Effective Continuous-Time Formulation for Short-Term Scheduling: I. Multipurpose Batch Processes

M.G. Ierapetritou and C. A. Floudas[1]
Department of Chemical Engineering,
Princeton University,
Princeton, NJ 08544-5263, USA

Abstract–During the last decade, the problem of production scheduling has been realized to be one of the most important problems in industrial plant operations especially when multipurpose/multiproduct batch processes are involved. This paper presents a novel mathematical formulation for the short-term scheduling of batch plants. The proposed formulation is based on a continuous time representation and results in a Mixed Integer Linear Programming (MILP) problem. The novel elements of the proposed formulation are (i) the decoupling of the task events from the unit events, (ii) the time sequencing constraints, and (iii) its linearity. In contrast to the previously presented continuous-time scheduling formulations, the proposed approach leads to smaller and simpler mathematical models which exhibit fewer binary and continuous variables, have smaller integrality gaps, require fewer constraints, need fewer linear programming relaxations, and can be solved in significantly less CPU time. Several examples are presented that illustrate the effectiveness of the proposed formulation and comparisons with other approaches are provided.

1 Introduction

The problem of short-term production scheduling has as objective the determination of the optimal production plan utilizing the available resources over a given time horizon while satisfying the production requirements at due dates and/or at the end of the horizon. The problem becomes more important for batch plants where the production of a large number of products relies on the utilization of a few multipurpose batch equipments. The inherent flexibility of batch plants provides the platform for great savings reflected in a good production schedule.

There is a considerable amount of work in the area of short-term scheduling for batch plants during the last decade. Extensive reviews can be found in Reklaitis[1] and Pantelides[2]. Most of the proposed approaches can be classified based on the time representation they follow. Early attempts to deal with batch scheduling rely on the discretization of the time horizon into a number of intervals of equal duration (Kondili et al.[3]), and the development of specific solution techniques to reduce the computational time required for the solution of resulting models (Shah et al.[4]). The main limitations of the time discretization methods[3,4], are that (i) they correspond to an approximation of the time horizon, and (ii) they result in

[1]Author to whom all correspondence should be addressed; Tel: (609) 258-4595; Fax:(609) 258-0211.

an unnecessary increase of the number of binary variables in particular, and in the overall size of the mathematical model. As a result of these shortcomings of the time discretization methods, recent work aims at developing efficient methods based on the continuous time representation[5;6;7;8;9;10;11]. Zhang and Sargent[11] presented a mixed integer nonlinear programming formulation based on the resource-state-task, RTN, network representation proposed by Pantelides[2]. They apply an exact linearization that gives rise to a very large MILP model that exhibits large integrality gap and consequently is difficult to be solved using conventional MILP solvers. Mockus and Reklaitis[5;6] followed the same principles and proposed a MINLP formulation using the STN representation. They proposed a Bayesian heuristic approach to solve the resulting nonconvex model. Pinto and Grossmann[7], proposed a continuous time formulation to solve the problem of minimizing the earliness of specific orders based on the idea of time slots. They restricted the problem on determining the sequence of tasks to satisfy product demand. Their major limitation is that they did not incorporate any resource constraints and consequently the batch sizes are assumed to be fixed parameters. In a subsequent paper[8], they incorporated preordering constraints in the solution of the resulting MILP formulation to reduce the computational complexity of the problem. They also proposed a decomposition based approach for solving large scale problems. Schilling and Pantelides[9], presented a simpler continuous time formulation for short-term scheduling based on the RTN representation that however gives rise to large MILP problems with large integrality gaps that call for special solution techniques. They, developed a special branch and bound solution methodology in which branching takes place on both continuous and binary variables. McDonald and Karimi[12;13] proposed mathematical models for production planning and short-term scheduling. For the problem of short-term scheduling they proposed two mathematical models that differ on the preassignment of slots to time periods. The proposed formulations can handle the problem of a single-stage multiproduct facility with parallel semi-continuous processors. The model complexity requires the use of preassignment slot to time periods, as well as problem decomposition in order to address medium to large size problems. It should also be pointed out that all slot-based formulations restrict the time representation and hence they result by definition into suboptimal solutions.

In this paper, the objective is to propose a new, simple mathematical model that addresses the general short-term scheduling problem of batch plants as stated in section 2 and which overcomes the aforementioned shortcomings. In section 3, a motivating example is presented to illustrate the complexity of the existing approaches and to motivate the need for new mathematical formulations for the short term scheduling problem. Section 4 presents the new mathematical formulation, whereas in section 5 the illustrative example is revisited and the complete formulation is explicitly presented. In section 6, the proposed approach is tested with example problems that appeared in the literature and comparisons are provided.

2 Problem Statement

The short term scheduling problem for systems of batch processes that is considered in this work is stated as follows. Given (i) the production recipe (i.e., the processing times for each task at the suitable units, and the amount of the materials required for the production of each product), (ii) the available units and their capacity limits, (iii) the available storage capacity for each of the materials, and (iv) the time horizon under consideration, then the objective is to determine (i) the optimal sequence of tasks taking place in each unit, (ii) the amount of material being processed at each time in each unit, and (iii) the processing time of each task in each unit, so as to satisfy the market requirements expressed as specific amounts of products at given time instances within the time horizon. It should be noted however, that in both part I and part II of this paper product demands are considered at the end of the time horizon. The consideration of multiple orders at intermediate due dates will be the subject of a future publication.

3 Motivating Example

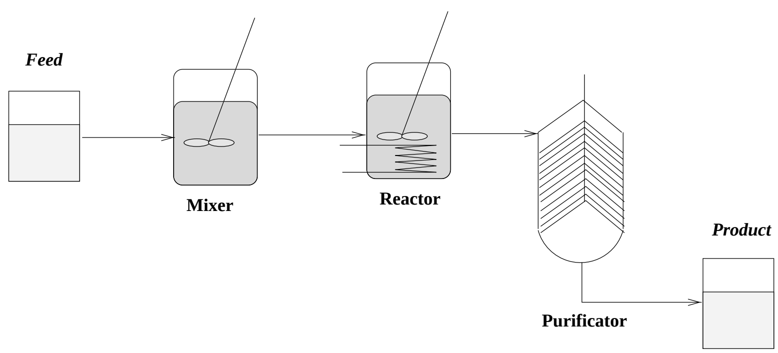

Figure 1: Plant Flowsheet for the Motivating Example

Figure 1 depicts the motivating example that involves the production of a single product through three processing stages, namely mixing, reaction and separation. The State Task Network (STN)[3] representation is selected throughout this paper to describe the plant flowsheet. Following this framework, all the materials are represented as states processed through a set of processing steps ("tasks"). For this simple example the STN is shown in Figure 2.

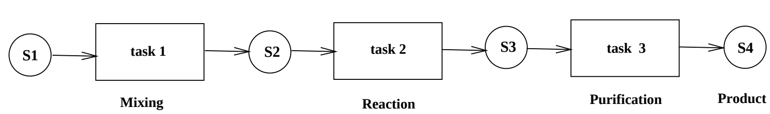

Figure 2: State Task Network representation for the Motivating Example

All the intermediates can be stored in a dedicated storage tank of limited capacity as shown in Table 1. The processing times are allowed to vary within ±33% around the mean values shown in Table 1.

The variable event time formulation of Zhang[10] consists of 263 constraints, 187 continuous variables and 48 binary variables for seven event times. The formulation of Schilling and Pantelides[9] features 220 constraints, 157 continuous variables and 46 binary variables for six event times. It appears that these models have a large number of variables and constraints,

Units	Capacity	Suitability	Mean Processing Time ($\bar{\tau}_{ij}$)
Unit 1	100	task1	4.5
Unit 2	75	task2	3.0
Unit 3	50	task3	1.5
States	**Storage Capacity**	**Initial Amount**	**Price**
State 1	Unlimited	Unlimited	0.0
State 2	100	0.0	0.0
State 3	100	0.0	0.0
State 4	Unlimited	0.0	1.0

Table 1: Data for the Motivating Example

given the simplicity of this motivating example. This simple observation highlights the important point that a new modeling effort is needed. The simple processing system shown in Figure 1 is used to illustrate the effect of variable processing times on production scheduling and clarify the proposed mathematical model that follows.

4 Novel Mathematical Formulation

4.1 Basic Features

The proposed formulation focuses around the following key ideas:

- Continuous time representation
  The proposed formulation is based on a continuous time representation that avoids the prepostulation of unnecessary time slots or intervals. It only requires the initial consideration of a necessary number of *event points* corresponding to either the initiation of a task or the beginning of unit utilization. The location of these points is unknown. A simple iterative procedure, as illustrated in the motivating example, can be used for determining the number of event points needed.

- Decoupling of task events from unit events
  The basic idea of the proposed formulation is that it decouples the *task events (i)* from the *unit events (j)*. This is achieved by the consideration of different variables to represent the *task events* (i.e., the beginning of the task), denoted as $wv(i,n)$, and the *unit events* (i.e., the beginning of unit utilization), denoted as $yv(j,n)$, as shown in Figure 3. If task event (i) starts at event point (n) then wv(i,n)=1, otherwise it is zero. If unit event (j) takes place at event point (n), then yv(j,n)=1, otherwise it is zero.

- Variable Processing times
  Processing times are considered to vary with respect to the amount of the material being processed by the specific task.

