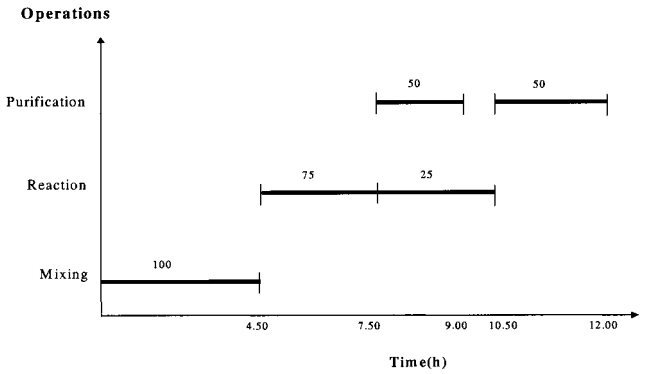


Figure 6. Gantt chart for the literature example: proposed approach 1.

Table 3. Values of the Binary Variables at Different Time Points for the Literature Example

$y(s,p)$	p_1	p_2	p_3	p_4	p_5
s_1	1	1	0	0	0
s_2	0	1	1	0	0
s_3	0	0	1	1	0

batch within the capacity limits of the purification unit, i.e., 50, could be processed within this time frame if the processing conditions allow. Temperature, pressure, catalyst type, operator efficiency, etc., determine these conditions. Figure 6 substantiates the foregoing statement. In Figure 6, the last batch processed in the purification stage is 50 capacity units, and the actual processing time is 1.5 h. This is equal to the mean processing time allowed in this processing step. The mean processing time is usually based on experience.

Second Literature Example.[18] The STN and SSN for the second literature example are given in parts a and b of Figure 7, respectively.

The data for this example appear in Table 4.

In this example, there are two reactors in which reactions 1−3 can be performed. Equal mean reaction times for the different reactions in each of the reactors imply similar performances for the reactors. The overall process consists of four units, i.e., heater, reactor 1, reactor 2, and separator. To handle the usage of feed C in two distinct reactions, i.e., reactions 1 and 3, different states were assigned to each of the streams of feed C, i.e., states s_3 and s_4, respectively. Constraints that exhibit structures similar to those presented in example 1 are not repeated.

Choice of Effective States. Because this problem involves more than one state entering some units, i.e., reactors 1 and 2, it is necessary to choose effective states before proceeding.

$$s^*_{\text{in},j} = \{s_1, s_{2,\text{in},j}, s_{6,\text{in},j}, s_9\}, \quad j = 2, 3$$

This implies there are eight effective states, leading to $8P$ binary variables. It should be mentioned that the only requirement for the choice of the effective states is that only one of the input states that are used simultaneously in a particular unit should be chosen. Therefore, the set given above is not unique. However, this does not change the binary dimension of the problem.

Capacity Constraints.

Reaction 1

Reactor 1 ($j = 2$):

$$m_u(s_{2,\text{in},2},p) + m_u(s_{3,\text{in},2},p) \le 50y(s_{2,\text{in},2},p), \quad \forall\, p \in P$$

Reactor 2 ($j = 3$):

$$m_u(s_{2,\text{in},3},p) + m_u(s_{3,\text{in},3},p) \le 80y(s_{2,\text{in},3},p), \quad \forall\, p \in P$$

Reaction 2

Reactor 1 ($j = 2$):

$$m_u(s_{6,\text{in},2},p) + m_u(s_{5,\text{in},2},p) \le 50y(s_{6,\text{in},2},p), \quad \forall\, p \in P$$

Reactor 2 ($j = 3$):

$$m_u(s_{6,\text{in},3},p) + m_u(s_{5,\text{in},3},p) \le 80y(s_{6,\text{in},3},p), \quad \forall\, p \in P$$

Reaction 3

Reactor 1 ($j = 2$):

$$m_u(s_{8,\text{in},2},p) + m_u(s_{4,\text{in},2},p) \le 50y(s_{8,\text{in},2},p), \quad \forall\, p \in P$$

Reactor 2 ($j = 3$):

$$m_u(s_{8,\text{in},3},p) + m_u(s_{4,\text{in},3},p) \le 80y(s_{8,\text{in},3},p), \quad \forall\, p \in P$$

Material Balances.

Unit mass balances: reaction 1

$$m_u(s_{2,\text{in},j},p-1) + m_u(s_{3,\text{in},j},p-1) = m_p(s_{6,\text{out},j},p), \quad \forall\, p \in P, j = 2, 3$$

Unit mass balances: reaction 2

$$m_u(s_{6,\text{in},j},p-1) + m_u(s_{5,\text{in},j},p-1) = \frac{10}{4} m_p(s_{7,\text{out},j},p), \quad \forall\, p \in P, j = 2, 3$$

The coefficient for $m_p(s_{7,\text{out},j},p)$ is derived from the stoichiometric values given in the SSN.

Unit mass balances: reaction 3

$$m_u(s_{8,\text{in},j},p-1) + m_u(s_{4,\text{in},j},p-1) = m_p(s_{9,\text{out},j},p), \quad \forall\, p \in P, j = 2, 3$$

Unit mass balances: separation

$$m_u(s_{9,\text{in},4},p-1) = \frac{10}{9} m_p(s_{10,\text{out},4},p), \quad \forall\, p \in P$$

Stoichiometric constraints

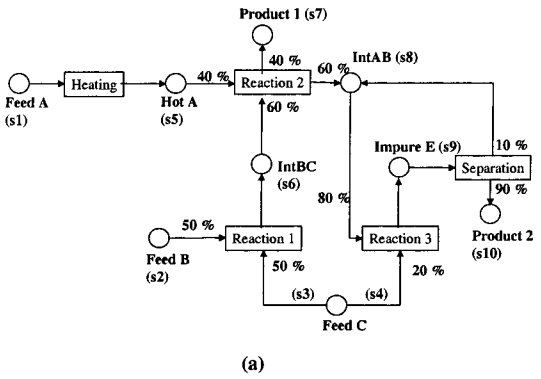
$$m_u(s_{3,\text{in},j},p) = m_u(s_{2,\text{in},j},p), \quad \forall\, p \in P, j = 2, 3$$

$$m_u(s_{6,\text{in},j},p) = \frac{60}{40} m_u(s_{5,\text{in},j},p), \quad \forall\, p \in P, j = 2, 3$$

$$m_u(s_{8,\text{in},j},p) = \frac{80}{20} m_u(s_{4,\text{in},j},p), \quad \forall\, p \in P, j = 2, 3$$

These stoichiometric constraints are derived from the data given in the SSN.