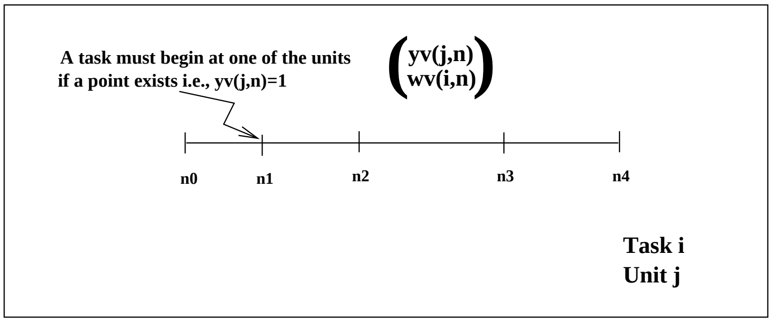

Figure 3: Consideration of events in the proposed formulation

The novel concept of decoupling the tasks from the unit events represents a distinct departure from all previously proposed continuous-time formulations. The decoupling of tasks from units results in smaller size than the previously presented continuous-time formulations mainly due to the fact that binary variables expressing the start of a task (i) in unit (j) at point (n) are avoided. For instance, for example 2 presented in section 6 of this paper, previously presented continuous-time models require 147 and 130 binary variables while the approach in this work needed only 48 binary variables. Moreover, nonlinear terms in the formulation which was a common characteristic of the continuous time formulations for the scheduling problem, are avoided, as will be shown in the section of the mathematical formulation.

4.2 Mathematical Model

The proposed formulation requires the following indices, sets, parameters, and variables:

Indices:

i tasks;

j units;

n event points representing the beginning of a task;

s states.

Sets:

I tasks;

I_j tasks which can be performed in unit (j);

I_s tasks which process state (s) and either produce or consume;

J units;

J_i units which are suitable for performing task (i);

N event points within the time horizon;

S set of all involved states (s).

Parameters:

V_{ij}^{min} denotes the minimum amount of material processed by task (i) required to start operating unit (j);

V_{ij}^{max} denotes the maximum capacity of the specific unit (j) when processing task (i);

$ST(s)^{max}$ available maximum storage capacity for state (s);

$r(s)$ market requirement for state (s) at the end of time horizon;

$\rho_{si}^{p}, \rho_{si}^{c}$ proportion of state (s) produced, consumed from task (i), respectively;

α_{ij} constant term of processing time of task (i) at unit (j);

β_{ij} variable term of processing time of task (i) at unit (j) expressing the time required by the unit to process one unit of material performing task (i);

H time horizon;

price(s) price of state (s).

Variables:

wv(i,n) binary variables that assign the beginning of task (i) at event point (n);

yv(j,n) binary variables that assign the utilization of unit (j) at event point (n);

B(i,j,n) amount of material undertaking task (i) in unit (j) at event point (n);

$d(s,n)$ amount of state (s) being delivered to the market at event point (n);

ST(s,n) amount of state (s) at event point (n);

$T^s(i,j,n)$ time that task (i) starts in unit (j) at event point (n);

$T^f(i,j,n)$ time that task (i) finishes in unit (j) while it starts at event point (n);

Based on this notation the mathematical model for the short-term scheduling of batch plants involves the following constraints:

Allocation Constraints

$$\sum_{i \in I_j} wv(i,n) = yv(j,n), \quad \forall j \in J, \quad n \in N \tag{1}$$

These constraints express that at each unit (j) and at an event point (n) only one of the tasks that can be performed in this unit (i.e., $i \in I_j$) should take place. If unit (j) is utilized at event point (n), that is, yv(j,n) equal 1, then one of the wv(i,n) variables should be activated. If unit (j) is not utilized at point (n), then all wv(i,n) variables take zero values, that is no assignments of tasks are made.

Capacity Constraints

$$V_{ij}^{min} wv(i,n) \leq B(i,j,n) \leq V_{ij}^{max} wv(i,n), \quad \forall i \in I, \quad j \in J_i, \quad n \in N \tag{2}$$

These constraints express the requirement for minimum amount, V_{ij}^{min}, of material in order for a unit (j) to start operating task (i), and the maximum capacity of a unit (j), V_{ij}^{max}, when performing task (i). If wv(i,n) equals one, then constraints (2) correspond to lower and upper bounds on the capacities B(i,j,n). If wv(i,n) equals zero, then all B(i,j,n) variables become zero.

Storage Constraints

$$ST(s,n) \leq ST(s)^{max}, \; \forall s \in S, \; n \in N \tag{3}$$

These constraints represent the maximum available storage capacity for each state (s), at each event point (n). Notice, that the above constraints simply represent an upper bound in the amount of the intermediates that is being produced. Detailed modeling of the storage limitations taking into account the shared limited storage capacity is presented in section 2.1 of the sequel part II paper[14].

Material Balances

$$\begin{aligned} ST(s,n) &= ST(s,n-1) - d(s,n) + \sum_{i \in I_s} \rho_{si}^p \sum_{j \in J_i} B(i,j,n-1) + \\ & \sum_{i \in I_s} \rho_{si}^c \sum_{j \in J_i} B(i,j,n), \quad \forall s \in S, \quad n \in N \end{aligned} \tag{4}$$

where $\rho_{si}^c \leq 0, \rho_{si}^p \geq 0$ represent the proportion of state (s) consumed or produced from task (i), respectively. According to these constraints the amount of material of state (s) at event point (n) is equal to that at event point (n-1) adjusted by any amounts produced or consumed between the event points (n-1) and (n) and the amount required by the market at event point (n) within the time horizon.

Demand Constraints

$$\sum_{n \in N} d(s,n) \geq r(s), \quad \forall s \in S \tag{5}$$

These constraints represent the requirement to produce at least as much as required by the market.

Duration Constraints

$$T^f(i,j,n) = T^s(i,j,n) + \alpha_{ij} wv(i,n) + \beta_{ij} B(i,j,n) \;\; \forall i \in I, \;\; j \in J_i, \;\; n \in N \tag{6}$$

where α_{ij}, β_{ij} are the constant and variable term of the processing time of task (i) at unit (j). Assuming a variation of 33% around the mean value of the processing time $(\bar{\tau}_{ij})$, α_{ij}

takes the value of (2/3)*$\bar{\tau}_{ij}$ and corresponds to the minimum processing time (τ_{ij}^{min}) and $\beta_{ij} = \frac{\tau_{ij}^{max} - \tau_{ij}^{min}}{V_{ij}^{max} - V_{ij}^{min}}$ where $\tau_{ij}^{max} = 4/3 * \bar{\tau}_{ij}$ and $\tau_{ij}^{min} = (2/3) * \bar{\tau}_{ij}$ and expresses the time required by the unit to process one unit of material while performing task (i). The duration constraints express the dependence of the time duration of task (i) at unit (j) at event point (n) from the amount of material being processed. If wv(i,n) equals one, then the last two terms in (6) are added to $T^s(i,j,n)$. If wv(i,n) equals zero, then the last two terms become zero due to the capacity constraints (2) and hence $T^f(i,j,n) = T^s(i,j,n)$.

A very important element of the proposed mathematical model is the set of sequence constraints. These provide the connections between starting and final times and the binary variables $wv(i,n)$ and $yv(j,n)$. The sequence constraints are classified into having (a) the same task in the same unit; (b) different tasks in the same unit; (c) different tasks in different units; and (d) completion of previous tasks, and are presented in detail in the following.

Sequence Constraints: Same task in the same unit

$$\begin{aligned} T^s(i,j,n+1) &\geq T^f(i,j,n) - H(2 - wv(i,n) - yv(j,n)) \\ &\quad \forall i \in I, \;\; j \in J_i, \;\; n \in N, n \neq N \qquad (7) \\ T^s(i,j,n+1) &\geq T^s(i,j,n), \;\; \forall i \in I, \;\; j \in J_i, \;\; n \in N, n \neq N \qquad (8) \\ T^f(i,j,n+1) &\geq T^f(i,j,n), \;\; \forall i \in I, \;\; j \in J_i, \;\; n \in N, n \neq N \qquad (9) \end{aligned}$$

The sequence constraints (7,8,9) state that task (i) starting at event point (n+1) should start after the end of the same task performed at the same unit (j) which has already started at event point (n). If task (i) takes place in unit (j) at event point (n) (i.e., wv(i,n)=yv(j,n)=1), then we have the second term of (7) become zero. If either wv(i,n) or yv(j,n) or both are equal to zero, then constraint (7) is relaxed. Constraints (8,9) are needed to impose the monotonicity in the task timing. An alternative way to pose the timing sequencing for the same task in the same unit is to eliminate the relaxation term of the right hand side of the constraint (7). In this case constraints (8,9) are redundant.

Sequence Constraints: Different tasks in the same unit

The following set of constraints (10) establishes the relationship between the starting time of a task (i) at point (n+1) and the end time of task (i') at event point (n) when these tasks take place at the same unit.

$$\begin{aligned} T^s(i,j,n+1) &\geq T^f(i',j,n) - H(2 - wv(i',n) - yv(j,n)) \\ &\quad \forall j \in J, \;\; i \in I_j, \;\; i' \in I_j, \;\; i \neq i', \;\; n \in N, n \neq N \qquad (10) \end{aligned}$$

Constraints (10) are written for tasks (i, i') that are performed in the same unit (j). If both tasks are performed in the same unit they should be at most consecutive. This is expressed by constraints (10) because if both $wv(i',n) = 1$ and $yv(j,n) = 1$ which means that task (i') takes place at unit (j) at event point (n), then the second term of (10) becomes zero forcing the starting time of task (i) at event point (n+1) to be greater than the end time of task (i') at event point (n); otherwise the right hand side of (10) becomes negative and the constraint is trivially satisfied.

Sequence Constraints: Different tasks in different units

$$\begin{aligned} T^s(i,j,n+1) \ &\geq \ T^f(i',j',n) - H(2 - wv(i',n) - yv(j',n)) \\ &\forall j,j' \in J, \ \ i \in I_j, i' \in I_{j'}, \ \ i \neq i', \ \ n \in N, n \neq N \end{aligned} \tag{11}$$

Constraints (11) are written for different tasks (i, i') that are performed in different units (j, j') but take place consecutively according to the production recipe. Note that if task (i') takes place in unit (j') at event point (n) (i.e., $wv(i', n) = yv(j', n) = 1$), then we have $T^s(i, j, n+1) \geq T^f(i', j', n)$ and hence task (i) in unit (j) has to start after the end of task (i') in unit (j'). Otherwise the right hand side becomes negative and the constraint is trivially satisfied.