Sequence Constraints. Because reactors 1 and 2 can conduct reactions 1−3, constraint (19) is necessary as mentioned earlier. However, this constraint is not necessary for the heater and separator. The following



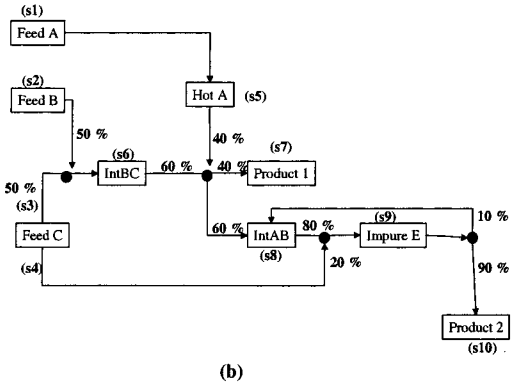


Figure 7. STN (a) and SSN (b) for the second literature example.

are the constraints corresponding to constraint (19) for this problem.

$$t_u(s^*_{in,j},p) \geq \sum_{p'=p_2}^{p} t_p(s_{6,out,j},p') - t_u(s_{2,in,j},p'-1) + t_p(s_{8,out,j},p') - t_u(s_{6,in,j},p'-1) + t_p(s_{9,out,j},p') - t_u(s_{8,in,j},p-1), \quad \forall\, p \in P,\ s^*_{in,j} = s_{2,in,j},\ s_{6,in,j},\ s_{8,in,j},\ j = 2, 3$$

The following constraints correspond to constraints (20) and (21) in the mathematical model.

States s_2, s_6, and s_8

$$t_u(s_{2,in,j},p) \geq t_p(s_{6,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{2,in,j},p) \geq t_p(s_{8,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{2,in,j},p) \geq t_p(s_{9,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{6,in,j},p) \geq t_p(s_{6,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{6,in,j},p) \geq t_p(s_{8,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{6,in,j},p) \geq t_p(s_{9,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{8,in,j},p) \geq t_p(s_{8,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{8,in,j},p) \geq t_p(s_{6,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{8,in,j},p) \geq t_p(s_{9,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

This set of constraints ensures that each reaction commences after the completion of the other reactions, because they share the same units. The following constraints are also required to ensure that raw materials and products of each reaction are used and produced at the same point in time, respectively.

Reacion 1

$$t_u(s_{2,in,j},p) = t_u(s_{3,in,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

Reaction 2

$$t_u(s_{6,in,j},p) = t_u(s_{5,in,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

$$t_u(s_{7,out,j},p) = t_p(s_{8,out,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

Table 4. Data for the Second Literature Example

unit	capacity	suitability	mean processing time (h)
heater	100	heating	1.0
reactor 1	50	reactions 1−3	2.0, 2.0, 1.0
reactor 2	80	reaction 1−3	2.0, 2.0, 1.0
still	200	separation	1 for product 2, 2 for intAB

state	storage capacity	initial amount	price
feed A	unlimited	unlimited	0.0
feed B	unlimited	unlimited	0.0
feed C	unlimited	unlimited	0.0
hot A	100	0.0	0.0
intAB	200	0.0	0.0
intBC	150	0.0	0.0
impure E	200	0.0	0.0
product 1	unlimited	0.0	10.0
product 2	unlimited	0.0	10.0

Reaction 3

$$t_u(s_{8,in,j},p) = t_u(s_{4,in,j},p), \quad \forall\, p \in P,\ j = 2, 3$$

The objective function for this formulation is the maximization of revenue for products 1 and 2.

Computational Results. Table 5 gives the computational results for the second literature example.

The results appearing in the second and third columns were obtained by the proposed method. The other results were taken directly from the literature[18] and were obtained using solvers different from the one used in the proposed method. Using five time points over an 8 h time horizon and modeling duration constraints as a function of the batch size gave the objective value of 1513.35. Using more time points did not increase the objective value but only increased the relaxed objective value from 1735.53 to 2054.68. It is worth noting that the proposed formulation requires only 40 binary variables compared to 48, 147, and 130 required in the formulations by Ierapetritou and Floudas,[18] Zhang,[13] and Schilling and Pantelides,[15] respectively. It should also be noted that, in this problem, the proposed formulation leads to a much smaller integrality gap, i.e., 12.8%, compared to the formulations by Ierapetritou and Floudas (24.24%), Zhang (33.7%), and Schilling and Pantelides (46.53%). Convergence was reached within a tolerance of 10^{-6}. The Gantt chart corresponding to this formulation is shown in Figure 8.

Table 6 shows the values of the binary variables at different time points.

It is worth noting that all of the binary variables are zero at the end of the time horizon, i.e., p_5, implying that no state can be used at this point.