Remark: The case where different units share the same tasks can be accommodated in the above formulation by considering each task in each unit as a different task with the same features (see also Example 2 in section 6).

Sequence Constraints: Completion of previous tasks

$$\begin{aligned} T^s(i,j,n+1) \ &\geq \ \sum_{n' \in N, n' \leq n} \sum_{i' \in I_j} (T^f(i',j,n') - T^s(i',j,n')) \\ &\forall i \in I, \ \ j \in J_i, \ \ n \in N, n \neq N \end{aligned} \tag{12}$$

The sequence constraints (12) represent the requirement of a task (i) to start after the completion of all the tasks performed in past event points at the same unit (j).

Time Horizon Constraints

$$T^f(i,j,n) \leq H, \ \ \forall i \in I, j \in J_i, n \in N \tag{13}$$

$$T^s(i,j,n) \leq H, \ \ \forall i \in I, j \in J_i, n \in N \tag{14}$$

The time horizon constraints represent the requirement of every task (i) to start and end within the time horizon (H).

Objective: Maximization of profit

$$\sum_s \sum_n \text{price}(s) d(s,n) \tag{15}$$

The objective shown in (15) is the maximization of production in terms of profit due to product sales. This objective is used throughout the part I of this paper. However, alternative objectives can also be incorporated to express different scheduling targets such as the minimization of makespan.

Remark: Note that since the allocation constraints correspond to definitions of the $yv(j,n)$ variables, we can substitute the $yv(j,n)$ variables in the set of constraints (7), (8), (9) and hence eliminate them, while the allocation constraints can be written as:

$$\sum_{i \in I_j} wv(i,n) \leq 1, \ \forall j \in J, n \in N$$

In the next section the mathematical model is presented for the motivating example presented in section 2.

5 Revisited Motivating Example

As shown in Figure 2, the motivating example involves four states and three tasks that can be performed in three different units. In particular, the raw material is first mixed with some additives in a mixer, then allowed to react in a reactor. In the final stage the final product is purified by any byproducts.

5.1 Mathematical Model

Considering five event points during the time horizon of 12 hours, the explicit constraints for the scheduling problem are the following:

Allocation Constraints

MIXER (j=1):
Since only the mixing task (i=1) is allowed to take place in the mixer unit (j=1), the following allocation constraints are considered:

$$\begin{aligned} wv(1,n_0) &= yv(1,n_0) \\ wv(1,n_1) &= yv(1,n_1) \\ wv(1,n_2) &= yv(1,n_2) \\ wv(1,n_3) &= yv(1,n_3) \\ wv(1,n_4) &= yv(1,n_4) \end{aligned}$$

REACTOR (j=2):
In the reactor the only task allowed is the reaction (i=2), and hence the allocation constraints are:

$$\begin{aligned} wv(2,n_0) &= yv(2,n_0) \\ wv(2,n_1) &= yv(2,n_1) \\ wv(2,n_2) &= yv(2,n_2) \\ wv(2,n_3) &= yv(2,n_3) \\ wv(2,n_4) &= yv(2,n_4) \end{aligned}$$

PURIFICATOR (j=3):
The separation (i=3) is performed in the purificator, and the allocation constraints take the form:

$$
\begin{aligned}
wv(3,n_0) &= yv(3,n_0) \\
wv(3,n_1) &= yv(3,n_1) \\
wv(3,n_2) &= yv(3,n_2) \\
wv(3,n_3) &= yv(3,n_3) \\
wv(3,n_4) &= yv(3,n_4)
\end{aligned}
$$

Remark: Since the allocation constraints have only one wv(i,n) variable in the left hand side, we substitute the $yv(j,n)$ variables in the sequence constraints and hence eliminate the $yv(j,n)$ variables. This results in a reduction of the binary variables and constraints by fifteen.

Capacity Constraints

Given from the description of the plant are the capacities of each unit when performing a specific task. Table 1 provides the upper and lower bounds. Note that the limits may change for different tasks for the same unit.

MIXER (j=1):
For the mixer the maximum capacity is specified to be 100 units and no minimum requirement is specified:

$$
\begin{aligned}
0.0 \leq B(1,1,n_0) \leq 100wv(1,n_0) \\
0.0 \leq B(1,1,n_1) \leq 100wv(1,n_1) \\
0.0 \leq B(1,1,n_2) \leq 100wv(1,n_2) \\
0.0 \leq B(1,1,n_3) \leq 100wv(1,n_3) \\
0.0 \leq B(1,1,n_4) \leq 100wv(1,n_4)
\end{aligned}
$$

REACTOR (j=2):
The reactor has a maximum capacity of 75 units and no minimum limit:

$$
\begin{aligned}
0.0 \leq B(2,2,n_0) \leq 75wv(2,n_0) \\
0.0 \leq B(2,2,n_1) \leq 75wv(2,n_1) \\
0.0 \leq B(2,2,n_2) \leq 75wv(2,n_2) \\
0.0 \leq B(2,2,n_3) \leq 75wv(2,n_3) \\
0.0 \leq B(2,2,n_4) \leq 75wv(2,n_4)
\end{aligned}
$$

PURIFICATOR (j=3):
The purificator can process a maximum of 50 units and has no minimum limit:

$$0.0 \leq B(3,3,n_0) \leq 50wv(3,n_0)$$
$$0.0 \leq B(3,3,n_1) \leq 50wv(3,n_1)$$
$$0.0 \leq B(3,3,n_2) \leq 50wv(3,n_2)$$
$$0.0 \leq B(3,3,n_3) \leq 50wv(3,n_3)$$
$$0.0 \leq B(3,3,n_4) \leq 50wv(3,n_4)$$

Storage Constraints

As given by the data of the problem the intermediate states (s=2,3) can be stored to a dedicated storage tank with limited capacity, whereas for the raw material (s=1) and the product (s=4) there is unlimited storage capacity. Consequently, storage constraints are posed for the intermediates:

STATE S2:

$$ST(s2,n_0) \leq 100$$
$$ST(s2,n_1) \leq 100$$
$$ST(s2,n_2) \leq 100$$
$$ST(s2,n_3) \leq 100$$
$$ST(s2,n_4) \leq 100$$

STATE S3:

$$ST(s3,n_0) \leq 100$$
$$ST(s3,n_1) \leq 100$$
$$ST(s3,n_2) \leq 100$$
$$ST(s3,n_3) \leq 100$$
$$ST(s3,n_4) \leq 100$$

Material Balances

The material balances express the recipe of the production indicating the required states and the relative proportions which are considered as (1) and (-1) for the production and consumption, respectively.

STATE S1:

$$\begin{aligned}
ST(s1,n_0) &= STin(s1) - B(1,1,n_0) \\
ST(s1,n_1) &= ST(s1,n_0) - B(1,1,n_1) \\
ST(s1,n_2) &= ST(s1,n_1) - B(1,1,n_2) \\
ST(s1,n_3) &= ST(s1,n_2) - B(1,1,n_3) \\
ST(s1,n_4) &= ST(s1,n_3) - B(1,1,n_4)
\end{aligned}$$

STATE S2:

$$\begin{aligned}
ST(s2,n_0) &= STin(s2) - B(2,2,n_0)\\
ST(s2,n_1) &= ST(s2,n_0) + B(1,1,n_0) - B(2,2,n_1)\\
ST(s2,n_2) &= ST(s2,n_1) + B(1,1,n_1) - B(2,2,n_2)\\
ST(s2,n_3) &= ST(s2,n_2) + B(1,1,n_2) - B(2,2,n_3)\\
ST(s2,n_4) &= ST(s2,n_3) + B(1,1,n_3) - B(2,2,n_4)
\end{aligned}$$

STATE S3:

$$\begin{aligned}
ST(s3,n_0) &= STin(s3) - B(3,3,n_0)\\
ST(s3,n_1) &= ST(s3,n_0) + B(2,2,n_0) - B(3,3,n_1)\\
ST(s3,n_2) &= ST(s3,n_1) + B(2,2,n_1) - B(3,3,n_2)\\
ST(s3,n_3) &= ST(s3,n_2) + B(2,2,n_2) - B(3,3,n_3)\\
ST(s3,n_4) &= ST(s3,n_3) + B(2,2,n_3) - B(3,3,n_4)
\end{aligned}$$

STATE S4:

$$\begin{aligned}
ST(s4,n_0) &= STin(s4) - d(s4,n_0)\\
ST(s4,n_1) &= ST(s4,n_0) + B(3,3,n_0) - d(s4,n_1)\\
ST(s4,n_2) &= ST(s4,n_1) + B(3,3,n_1) - d(s4,n_2)\\
ST(s4,n_3) &= ST(s4,n_2) + B(3,3,n_2) - d(s4,n_3)\\
ST(s4,n_4) &= ST(s4,n_3) + B(3,3,n_3) - d(s4,n_4)
\end{aligned}$$

Demand Constraints

The product corresponds to state (s4) for which the objective is to maximize the production over the time horizon. No market requirements are specified and consequently no demand constraints are posed.

Timing Constraints

-<u>Duration constraints</u>

The processing times are considered to vary proportional to the amount of material being processed. A variation of ±33% of the mean processing time shown in Table 1 is allowed for all processes.

$$\alpha_{11} = (2/3)\bar{\tau}_{11} = (2/3) * 4.5 = 3.0$$

$$\beta_{11} = \frac{\tau_{11}^{max} - \tau_{11}^{min}}{V_{11}^{max} - V_{11}^{min}} = 3.0/100 = 0.03$$

MIXING:

$$
\begin{aligned}
T^f(1,1,n_0) &= T^s(1,1,n_0) + 3.0 * wv(1,n_0) + 0.03 * B(1,1,n_0) \\
T^f(1,1,n_1) &= T^s(1,1,n_1) + 3.0 * wv(1,n_1) + 0.03 * B(1,1,n_1) \\
T^f(1,1,n_2) &= T^s(1,1,n_2) + 3.0 * wv(1,n_2) + 0.03 * B(1,1,n_2) \\
T^f(1,1,n_3) &= T^s(1,1,n_3) + 3.0 * wv(1,n_3) + 0.03 * B(1,1,n_3) \\
T^f(1,1,n_4) &= T^s(1,1,n_4) + 3.0 * wv(1,n_4) + 0.03 * B(1,1,n_4)
\end{aligned}
$$

REACTION:

$$\alpha_{22} = (2/3)\bar{\tau}_{22} = (2/3) * 3.0 = 2.0$$

$$\beta_{22} = \frac{\tau_{22}^{max} - \tau_{22}^{min}}{V_{22}^{max} - V_{22}^{min}} = 2.0/75 = 0.0266$$

$$
\begin{aligned}
T^f(2,2,n_0) &= T^s(2,2,n_0) + 2.0 * wv(2,n_0) + 0.0266 * B(2,2,n_0) \\
T^f(2,2,n_1) &= T^s(2,2,n_1) + 2.0 * wv(2,n_1) + 0.0266 * B(2,2,n_1) \\
T^f(2,2,n_2) &= T^s(2,2,n_2) + 2.0 * wv(2,n_2) + 0.0266 * B(2,2,n_2) \\
T^f(2,2,n_3) &= T^s(2,2,n_3) + 2.0 * wv(2,n_3) + 0.0266 * B(2,2,n_3) \\
T^f(2,2,n_4) &= T^s(2,2,n_4) + 2.0 * wv(2,n_4) + 0.0266 * B(2,2,n_4)
\end{aligned}
$$

PURIFICATION:

$$\alpha_{33} = (2/3)\bar{\tau}_{33} = (2/3) * 1.5 = 1.0$$

$$\beta_{33} = \frac{\tau_{33}^{max} - \tau_{33}^{min}}{V_{33}^{max} - V_{33}^{min}} = 1.0/50 = 0.02$$

$$
\begin{aligned}
T^f(3,3,n_0) &= T^s(3,3,n_0) + 1.0 * wv(3,n_0) + 0.02 * B(3,3,n_0) \\
T^f(3,3,n_1) &= T^s(3,3,n_1) + 1.0 * wv(3,n_1) + 0.02 * B(3,3,n_1) \\
T^f(3,3,n_2) &= T^s(3,3,n_2) + 1.0 * wv(3,n_2) + 0.02 * B(3,3,n_2) \\
T^f(3,3,n_3) &= T^s(3,3,n_3) + 1.0 * wv(3,n_3) + 0.02 * B(3,3,n_3) \\
T^f(3,3,n_4) &= T^s(3,3,n_4) + 1.0 * wv(3,n_4) + 0.02 * B(3,3,n_4)
\end{aligned}
$$

<u>Remark</u>: The duration constraints are definitions of the final time and as such they can be substituted in the sequence constraints directly, leading to an additional reduction in the number of variables and constraints by fifteen.

Sequence constraints: Same task in the same unit

At each unit any task should start after the end of the previous one. Since in this example only one task is allowed per unit, its task should start after the end of the same task at the previous point if this task takes place at this point:

MIXER:

$$\begin{aligned}
T^s(1,1,n_1) &\geq T^f(1,1,n_0) - H(2 - wv(1,n_0) - yv(1,n_0)) \\
T^s(1,1,n_2) &\geq T^f(1,1,n_1) - H(2 - wv(1,n_1) - yv(1,n_1)) \\
T^s(1,1,n_3) &\geq T^f(1,1,n_2) - H(2 - wv(1,n_2) - yv(1,n_2)) \\
T^s(1,1,n_4) &\geq T^f(1,1,n_3) - H(2 - wv(1,n_3) - yv(1,n_3))
\end{aligned}$$

$$\begin{aligned}
T^s(1,1,n_1) &\geq T^s(1,1,n_0) \\
T^s(1,1,n_2) &\geq T^s(1,1,n_1) \\
T^s(1,1,n_3) &\geq T^s(1,1,n_2) \\
T^s(1,1,n_4) &\geq T^s(1,1,n_3)
\end{aligned}$$

$$\begin{aligned}
T^f(1,1,n_1) &\geq T^f(1,1,n_0) \\
T^f(1,1,n_2) &\geq T^f(1,1,n_1) \\
T^f(1,1,n_3) &\geq T^f(1,1,n_2) \\
T^f(1,1,n_4) &\geq T^f(1,1,n_3)
\end{aligned}$$

REACTOR:

$$\begin{aligned}
T^s(2,2,n_1) &\geq T^f(2,2,n_0) - H(2 - wv(2,n_0) - yv(2,n_0)) \\
T^s(2,2,n_2) &\geq T^f(2,2,n_1) - H(2 - wv(2,n_1) - yv(2,n_1)) \\
T^s(2,2,n_3) &\geq T^f(2,2,n_2) - H(2 - wv(2,n_2) - yv(2,n_2)) \\
T^s(2,2,n_4) &\geq T^f(2,2,n_3) - H(2 - wv(2,n_3) - yv(2,n_3))
\end{aligned}$$

$$\begin{aligned}
T^s(2,2,n_1) &\geq T^s(2,2,n_0) \\
T^s(2,2,n_2) &\geq T^s(2,2,n_1) \\
T^s(2,2,n_3) &\geq T^s(2,2,n_2) \\
T^s(2,2,n_4) &\geq T^s(2,2,n_3)
\end{aligned}$$

$$\begin{aligned}
T^f(2,2,n_1) &\geq T^f(2,2,n_0) \\
T^f(2,2,n_2) &\geq T^f(2,2,n_1) \\
T^f(2,2,n_3) &\geq T^f(2,2,n_2) \\
T^f(2,2,n_4) &\geq T^f(2,2,n_3)
\end{aligned}$$

PURIFICATOR:

$$\begin{aligned}
T^s(3,3,n_1) &\geq T^f(3,3,n_0) - H(2 - wv(3,n_0) - yv(3,n_0))\\
T^s(3,3,n_2) &\geq T^f(3,3,n_1) - H(2 - wv(3,n_1) - yv(3,n_1))\\
T^s(3,3,n_3) &\geq T^f(3,3,n_2) - H(2 - wv(3,n_2) - yv(3,n_2))\\
T^s(3,3,n_4) &\geq T^f(3,3,n_3) - H(2 - wv(3,n_3) - yv(3,n_3))
\end{aligned}$$

$$\begin{aligned}
T^s(3,3,n_1) &\geq T^s(3,3,n_0)\\
T^s(3,3,n_2) &\geq T^s(3,3,n_1)\\
T^s(3,3,n_3) &\geq T^s(3,3,n_2)\\
T^s(3,3,n_4) &\geq T^s(3,3,n_3)
\end{aligned}$$

$$\begin{aligned}
T^f(3,3,n_1) &\geq T^f(3,3,n_0)\\
T^f(3,3,n_2) &\geq T^f(3,3,n_1)\\
T^f(3,3,n_3) &\geq T^f(3,3,n_2)\\
T^f(3,3,n_4) &\geq T^f(3,3,n_3)
\end{aligned}$$

Sequence constraints: Different tasks in different units

The following constraints express the task sequence within the examined system. In this case the reaction should start after the mixing and the separation after the reaction:

MIXING-REACTION:

$$\begin{aligned}
T^s(2,2,n_1) &\geq T^f(1,1,n_0) - H(2 - wv(1,n_0) - yv(1,n_0))\\
T^s(2,2,n_2) &\geq T^f(1,1,n_1) - H(2 - wv(1,n_1) - yv(1,n_1))\\
T^s(2,2,n_3) &\geq T^f(1,1,n_2) - H(2 - wv(1,n_2) - yv(1,n_2))\\
T^s(2,2,n_4) &\geq T^f(1,1,n_3) - H(2 - wv(1,n_3) - yv(1,n_3))
\end{aligned}$$

REACTION-SEPARATION:

$$\begin{aligned}
T^s(3,3,n_1) &\geq T^f(2,2,n_0) - H(2 - wv(2,n_0) - yv(2,n_0))\\
T^s(3,3,n_2) &\geq T^f(2,2,n_1) - H(2 - wv(2,n_1) - yv(2,n_1))\\
T^s(3,3,n_3) &\geq T^f(2,2,n_2) - H(2 - wv(2,n_2) - yv(2,n_2))\\
T^s(3,3,n_4) &\geq T^f(2,2,n_3) - H(2 - wv(2,n_3) - yv(2,n_3))
\end{aligned}$$

Sequence constraints: Completion of previous tasks

At each unit any suitable process should start after the end of all previous ones:

MIXER:

$$
\begin{aligned}
T^s(1,1,n_1) &\geq T^f(1,1,n_0) - T^s(1,1,n_0) \\
T^s(1,1,n_2) &\geq (T^f(1,1,n_0) - T^s(1,1,n_0)) + (T^f(1,1,n_1) - T^s(1,1,n_1)) \\
T^s(1,1,n_3) &\geq (T^f(1,1,n_0) - T^s(1,1,n_0)) + (T^f(1,1,n_1) - T^s(1,1,n_1)) \\
&\quad +(T^f(1,1,n_2) - T^s(1,1,n_2)) \\
T^s(1,1,n_4) &\geq (T^f(1,1,n_0) - T^s(1,1,n_0)) + (T^f(1,1,n_1) - T^s(1,1,n_1)) \\
&\quad +(T^f(1,1,n_2) - T^s(1,1,n_2)) + (T^f(1,1,n_3) - T^s(1,1,n_3))
\end{aligned}
$$