Table 5. Results for the Second Literature Example[a]

	proposed approach 1	proposed approach 2	formulation of Ierapetritou and Floudas[18]	formulation of Zhang[13]	formulation of Schilling and Pantelides[15]
NTP	6	5	6	7	6
NC	1064	643	465	741	587
NV	696	420	310	497	386
NB	48	40	48	147	130
MILP solution	3315.00	1513.35	1503.18	1497.69	1488.05
relaxed objective	3461.25	1735.53	1984.17	2258.71	2783.14
CPU time (s)	12.915	4.23	2.91	1027.5	N/A

[a] NTP = number of time points; NC = number of constraints; NV = total number of variables; NB = number of binary variables.

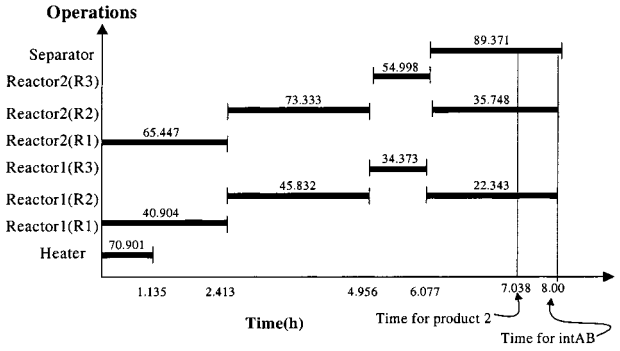


Figure 8. Gantt chart for the second literature example using proposed approach 2.

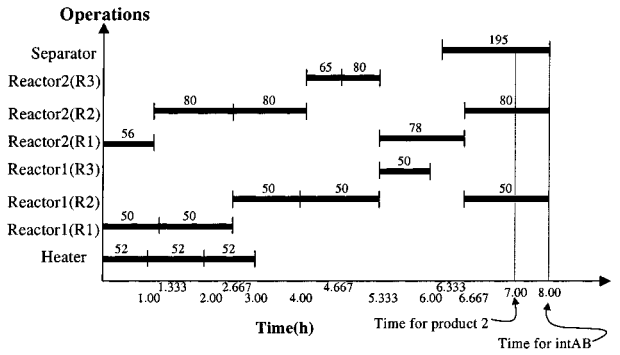


Figure 9. Gantt chart for the second literature example using proposed approach 1.

Table 6. Values for the Binary Variable at Different Time Points for the Second Example

$y(s,p)$	p_1	p_2	p_3	p_4	p_5
s_1	1	0	0	0	0
$s_{2,in,2}$	1	0	0	0	0
$s_{2,in,3}$	1	0	0	0	0
$s_{6,in,2}$	0	0	0	1	0
$s_{6,in,3}$	0	0	0	1	0
$s_{8,in,2}$	0	1	1	0	0
$s_{8,in,3}$	0	1	1	0	0
s_2	0	0	0	0	0

The results shown in the second column of Table 5 were obtained by modeling duration constraints such that they are only influenced by the degrees of freedom that are intrinsic in batch operations rather than the batch sizes. Six time points and an 8 h time horizon were used in the proposed formulation, which leads to an objective value of 3315.00. This formulation required 48 binary variables, 1064 constraints, and 383 continuous variables. The solution was obtained in 12.915 CPU s. The Gantt chart resulting from this formulation is shown in Figure 9.

In both Figures 8 and 9, product 2 is produced almost 1 h prior to the completion of separation in accordance with problem specification in Table 4.

6. Industrial Case Study

Figure 10 is the flowsheet for the industrial case study used to illustrate the application of the proposed method.

The SSN and STN are given in Figure 11.

The process involved consists of five consecutive steps. The first step involves a reaction which forms an arsenate salt. This reaction requires two raw materials, raw 3 and raw 4, and can be conducted in either reactor R1 or R2. The arsenate salt from the first step is then transferred to either reactor R3 or R4, wherein two consecutive reactions take place. The first of these reactions is aimed at converting the arsenate salt to a disodium salt using raw material 1 (raw 1). The disodium salt is then reacted further to form the monosodium salt using raw material 2 (raw 2). The monosodium salt solution is then transferred to the settling step in order to remove the solid byproduct. Settling can be conducted in any of the three settlers, i.e., SE1, SE2, or SE3. The solid byproduct is dispensed with as waste, and the remaining monosodium salt solution is transferred to the final step. This step consists of two evaporators, EV1 and EV2, which remove the excess amount of water from the monosodium solution. Evaporated water is removed as effluent, and the monosodium salt (product) is taken to storage. States s_1 and s_9 in the SSN represent raw 3 and raw 4, respectively. States s_{10} and s_2 represent raw 1 and the arsenate salt, while states s_{11} and s_3 represent raw 2 and the disodium salt, respectively. State s_4 is the monosodium solution that is transferred to the settlers to form states s_8 (solid byproduct) and s_5 (remaining monosodium solution). State s_5 is separated into states s_7 (water) and s_6 (product). Table 7 shows the data for the case study.

The stoichiometric data are included in order to perform material balances in each unit operation. The second column of the stoichiometric data shows the amount of raw material required (tons) per unit mass (tons) of the overall output, i.e., $s_6 + s_7 + s_8$. The third column shows the ratio of each byproduct (s_7 and s_8) to product (s_6) in ton/ton product. The objective function is the maximization of product (s_6) output. A 20% variation in processing times was assumed.

Computational Results. The results appearing in the third column of Table 8 were obtained by modeling the duration constraints such that the batch size affects the processing time.