REACTOR:

$$
\begin{aligned}
T^s(2,2,n_1) &\geq T^f(2,2,n_0) - T^s(2,2,n_0) \\
T^s(2,2,n_2) &\geq (T^f(2,2,n_0) - T^s(2,2,n_0)) + (T^f(2,2,n_1) - T^s(2,2,n_1)) \\
T^s(2,2,n_3) &\geq (T^f(2,2,n_0) - T^s(2,2,n_0)) + (T^f(2,2,n_1) - T^s(2,2,n_1)) \\
&\quad +(T^f(2,2,n_2) - T^s(2,2,n_2)) \\
T^s(2,2,n_4) &\geq (T^f(2,2,n_0) - T^s(2,2,n_0)) + (T^f(2,2,n_1) - T^s(2,2,n_1)) \\
&\quad +(T^f(2,2,n_2) - T^s(2,2,n_2)) + (T^f(2,2,n_3) - T^s(2,2,n_3))
\end{aligned}
$$

PURIFICATOR:

$$
\begin{aligned}
T^s(3,3,n_1) &\geq T^f(3,3,n_0) - T^s(3,3,n_0) \\
T^s(3,3,n_2) &\geq (T^f(3,3,n_0) - T^s(3,3,n_0)) + (T^f(3,3,n_1) - T^s(3,3,n_1)) \\
T^s(3,3,n_3) &\geq (T^f(3,3,n_0) - T^s(3,3,n_0)) + (T^f(3,3,n_1) - T^s(3,3,n_1)) \\
&\quad +(T^f(3,3,n_2) - T^s(3,3,n_2)) \\
T^s(3,3,n_4) &\geq (T^f(3,3,n_0) - T^s(3,3,n_0)) + (T^f(3,3,n_1) - T^s(3,3,n_1)) \\
&\quad +(T^f(3,3,n_2) - T^s(3,3,n_2)) + (T^f(3,3,n_3) - T^s(3,3,n_3))
\end{aligned}
$$

Time horizon constraints:

$$
\begin{aligned}
T^f(1,1,n) \leq 12,\ T^s(1,1,n) \leq 12,\ \forall n \in N \\
T^f(2,2,n) \leq 12,\ T^s(1,1,n) \leq 12,\ \forall n \in N \\
T^f(3,3,n) \leq 12,\ T^s(1,1,n) \leq 12,\ \forall n \in N
\end{aligned}
$$

Remark: The final form of the sequence constraints and time horizon constraints will feature the substitution of the final times $T^f(i,j,n)$ based on the duration constraints which are eliminated, and the substitution of the yv(j,n) variables with the corresponding wv(i,n) variables according to the allocation constraints which are also eliminated.

Initial conditions for the intermediates:

$$STin(s2) = 0.0$$
$$STin(s3) = 0.0$$
$$STin(s4) = 0.0$$

Note that raw material has unlimited availability.

Objective: Maximization of profit

The objective of this problem is the maximization of revenue from the product sales for which the price is given to be 1.0 unit. For the rest of the materials no prices are specified. Considering also the amount of product at the final point, the objective has the following form:

$$\max \ d(s4, n_0) + d(s4, n_1) + d(s4, n_2) + d(s4, n_3) + d(s4, n_4) + ST(s4, n_4)$$

Note that in this example only one task can take place in each unit and consequently the binary variables $yv(j, n)$ that correspond to the assignment of the unit (j) to the event (n) can be eliminated from the formulation. The allocation constraints can then be rewritten in the following form:

$$\sum_{i \in I_j} wv(i, n) \leq 1.0, \quad \forall j \in J, \quad n \in N \tag{16}$$

Also since the assignment of the task (i) at the event point (n), (i.e., $wv(i, n) = 1$), the assignment of the corresponding unit (j) at the event point (n), (i.e., $yv(j, n) = 1$), the relaxation terms $H(2 - wv(i, n) - yv(j, n))$ of the sequence constraints are reformulated to the terms $H(1 - wv(i, n))$. Note that it is always possible to substitute $yv(j, n)$ using equation (1) and hence result in a mathematical model with only the $wv(i, n)$ variables.

5.2 Computational Results and Comparisons

The resulting MILP formulation involves 108 constraints including the bounds, 105 continuous variables and 15 binary variables. The GAMS/CPLEX code is used for the solution of MILP. It requires 0.05 CPU sec in HP-C160 workstation to reach the solution within 10E-6 integrality tolerance. The integrality gap of the proposed formulation is 28.5 with 100 being the objective of the LP relaxation problem. The resulting gantt chart is shown in Figure 4. It corresponds to an objective function of 71.518 units. The optimal solution corresponds to the values of $wv(i, n)$ variables provided in Table 2.

As Table 3 shows the proposed formulation has the following advantages compared to the previously presented continuous time formulations for short-term scheduling: (i) it gives rise to a smaller model mainly in terms of integer variables 15 compared to 48 of Variable Event Time formulation of Zhang[11], and 46 of Schilling and Pantelides formulation[9],

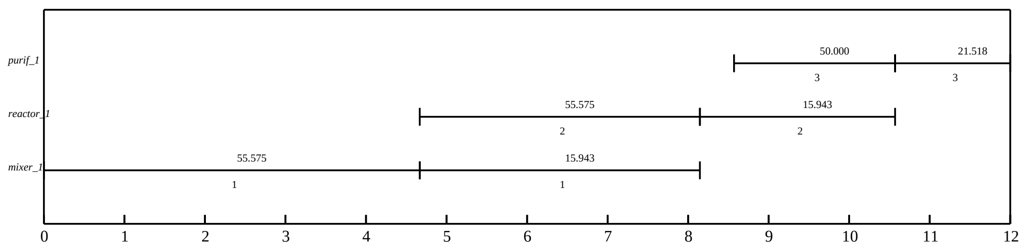

Figure 4: Gantt Chart for the Motivating Example

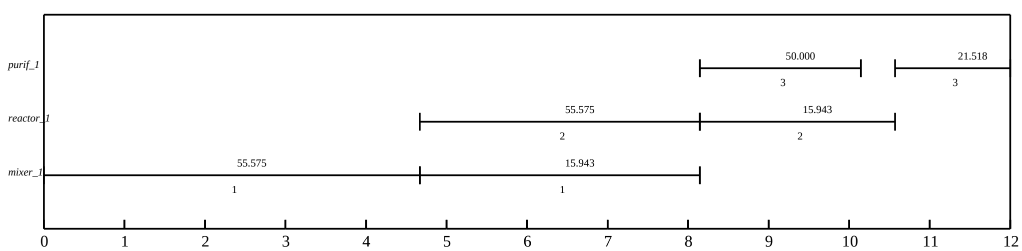

Figure 5: Alternative Schedule for the Motivating Example

Optimal Solution ($wv(i,n)$)					
Task	n_0	n_1	n_2	n_3	n_4
1	1	1	0	0	0
2	0	1	1	0	0
3	0	0	1	1	0

Table 2: Results for the Motivating Example

(ii) the resulting MILP problem has smaller integrality gap 28.5% compared to 52.36% of Zhang's formulation and smaller than the formulation of Schilling and Pantelides[9] which has 138.96%, (iii) the scheduling found from the solution of the proposed formulation corresponds to higher profit value and consequently better utilization of the available resources, namely 71.518 compared to 71.45 and 71.47 presented by the two other formulations, (iv) the CPU time required is 0.05 CPU sec in HP-C160 workstation compared to 21.9 CPU sec reported by Zhang[10] on Sun Sparc10/41 workstation using SCICONIC as a MILP solver, and (v) the number of linear programming problems required from the proposed approach is only 13 compared to 528 required by Zhang's formulation using SCICONIC as a MILP solver, and 418 required by the formulation of Schilling and Pantelides[9], using their specialized branch and bound algorithm.

	Proposed Approach	VET Formulation (Zhang[10])	Formulation of Schilling and Pantelides[9]
NEI	5	7	6
NC	108 (58)	263	220
NV	105 (48)	187	157
NIV	15 (6)	48	46
OBJ MILP	71.518	71.45	71.47
OBJ Relaxed LP	100 (100)	149.99	170.79
LPs for MILP	13	528	418
CPU for MILP	0.05 (0.03)	21.9	-

Table 3: Results for the Motivating Example

5.3 Alternative Schedules

It is interesting to note that, as shown in Figure 5, an alternative schedule exists that corresponds to the same global optimal value of the objective function but with different timings of the tasks performed in the purificator.

5.4 Number of Event Points

Note that the proposed approach requires the consideration of 5 event points which is the optimal number of event points for this example obtained through a general iterative procedure that is described in the following. First, the example is solved considering only 4 event points and the solution obtained corresponds to the value of profit equal to 50 units which is obtained after the solution of 4 LP relaxation problems in 0.04 CPU sec in a HP-C160 workstation using GAMS/CPLEX. Then the number of event points is increased to 5 and the obtained schedule has an objective of 71.518, found after the solution of 13 LP relaxation problem that require 0.05 CPU sec in HP-C160 workstation. An additional increase in the number of event points to 6 *does not* result in an improvement of the objective function. Since the consideration of additional event points does not improve the solution of the problem, the iterative procedure terminates with the 5 event points.

5.5 STN and Further Reduction of Variables and Constraints

It is important to note that we can further reduce the number of binary and continuous variables and the number of constraints by exploiting the information provided by the STN representation shown in Figure 2 and its connection to the different event points. This can be achieved in general but it will be explained via the motivating example for clarity of the presentation.

First, since the last event point corresponds to the end of time horizon, no task should be assigned at this event point and consequently $wv(i, n_4) = 0, B(i, j, n_4) = 0, \forall i \in I, j \in J_i$. Furthermore, according to the STN of Figure 2, task 2 takes place after task 1 while task 3 takes place after task 1 and task 2. Hence, $wv(2, n_0) = wv(3, n_0) = 0$ and $wv(3, n_1) = 0$ since there should at least two event points prior to task 3 taking place. Similarly, task 2 cannot take place in the last active event point, n_3, while task 1 cannot take place in the last two active event points n_2 and n_3. Hence, $wv(2, n_3) = 0$ and $wv(1, n_2) = wv(1, n_3) = 0$. Furthermore, if task 3 does not take place in the last event, n_3, (i.e., it takes place earlier, e.g., at n_2), then task 2 has to take place at least one event point earlier than task 3, while task 1 must take place at least two event points prior to task 3. These logical statements can be expressed with the additional constraints:

$$\begin{aligned} wv(2, n_2) &\leq wv(3, n_3) \\ wv(2, n_1) &\leq wv(3, n_3) \end{aligned}$$

Note that if $wv(3, n_3) = 0$, then $wv(2, n_2) = wv(1, n_1) = 0$.

Based on the aforementioned remarks, we reduce the binary variables $wv(i, n)$ from 15 to 6, and the number of continuous variables from 105 to 48. Furthermore, a number of duration constraints and sequence constraints become redundant reducing the number of constraints form 108 to 58. Using this reduced mathematical model, we can obtain the optimal solution in 0.03 CPU sec in a HP-C160 workstation.

6 Computational Studies and Discussion

In this section, two additional example problems are considered and the effectiveness of the proposed approach is illustrated. Comparisons with previously published approaches are provided. In Appendix A a number of clarification points in regard to (a) optimality, (b) feasibility and (c) number of event points are presented.

6.1 Example 2

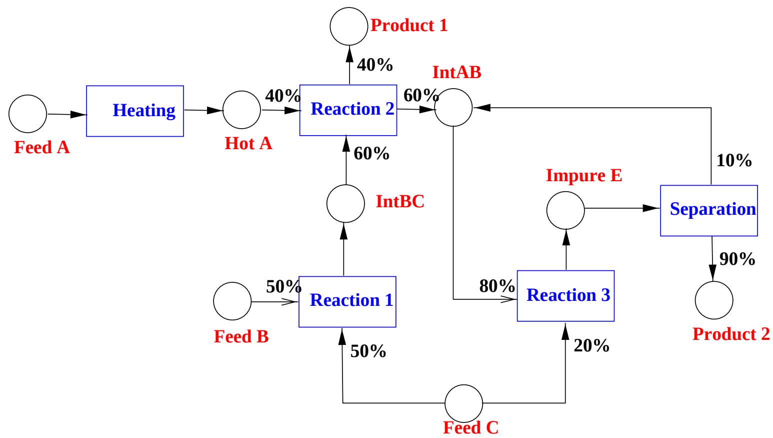

Figure 6: State Task Network representation for Example 2

In this example two different products are produced through five processing stages, heating, reaction 1, 2 and 3, and separation of product 2 from impure E as shown in the STN representation of the plant flowsheet in Figure 6. The data for this example are presented in Table 4. The processing times are allowed to vary within ±33% around the mean values shown in Table 4. The time horizon of interest is 8 hours.

As mentioned in section 4, since different production tasks (i.e., the three reactions) can take place in different units (i.e., the two reactors), each reaction is represented in the proposed formulation by two tasks each one performed at each one of the two reactors. Note here that the three required reactions can be performed in reactors 1 and 2. Consequently, reaction 1 corresponds to tasks 2 and 3 if it takes place at reactor 1 or 2, respectively, and reaction 2 corresponds to tasks 4 and 5 if it takes place at reactor 1 or 2, respectively, and reaction 3 to tasks 6 and 7, if it takes place at reactor 1 or 2, respectively. Constraints (8) for five event points take the following form for the two different reactors:

Units	Capacity	Suitability	Mean Processing Time ($\bar{\tau}_{ij}$)
Heater	100	heating	1.0
Reactor 1	50	reaction 1, 2, 3	2.0, 2.0, 1.0
Reactor 2	80	reaction 1, 2, 3	2.0, 2.0, 1.0
Still	200	separation	1 for product 2, 2 for IntAB
States	**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
ImpureE	200	0.0	0.0
Product 1	Unlimited	0.0	10.0
Product 2	Unlimited	0.0	10.0

Table 4: Data for Example 2

REACTOR 1 (j=2):

$$
\begin{aligned}
T^s(2,2,n+1) &\geq T^f(4,2,n) - H*(2 - wv(4,n) - yv(2,n)) \quad n = n_0, ..., n_4 \\
T^s(2,2,n+1) &\geq T^f(6,2,n) - H*(2 - wv(6,n) - yv(2,n)) \quad n = n_0, ..., n_4 \\
T^s(4,2,n+1) &\geq T^f(2,2,n) - H*(2 - wv(2,n) - yv(2,n)) \quad n = n_0, ..., n_4 \\
T^s(4,2,n+1) &\geq T^f(6,2,n) - H*(2 - wv(6,n) - yv(2,n)) \quad n = n_0, ..., n_4 \\
T^s(6,2,n+1) &\geq T^f(2,2,n) - H*(2 - wv(2,n) - yv(2,n)) \quad n = n_0, ..., n_4 \\
T^s(6,2,n+1) &\geq T^f(4,2,n) - H*(2 - wv(4,n) - yv(2,n)) \quad n = n_0, ..., n_4
\end{aligned}
$$

Similarly constraints (8) for reactor 2 (j=3) are the following:

REACTOR 2 (j=3):

$$
\begin{aligned}
T^s(3,3,n+1) &\geq T^f(5,3,n) - H*(2 - wv(5,n) - yv(3,n)) \quad n = n_0, ..., n_4 \\
T^s(3,3,n+1) &\geq T^f(7,3,n) - H*(2 - wv(7,n) - yv(3,n)) \quad n = n_0, ..., n_4 \\
T^s(5,3,n+1) &\geq T^f(3,3,n) - H*(2 - wv(3,n) - yv(3,n)) \quad n = n_0, ..., n_4 \\
T^s(5,3,n+1) &\geq T^f(7,3,n) - H*(2 - wv(7,n) - yv(3,n)) \quad n = n_0, ..., n_4 \\
T^s(7,3,n+1) &\geq T^f(3,3,n) - H*(2 - wv(3,n) - yv(3,n)) \quad n = n_0, ..., n_4 \\
T^s(7,3,n+1) &\geq T^f(5,3,n) - H*(2 - wv(5,n) - yv(3,n)) \quad n = n_0, ..., n_4
\end{aligned}
$$

Remark: It should be pointed out that since each reaction is represented by two tasks each performed in one of the two reactors, we on the one hand increase the number of tasks from four to eight but on the other hand we maintain the one to one correspondence of the tasks to units. This implies that we can eliminate the $yv(j,n)$ variables by substituting them by their definitions via the allocation constraints.

6.1.1 Results and Comparisons

In this section, we consider the computational results for the two cases (a) five event points and (b) six event points.

(a) Five event points

For five event points, the resulting MILP formulation involves 374 constraints, 260 continuous variables and 40 binary variables. The solution of this problem with GAMS/CPLEX requires 0.28 CPU sec in a HP-C160 workstation. The optimal objective function corresponds to 1503.15 units within the time horizon of 8 hours. The obtained values of $wv(i,n)$ are given in Table 5 and the corresponding gantt chart is shown in Figure 7.

Optimal Solution $(wv(i,n))$					
Task	n_0	n_1	n_2	n_3	n_4
1	1	0	1	0	0
2	1	0	0	0	0
3	1	0	0	0	0
4	0	1	0	1	0
5	0	1	0	1	0
6	0	0	1	0	0
7	0	0	1	0	0
8	0	0	0	1	0

Table 5: Results for Example 2 using five event points

Table 6 shows the results of the proposed formulation compared with the results found in the literature for this example. Note that the model of the proposed formulation involves less constraints, 374 compared to 741 and 587 constraints required by the other formulations. Also note that the number of variables is much smaller, 260, compared to 497 required by Zhang's VET formulation[10] and 386 by Schilling and Pantelides[9]. More importantly the binary variables involved in the proposed formulation are only 40 compared to 147 and 130 involved in the two other models. Consequently, the proposed formulation can be solved efficiently within 10E-6 integrality gap resulting in better objective value, that is, better utilization of the existing units. The required number of linear programming relaxation is 51 while 9575 and 2768 linear programming relaxations are needed for the formulation of Zhang[10] on Sun Sparc10/41 workstation using SCICONIC as a MILP solver and the formulation of Schilling and Pantelides[9], who used their specialized branch and bound algorithm.

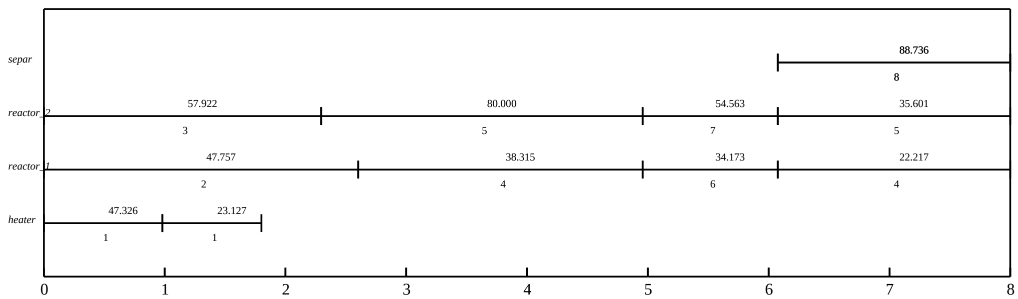

Figure 7: Gantt Chart for Example 2 using five event points

(b) Six Event Points

For six event points, Table 6 shows that the resulting MILP formulation involves 465 constraints, 310 continuous variables and 48 binary variables. The solution of this problem with GAMS/CPLEX requires 2.91 CPU sec in a HP-C160 workstation. The optimal objective function corresponds to 1503.1832 units within the time horizon of 8 hours. The corresponding solution and gantt chart is shown in Table 7 and in Figure 8.