These results were obtained using seven time points and a 12 h time horizon. Using more time points only increased the integrality gap but did not improve the results. Using less time points gave an objective value

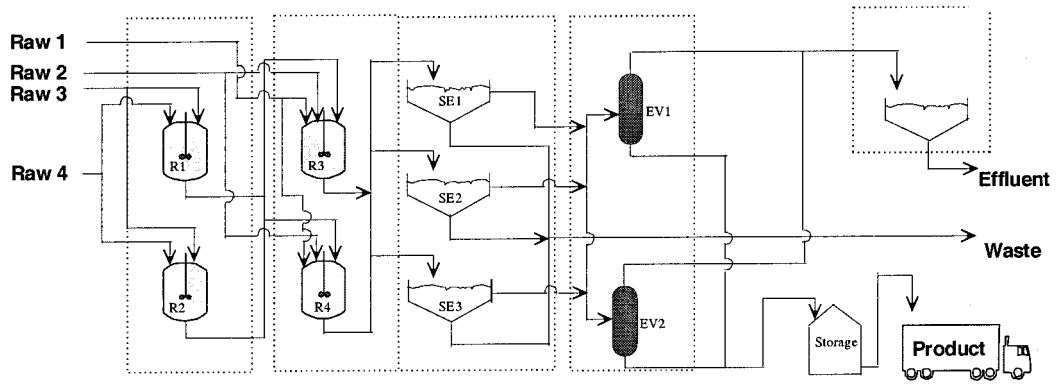


Figure 10. Flowsheet for the industrial case study.

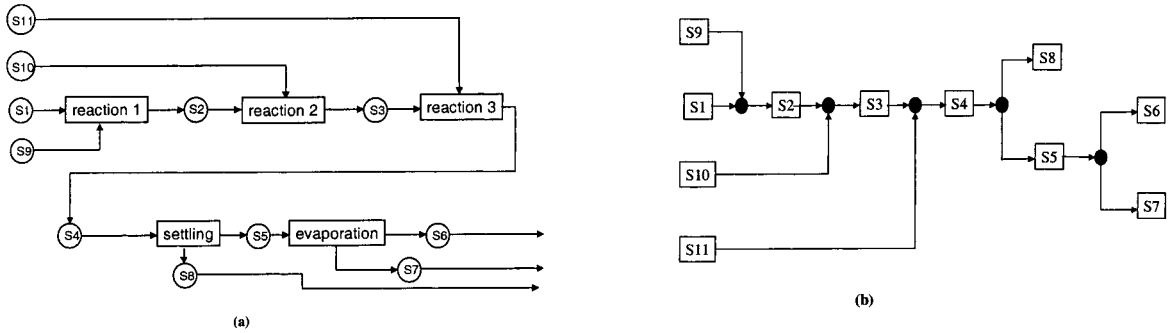


Figure 11. STN (a) and SSN (b) for the industrial case study.

Table 7. Scheduling Data for the Industrial Case Study

unit	capacity	suitability	mean processing time (h)
R1	10	reaction 1	2
R2	10	reaction 1	2
R3	10	reaction 2, reaction 3	3, 1
R4	10	reaction 2, reaction 3	3, 1
SE1	10	settling	1
SE2	10	settling	1
SE3	10	settling	1
EV1	10	evaporation	3
EV2	10	evaporation	3

state	storage capacity	initial amount
S1	unlimited	unlimited
S2	100	0
S3	100	0
S4	100	0
S5	100	0
S6	100	0
S7	100	0
S8	100	0
S9	unlimited	unlimited
S10	unlimited	unlimited
S11	unlimited	unlimited

Stoichiometric Data

state	ton/ton output	ton/ton product
S1	0.20	
S9	0.25	
S10	0.35	
S11	0.20	
S7		0.7
S8		1

of 7.41. The proposed formulation leads to 77 binary variables, 1022 constraints, and 712 continuous variables and gives an optimum objective value of 9.183 in 44.438 CPU s. The Gantt chart corresponding to this formulation is shown in Figure 12.

Table 8. Results for the Industrial Case Study

	proposed approach 1	proposed approach 2	formulation of Ierapetritou and Floudas[18]
NTP	7	7	7
NC	1562	1022	1157
NV	1111	789	439
NB	77	77	98
MILP solution	18.518	9.183	9.183
relaxed objective	22.222	14.821	12.353
CPU time (s)	72.215	44.438	238.98

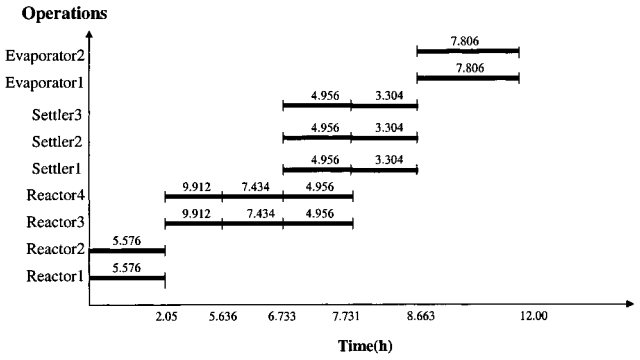


Figure 12. Gantt chart for the industrial case study using proposed approach 2.

Applying the formulation proposed by Ierapetritou and Floudas[18] also required seven time points for optimal results. Their formulation involves 98 binary variables, 1157 constraints, and 439 variables (binary plus continuous variables) and gives the same objective value in 238.98 CPU s. The longer solution time is due to the large number of binary variables. This is due to the separation of task and unit events and the large number of constraints that arises from sequencing. In both formulations, convergence was reached within a

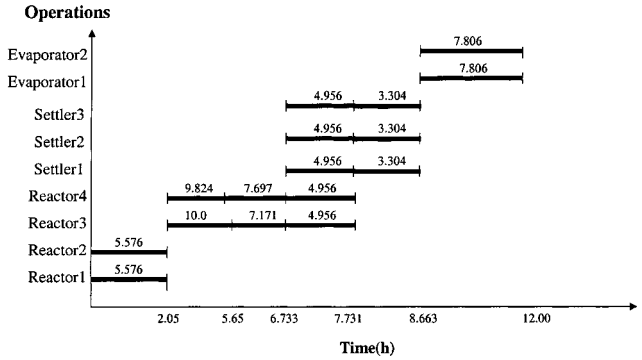


Figure 13. Gantt chart for the industrial case study using the approach by Ierapetritou and Floudas.