6.1.2 Imposing Additional Restrictions: Alternative Schedules

(a) Five Event Points

Since reaction 2 that corresponds to tasks 4 and 5 which can take place in reactor 1 and 2, respectively, should start after the heating task, constraints (9) are of the following form:

REACTOR 1 (j=2):

$$T^s(4,2,n+1) \;\; \geq \;\; T^f(1,1,n) + H*(2 - yv(1,n) - wv(1,n)) \;\; n = n_0, ..., n_4$$

REACTOR 2 (j=3):

$$T^s(5,3,n+1) \;\; \geq \;\; T^f(1,1,n) + H*(2 - yv(1,n) - wv(1,n)) \;\; n = n_0, ..., n_4$$

	Proposed Approach		VET Formulation (Zhang[10])	Formulation of Schilling and Pantelides[9]
NEI	5	6	7	6
NC	374	465	741	587
NV	260	310	497	386
NIV	40	48	147	130
OBJ MILP	1503.15	1503.18	1497.69	1488.05
OBJ Relaxed LP	1732.36	1984.17	2258.71	2783.14
LPs for MILP	51	554	9575	1230
CPU for MILP	0.28	2.91	1027.5	-

Table 6: Results for Example 2

Note that if one wishes to enforce that the reaction 2 starts immediately after the end of the heating task in the heater, the following set of sequence constraints are added together with constraints (9):

REACTOR 1 (j=2):

$$T^s(4,2,n+1) \leq T^f(1,1,n) + H * (2 - yv(1,n) - wv(1,n)) \quad n = n_0, ..., n_4$$

REACTOR 2 (j=3):

$$T^s(5,3,n+1) \leq T^f(1,1,n) + H * (2 - yv(1,n) - wv(1,n)) \quad n = n_0, ..., n_4$$

It is worth noting that in the case where these constraints are considered the optimal schedule obtained has the same value of profit equal to 1503.15 units but different timings for the heating tasks as shown in Figure 9.

(b) Six Event Points

In a similar way if six event points are considered and the additional set of constraints are incorporated in order to enforce the final time of heating task to equal the starting time of reaction 2, the optimal schedule shown in Figure 10, is obtained having the same value of profit equal to 1503.1832 units but different timings for the heating tasks.

6.1.3 Different Horizon Constraint

The same example is considered here except that the time horizon is extended to 10 hr and a price of -1 is considered for the intermediates products in order to penalize the remaining intermediates at the end of the time horizon in storage as considered by Kondili et al.[3].

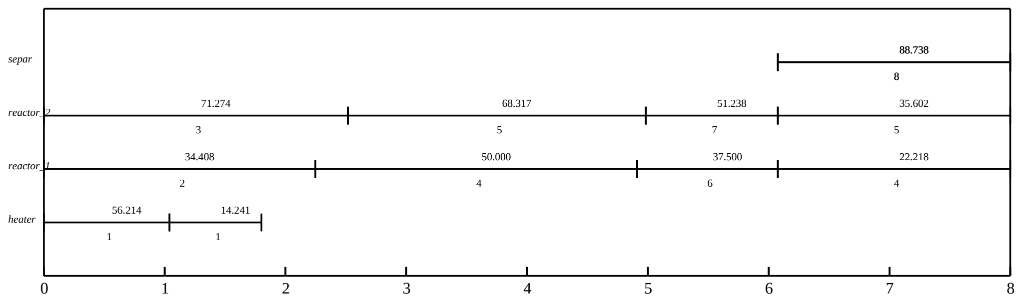

Figure 8: Gantt Chart for Example 2 using six event points

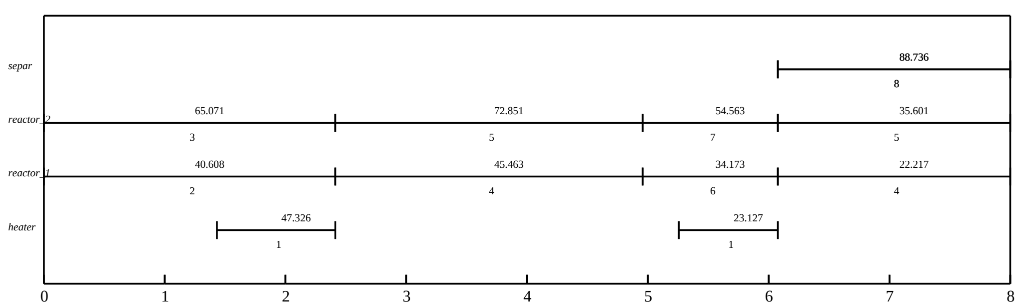

Figure 9: Alternative Schedule for Example 2 using five event points

Optimal Solution ($wv(i,n)$)						
Task	n_0	n_1	n_2	n_3	n_4	n_5
1	1	0	0	1	0	0
2	1	0	0	0	0	0
3	0	1	0	0	0	0
4	0	1	0	1	0	0
5	0	0	1	0	1	0
6	0	0	1	0	0	0
7	0	0	0	1	0	0
8	0	0	0	0	1	0

Table 7: Results for Example 2 using six event points

Assuming variable processing times, the proposed formulation requires the utilization of 7 event points and results in a mathematical model involving 560 constraints, 360 continuous variables, and 42 binary variables. Using GAMS/CPLEX for the solution of this mathematical model requires 8.9 CPU sec in HP-C160 workstation and 1745 nodes in order to obtain the optimal solution of 1949.13 units that corresponds to the schedule shown in Figure 11. Note, that compared to the discrete time formulation using constant processing times, the proposed formulation results in smaller profit 1949.13 compared to 2744 obtained form the discrete time formulation using constant processing times, since the amount of material being processed does not depend in this case on the processing time.

6.1.4 STN and Further Reduction of Variables and Constraints

Based on the STN representation shown in Figure 6, we can rigorously introduce the following simplifications:

$$\begin{aligned} wv(i,n_0) &= 0, \quad i = 4,5,6,7,8 \\ wv(i,n_1) &= 0, \quad i = 6,7,8 \\ wv(i,n_2) &= 0, \quad i = 8 \\ wv(i,n_f-1) &= 0, \quad i = 1,2,3,7,8 \\ wv(i,n_f) &= 0, \quad \forall i \in I \end{aligned}$$

where n_f corresponds to the final event point used. These simplifications reduce the number of binary variables from 40 to 19 for the case where five event points are used and from 48 to 27 if six event points are used. In this case the optimal solution is obtained in 0.23 and 2.16 CPU sec in a HP-C160 workstation considering five and six event points, respectively.

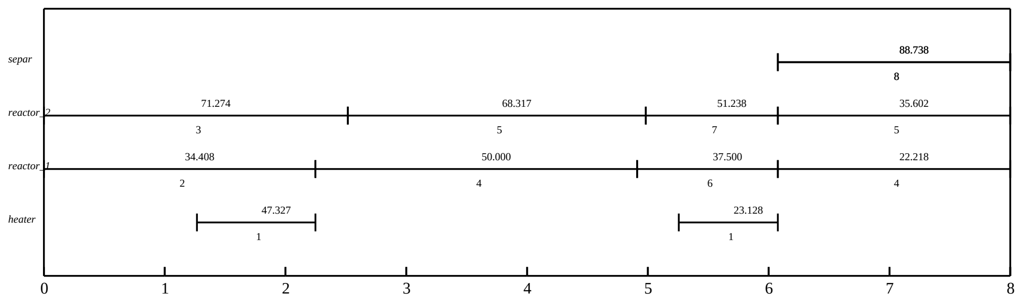

Figure 10: Alternative Schedule for Example 2 using six event points

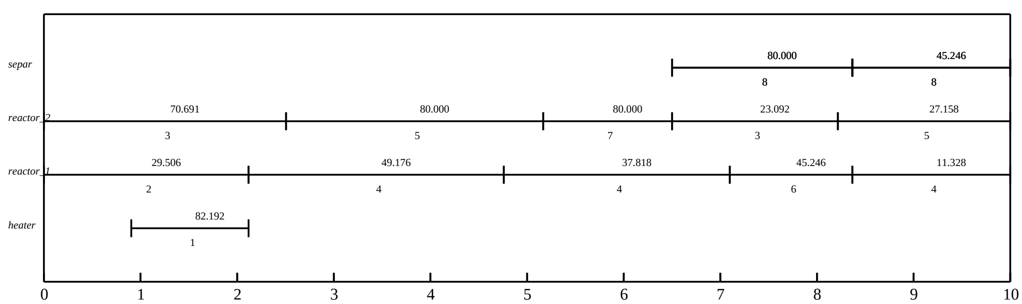

Figure 11: Gantt Chart for Example 2 with time horizon H=10hrs

6.2 Example 3

This example involves the production of one product through four processing tasks: reaction, mixing, filtering and stripping. Figure 12 illustrates the STN representation of this plant network. The time horizon is 76 hours and the processing times are considered to vary within ±33% of the mean time shown in Table 8. The detailed data of this example are also presented in Table 8.

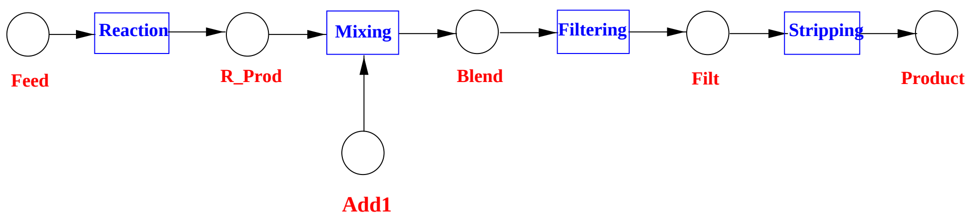

Figure 12: State Task Network representation for Example 3

6.2.1 Results and Comparisons

Note that similarly to the example 2, the mixing and stripping tasks can take place in different units, the two mixers and the two Strip-tanks, respectively, and consequently they are formulated as different tasks, that is the mixing task is split into two different tasks that take place into two different mixers, and the stripping task is formulated as two different stripping tasks that take place into two different Strip-tanks. Also note that the nature of the problem enables the elimination of the binary variables ($yv(j,n)$) as it was done in the motivating example. The proposed formulation gives rise to an MILP problem with 340 constraints, 259 continuous variables and 42 binary variables. The solution of this problem with GAMS/CPLEX requires 1.24 CPU sec in a HP-C160 workstation. The optimal objective function corresponds to 59.25 units. The corresponding results and gantt chart is shown in Table 9 and in Figure 13.