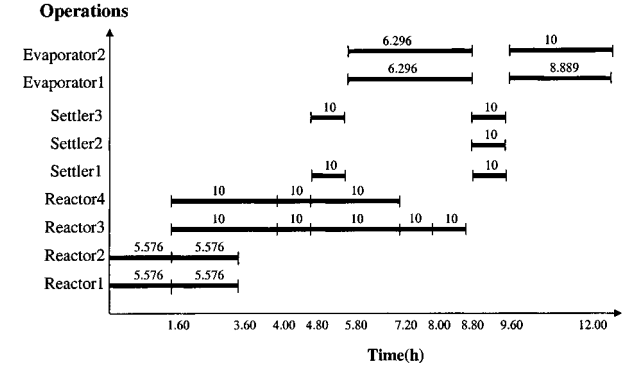


Figure 14. Gantt chart for the industrial case study using proposed approach 1.

tolerance of 10^{-6}. The Gantt chart corresponding to the formulation by Ierapetritou and Floudas[18] is shown in Figure 13.

Exploring the added degrees of freedom that are intrinsic in batch operations in modeling the duration constraints gives the results in the second column of Table 8. Using seven time points in this model, the proposed approach involves 77 binary variables, 1562 constraints, and 1034 continuous variables. The optimal objective value is 18.518 with an integrality gap of 20%. This implies that 18.518 tons of product can be produced within the time horizon of interest, i.e., 12 h. Using five and six time points gave objective values of 7.407 and 14.814, respectively. More time points than seven did not improve the result. The solution was obtained in 72.215 CPU s. This result is 50.03% better than that obtained using batch sizes in modeling duration constraints. The Gantt chart corresponding to this formulation is shown in Figure 14.

7. Application of Aggregation Models in Reducing the Binary Dimension

The aggregation models are introduced in this paper in order to demonstrate that, in operations where all of the units involved in a particular stage have the same performance, the binary dimension can be dramatically reduced, thereby alleviating computational intensity. In using aggregation models, all of the processors involved in a single stage are treated as a single unit operation. It is worth noting, however, that the implicit assumption made when applying aggregation models is that all of the processors in a single stage operate in phase. Although this condition seems like an oversimplification, the results obtained are very similar to those obtained using the general model presented earlier in this paper. Example 2 and the industrial case study are revisited in order to demonstrate the effectiveness of this concept.

Second Literature Example[18] Revisited. Table 4 shows the data for this example. Because aggregation entails a combination of all of the processors in a particular stage, the main distinction from the general formulation is in the capacity constraints and the material balances. These are presented below.

Capacity Constraints.

Reaction 1

States s_2 and s_3 producing state s_6

$$m_u(s_2,p) + m_u(s_3,p) \le (50 + 80)y(s_2,p), \quad \forall\, p \in P$$

The coefficient of the binary variable is the sum of the capacities for the two reactors. State s_2 has been chosen as the effective state.

Reaction 2

States s_5 and s_3 producing state s_7

$$m_u(s_6,p) + m_u(s_5,p) \le (50 + 80)y(s_6,p), \quad \forall\, p \in P$$

In this reaction state s_6 has been chosen as the effective state.

Reaction 3

States s_4 and s_8 producing state s_9

$$m_u(s_8,p) + m_u(s_4,p) \le (50 + 80)y(s_4,p), \quad \forall\, p \in P$$

In this reaction state s_4 has been chosen as the effective state.

Material Balances.

Reaction 1

$$m_u(s_2,p) + m_u(s_3,p) = m_p(s_6,p), \quad \forall\, p \in P$$

Reaction 2

$$m_u(s_6,p) + m_u(s_5,p) = \frac{10}{4} m_p(s_7,p), \quad \forall\, p \in P$$

The coefficient of $m_p(s_7,p)$ is taken from the mass fractions of product 1 and intAB given in the SSN.

Reaction 3

$$m_u(s_8,p) + m_u(s_4,p) = m_p(s_9,p), \quad \forall\, p \in P$$

It should be noted that, unlike in the general formulation, in all of these equations no distinction is made as to which unit the state goes.

Computational Results. Table 9 gives the computational results from the aggregation model.

The results shown in the second and third columns were obtained using GAMS 2.25/OSL in a 333 MHz AMD K6-2 processor. The other results were taken directly from the literature.[18] It should be realized that the aggregation model gives the same value of the objective function as the other general formulations (columns 4−6 in Table 9). It is, however, highly imperative to note that the proposed formulation requires only 25 binary variables compared to 48, 147, and 130 required in the formulations by Ierapetritou and Floudas,[18] Zhang,[13] and Schilling and Pantelides,[15] respec-

Table 9. Results for the Second Literature Example Using Aggregation Models[a]

	aggregation		formulation of		
	proposed approach 1	proposed approach 2	Ierapetritou and Floudas[18]	Zhang[13]	Schilling and Pantelides[15]
NTP	6	5	6	7	6
NC	629	343	465	741	587
NV	431	236	310	497	386
NB	30	25	48	147	130
MILP solution	3315.00	1498.257	1503.18	1497.69	1488.05
relaxed objective	3461.25	1730.652	1984.17	2258.71	2783.14
CPU time (s)	2.195	0.883	2.91	1027.5	N/A

[a] NTP = number of time points; NC = number of constraints; NV = total number of variables; NB = number of binary variables.

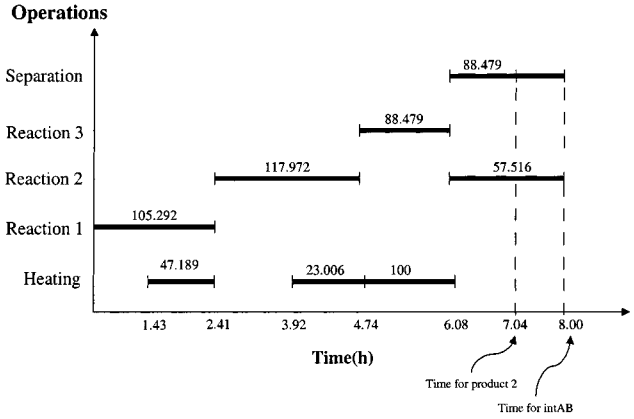


Figure 15. Gantt chart for the second literature example using an aggregation model (proposed approach 2).

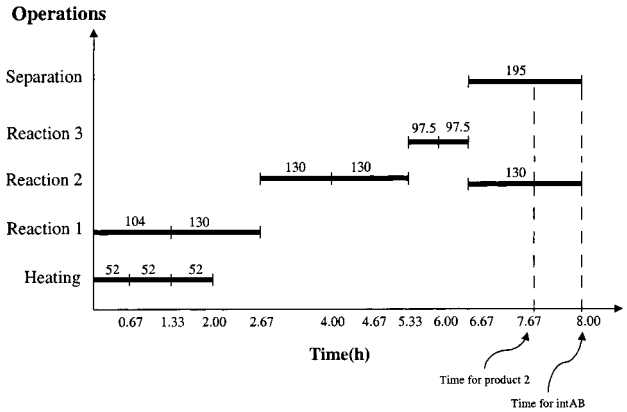


Figure 16. Gantt chart for the second literature example using an aggregation model (proposed approach 1).

tively. Moreover, this formulation results in the least number of constraints and variables compared to the other formulations. More time points than five did not improve the results. The Gantt chart corresponding to this formulation is shown in Figure 15.

It is evident that the Gantt chart does not show how much should be allocated to each processor in stages with more than one processor, e.g., in reactions 1−3. Considering reaction 1, 105.292 capacity units are required between 0 and 2.41 h. This is easily overcome by realizing that the performance of a reactor is not dependent on the capacity of the material, as long as it is within the design capacity limits. Therefore, this amount can be split into any feasible proportions to reactors 1 and 2, implying that both reactors should be conducting reaction 1 during this time interval. The same reasoning also applies to reactions 2 and 3. For an example, if the capacity requirements for reactions 1−3 are split into a 38:62 ratio between reactors 1 and 2, respectively, a schedule which is very similar to that given in Figure 8 is obtained. The ratio that is mentioned in the foregoing statement is, in essence, the ratio of the reactor capacities given in the problem description (Table 4). This proves that the general formulation could have been avoided by simply using the aggregation model. This is always possible in situations where the performance of processors in every stage is the same, and they work in-phase. Although the in-phase operation was not mentioned in the problem description, it is evident from Figure 8 that the optimal performance is achieved by adopting an in-phase operation. It should also be noted that the proposed formulation leads to a much smaller integrality gap, i.e., 13.43%, compared to the formulations by Ierapetritou and Floudas (24.24%), Zhang (33.7%), and Schilling and Pantelides (46.53%). Convergence was reached within a tolerance of 10^{-6}.

Table 10. Results for the Industrial Case Study Using Aggregation Models

	aggregation		formulation of
	proposed approach 1	proposed approach 2	Ierapetritou and Floudas[18]
NTP	7	6	7
NC	743	456	1157
NV	495	306	439
NB	35	30	98
MILP solution	18.518	9.183	9.183
relaxed objective	22.222	14.821	12.353
CPU time (s)	2.473	0.816	238.98

The results shown in the second column of Table 9 were obtained by modeling duration constraints independent of the batch size. Six time points and an 8 h time horizon were used in the proposed formulation, which leads to an objective value of 3315.00. This is the same objective obtained using the general formulation presented earlier in this paper (Table 5). However, this formulation only requires 30 binary variables, 629 constraints, and 401 continuous variables. The solution was obtained in 2.195 CPU s. The Gantt chart resulting from this formulation is shown in Figure 16.

It is worth mentioning, however, that, although the objective function is similar to the general case, this Gantt chart is different from the one given in Figure 9. This is due to the fact that the latter requires an out-of-phase operation of reactors, especially with respect to reaction 1. In both Figures 15 and 16, product 2 is produced almost 1 h prior to the completion of separation in accordance with problem specification in Table 4.

Industrial Case Study Revisited. The data for the industrial case study appear in Table 7.

Computational Results. The results for the application of the aggregation model to the industrial case study are given in Table 10.

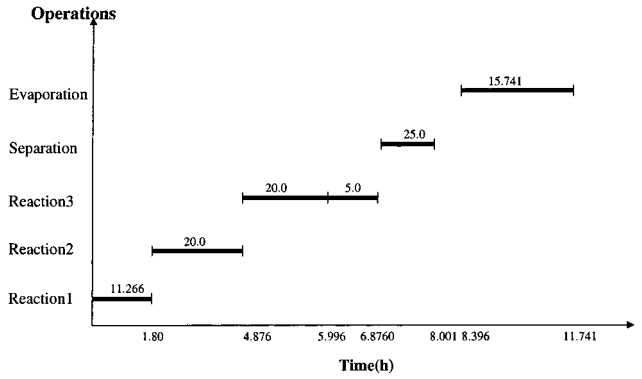


Figure 17. Gantt chart for the industrial case study using an aggregation model (proposed approach 2).

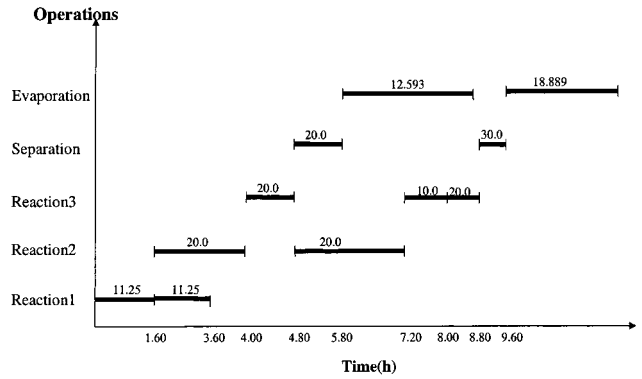


Figure 18. Gantt chart for the industrial case study using an aggregation model (proposed approach 1).

It is worth noting that, although the aggregation model leads to much smaller problems and requires significantly reduced computational effort (CPU time), it gives results similar to those of the general formulation, which renders it a potential candidate for large-scale industrial problems. The Gantt charts corresponding to proposed approach 2 and proposed approach 1 are shown in Figures 17 and 18, respectively.

8. Conclusions

A novel SSN representation is proposed in this paper. On the basis of this representation, a continuous time formulation for scheduling of multipurpose batch processes is developed. This representation involves states only, which are characteristic of the units and tasks present in the process. Because of the elimination of tasks and units which are encountered in formulations based on the STN, the SSN-based formulation leads to a much smaller number of binary variables and fewer constraints. This eventually leads to much shorter CPU times as substantiated by all of the examples presented in this paper. This advantage becomes more apparent as the problem size increases. In the second literature example, which involved a multipurpose plant producing two products, this formulation required 40 binary variables and gave a performance index of 1513.35, while other continuous-time formulations required between 48[18] and 147 binary variables.[13] For the industrial case study, this formulation required 77 binary variables and 44.438 CPU s, instead of 98 binary variables and 238.98 CPU s required in the formulation by Ierapetritou and Floudas.[18]

Also presented in this paper is the new formulation for the duration constraints. In the past, most of the published formulations associated variation in batch times to batch sizes. In this work, we first identify the fact that the variation in batch times is mainly due to degrees of freedom that are inherent in batch processes. These degrees of freedom are due to the choice of catalysts, purity of selected raw materials, and operator response times. We further demonstrate that the formulations which associate batch times with batch sizes lead to suboptimal results because of the implicit restrictions imposed on the time domain. This is substantiated by the literature examples and the industrial case study presented in this paper. In the two literature examples presented, the proposed formulation resulted in improvements of 40% and 121% of the performance index (revenue). In the industrial case study, a 100% improvement in the performance index (production throughput), i.e., 9.183−18.518 tons, was realized by using the proposed formulation. The slack variables were used to cater to the degrees of freedom.

Last, this paper presents the concept of aggregation as a means of reducing the binary dimension in large-scale problems. In all of the examples cited, the objective values predicted by the aggregation model were similar to those predicted by the general formulation. However, the aggregation model requires a much smaller number of binary variables, which is concomitant with significantly reduced computational effort. For the second literature example, this formulation required only 25 binary variables, while other continuous-time formulations required between 48[18] and 147 binary variables.[13] For the industrial case study, this formulation required only 30 binary variables and 0.816 CPU s, instead of 98 binary variables and 238.98 CPU s required in the formulation by Ierapetritou and Floudas.[18] This was achieved without compromising the value of the performance index.

Appendix: Linearization of Bilinear Terms for the Duration Constraints

Let $y(s,p)\,\rho^+(s,p) = \Gamma^+(s,p)$ and $y(s,p)\,\rho^-(s,p) = \Gamma^-(s,p)$.

Upper and lower bounds for the variables:

$$0 \le y(s,p) \le 1 \tag{a.1}$$

$$-v(s)\,\tau(s) \le \rho^-(s,p) \le 0 \tag{a.2}$$

$$0 \le \rho^+(s,p) \le v(s)\,\tau(s) \tag{a.3}$$

First, consider eqs a.1 and a.2.

Because $y(s,p) - 1 \le 0$ and $\rho^-(s,p) \le 0$,

$$\Rightarrow [y(s,p) - 1]\rho^-(s,p) \ge 0$$

$$\Rightarrow \rho^-(s,p)\,y(s,p) - \rho^-(s,p) \ge 0$$

Because $y(s,p) - 1 \le 0$ and $\rho^-(s,p) + v(s)\,\tau(s) \ge 0$,

$$\Rightarrow [y(s,p) - 1][\rho^-(s,p) + v(s)\,\tau(s)] \le 0$$

$$\Rightarrow y(s,p)\,\rho^-(s,p) + y(s,p)\,v(s)\,\tau(s) - \rho^-(s,p) - v(s)\,\tau(s) \le 0$$

$$\Rightarrow \Gamma^-(s,p) \le \rho^-(s,p) + v(s)\,\tau(s)\,[1 - y(s,p)]$$

$$\therefore\ \rho^-(s,p) \le \Gamma^-(s,p) \le \rho^-(s,p) + v(s)\,\tau(s)\,[1 - y(s,p)]$$

Because $y(s,p) \ge 0$ and $\rho^-(s,p) \le 0$

$$\Rightarrow y(s,p)\,\rho^{-}(s,p) \le 0$$

$$\Rightarrow \Gamma^{-}(s,p) \le 0$$

$y(s,p) \ge 0$ and $\rho^{-}(s,p) + v(s)\,\tau(s) \ge 0$,

$$\Rightarrow y(s,p)[\rho^{-}(s,p) + v(s)\,\tau(s)] \ge 0$$

$$\Rightarrow y(s,p)\rho^{-}(s,p) \ge -v(s)\,\tau(s)\,y(s,p)$$

$$\Rightarrow \Gamma^{-}(s,p) \ge -v(s)\,\tau(s)\,y(s,p)$$

$$\therefore -v(s)\,\tau(s)\,y(s,p) \le \Gamma^{-}(s,p) \le 0$$

Then, consider eqs a.1 and a.3.

Because $y(s,p) - 1 \le 0$ and $\rho^{+}(s,p) \le v(s)\,\tau(s)$, where

$$\Rightarrow [y(s,p) - 1][\rho^{+}(s,p) - v(s)\,\tau(s)] \ge 0$$

$$\Rightarrow \rho^{+}(s,p)\,y(s,p) - y(s,p)\,v(s)\,\tau(s) - \rho^{+}(s,p) + v(s)\,\tau(s) \ge 0$$

$$\Rightarrow \Gamma^{+}(s,p) \ge \rho^{+}(s,p) - v(s)\,\tau(s)\,[1 - y(s,p)]$$

where $\Gamma^{+}(s,p) = \rho^{+}(s,p)\,y(s,p)$.

Because $y(s,p) - 1 \le 0$ and $\rho^{+}(s,p) \ge 0$

$$\Rightarrow [y(s,p) - 1]\rho^{+}(s,p) \le 0$$

$$\Rightarrow \rho^{+}(s,p)\,y(s,p) \le \rho^{+}(s,p)$$

$$\Rightarrow \Gamma^{+}(s,p) \le \rho^{+}(s,p)$$

$$\therefore \rho^{+}(s,p) - v(s)\,\tau(s)\,[1 - y(s,p)] \le \Gamma^{+}(s,p) \le \rho^{+}(s,p)$$

Because $y(s,p) \ge 0$ and $\rho^{+}(s,p) - v(s)\,\tau(s) \le 0$,

$$\Rightarrow y(s,p)[\rho^{+}(s,p) - v(s)\,\tau(s)] \le 0$$

$$\Rightarrow \rho^{+}(s,p)\,y(s,p) - y(s,p)\,v(s)\,\tau(s) \le 0$$

$$\Rightarrow \Gamma^{+}(s,p) \le y(s,p)\,v(s)\,\tau(s)$$

Because $y(s,p) \ge 0$ and $\rho^{+}(s,p) \ge 0$

$$\Rightarrow y(s,p)\,\rho^{+}(s,p) \ge 0$$

$$\Rightarrow \Gamma^{+}(s,p) \ge 0$$

$$\therefore 0 \le \Gamma^{+}(s,p) \le v(s)\,\tau(s)\,y(s,p)$$

Literature Cited

(1) Sparrow, R. E.; Forder, G. J.; Rippin, D. W. T. The choice of equipment sizes for multiproduct batch plantssheuristics vs branch and bound. *Ind. Eng. Chem. Process Des. Dev.* **1975**, *14*, 197−203.

(2) Grossmann, I. E.; Sargent, R. W. H. Optimum design of multipurpose chemical plants. *Ind. Eng. Chem. Process Des. Dev.* **1979**, *18*, 343−348.

(3) Knopf, F. C.; Okos, M. R.; Reklaitis, G. V. Optimal design of batch/semicontinuous processes. *Ind. Eng. Chem. Process Des. Dev.* **1982**, *21*, 79−86.

(4) Fattlar et al., 1973.

(5) Gabrielle and Ragsdell, 1976.

(6) Ravemark, D. E.; Rippin, D. W. T. Optimal design of a multiproduct batch plant. *Comput. Chem. Eng.* **1998**, *22*, 177−183.

(7) Suhami, I.; Mah, R. S. H. Optimal design of multipurpose batch plants. *Ind. Eng. Chem. Process Des. Dev.* **1982**, *21*, 94−100.

(8) Lee, H. K.; Lee, I. B. A synthesis of multiproduct batch plants considering both in-phase and out-of-phase modes. *Comput. Chem. Eng.* **1996**, *20*, S195−S200.

(9) Tan, S. T.; Mah, R. S. H. Evolutionary design of noncontinuous plants. *Comput. Chem. Eng.* **1998**, *22* (1−2), 69−85.

(10) Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations—I. MILP formulation. *Comput. Chem. Eng.* **1993**, *17* (2), 211.

(11) Shah, N.; Pantelides, C. C.; Sargent, R. W. H. A general algorithm for short-term scheduling of batch operations—II. Computational issues. *Comput. Chem. Eng.* **1993**, *17* (2), 229−244.

(12) Zhang, X.; Sargent, R. W. H. The optimal operation of mixed production facilities—a general formulation and some solution approaches for the solution. *Proceedings of the 5th International Symposium on Process Systems Engineering*, Kyongju, Korea, 1994; pp 171−177.

(13) Zhang, X. Algorithms for optimal scheduling using nonlinear models. Ph.D. Thesis, University of London, London, U.K., 1995.

(14) Pantelides, 1994.

(15) Schilling, G.; Pantelides, C. C. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput. Chem. Eng.* **1996**, *20*, S1221−S1226.

(16) Glover, F. Improved linear integer programming formulations of nonlinear integer problems. *Manage. Sci.* **1975**, *22* (4), 455−460.

(17) Mockus and Reklaitis, 1997.

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

(19) Pinto, J. M.; Grossmann, I. E. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comput. Chem. Eng.* **1994**, *18*, 797.

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

(21) Karimi, I. A.; McDonald, C. M. Planning and scheduling of parallel semicontinuous processes. 2. Short-term scheduling. *Ind. Eng. Chem. Res.* **1997**, *36*, 2701−2714.

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

(23) McComick, G. P. Computability of global solutions to factorable nonconvex programs. Part I. Convex underestimating problems. *Math. Prog.* **1976**, *10*, 146−175.

Received for review June 5, 2000
Revised manuscript received August 23, 2001
Accepted September 4, 2001

IE0005452