Table 10 shows the results of the proposed formulation compared with the results found in the literature for this example. Note that the number of constraints and variables are much smaller than the constraints and variables used in the other formulations. In particular, the proposed formulation involves 340 constraints compared to 605 and 643 constraints used by the other two formulations, 42 binary variables compared to 119 and 126 number of binary variables required by Zhang's[10] and Schilling and Pantelides[9] formulations, respectively. This enables the solution of the problem within 10E-6 integrality tolerance that gives rise to an improved objective function in much smaller CPU time. In this case 645 linear programming relaxations are required compared to 1836 needed by Zhang's formulation[10] using SCICONIC as a MILP solver on Sun Sparc10/41 workstation, and 11954 required by Schilling and Pantelides[9] formulation using their specialized branch and bound algorithm.

It should be pointed out that in this example due to the limited capacity of the units, the same optimal solution is obtained without splitting the mixing and the stripping tasks. The optimal schedule obtained corresponds to the same global optimum value of profit equal

Units	**Capacity**	**Suitability**	**Mean Processing Time** ($\bar{\tau}_{ij}$)
Mixer 1	20	mixing	4.0
Mixer 2	20	mixing	4.0
Reactor	20	reaction	26.0
Filter	20	filtering	6.0
Strip-tank 1	20	Stripping	8.0
Strip-tank 2	20	Stripping	8.0
States	**Storage Capacity**	**Initial Amount**	**Price**
Feed	Unlimited	Unlimited	0.0
Add 1	Unlimited	Unlimited	0.0
R-Prod	100	0.0	0.0
Blend	100	0.0	0.0
Filt	100	0.0	0.0
Prod	Unlimited	0.0	2.0

Table 8: Data for Example 3

Optimal Solution ($wv(i,n)$)							
Task	n_0	n_1	n_2	n_3	n_4	n_5	n_6
1	1	1	0	0	0	0	0
2	0	1	0	1	0	0	0
3	0	1	0	1	0	0	0
4	0	0	1	1	1	0	0
5	0	0	0	0	1	1	0
6	0	0	0	1	0	1	0

Table 9: Results for Example 3

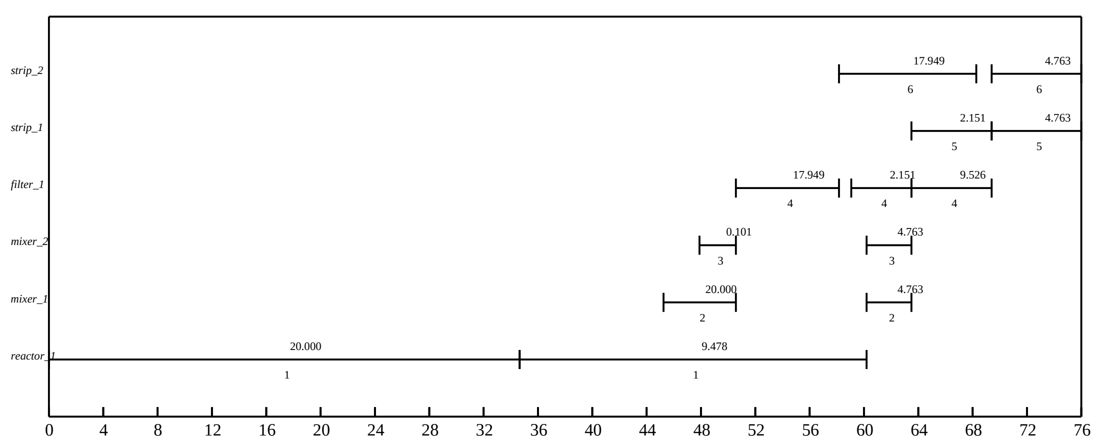

Figure 13: Gantt Chart for Example 3

	Proposed Approach	VET Formulation (Zhang[10])	Formulation of Schilling and Pantelides[9]
NEI	7	8	8
NC	340	605	643
NV	259 (245)	426	414
NIV	42 (28)	119	126
OBJ MILP	59.25	57.72	58.55
OBJ Relaxed LP	88.16	88.12	80.04
LPs for MILP	645 (84)	1836	11954
CPU for MILP	1.24 (0.34)	176.2	-

Table 10: Results for Example 3

to 59.25 units and the schedule illustrated in Figure 14. Note that in this case we only have four tasks since we do not split the mixing and splitting tasks. Hence task 2 represents the mixing and task 4 represents the stripping. In this case the solution of the proposed formulation having 28 binary and 245 continuous variables, requires the solution of only 84 LP relaxation problems solved in 0.34 CPU sec in a HP-C160 workstation.

6.2.2 Minimum Capacity Requirement

The consideration of the additional requirement of 5 units to ensure unit operation results in a mathematical formulation with 36 binary and 223 continuous variables and 289 constraints, the solution of which requires 0.19 CPU sec in a HP-C160 workstation to obtain the optimal schedule of 58.814 units solving 60 LP relaxation problems. The optimal schedule is shown in Figure 15. Note that this optimal Schedule is still better than the optimal solution of Zhang[10] and Schilling and Pantelides[9] shown in Table 10. It should be noted that in this case the mathematical model exhibits smaller integrality gap since the objective of the first LP relaxation problem corresponds to a value of 80 units instead of 88.16 units which is the objective function of the relaxation problem in the case that no minimum capacity requirements are incorporated.

6.2.3 STN and Further Reduction of Variables and Constraints

Based on the STN representation shown in Figure 12, we can rigorously introduce the following simplifications:

$$\begin{aligned}
wv(i,n_0) &= 0, \quad i = 2,3,4,5,6\\
wv(i,n_1) &= 0, \quad i = 4,5,6\\
wv(i,n_2) &= 0, \quad i = 5,6\\
wv(i,n_5) &= 0, \quad i = 1,2,3,4\\
wv(i,n_6) &= 0, \quad \forall i \in I
\end{aligned}$$

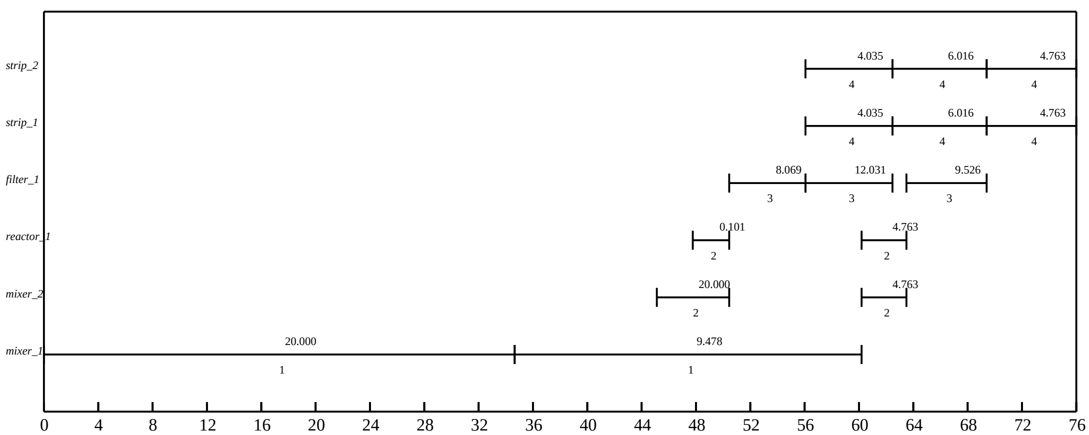

Figure 14: Alternative Schedule for Example 3

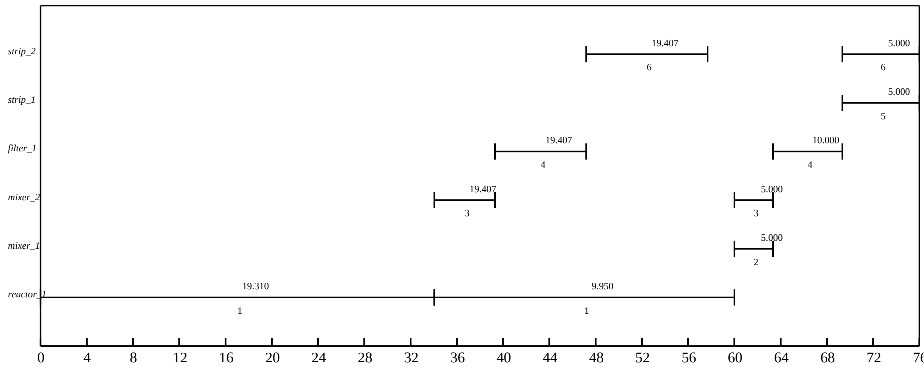

Figure 15: Consideration of minimum capacity requirements for Example 3

which reduce the number of binary variables from 42 to 22. Using this reduced model, we can obtain the optimal schedule in 0.8 CPU sec in a HP-C160 workstation solving 534 LP relaxation problems (without eliminating the zero $wv(i,n)$ variables).

7 Conclusions

In this paper, a novel continuous-time formulation is presented for the short-term scheduling of batch processes. It is shown that the proposed formulation results in smaller size mathematical models both in terms of continuous variables and constraints but primarily in terms of binary variables. The proposed approach has as a significant departure from previous approaches the decoupling of the unit events from the task events. This results in fewer binary variables and small integrality gaps. Moreover, better objective values can be easily accomplished since the models are easier to solve to optimality. It will be shown in the part II sequel paper that the approach can be extended to address continuous and semi-continuous plants involving batch as well as continuous processes and it will be applied to two industrial case studies.

Acknowledgments
The authors gratefully acknowledge financial support from the National Science Foundation and Mobil Technology Company.

Appendix A

A.1. Optimality issues

An additional example is considered here to further clarify the issues related to consideration of the event points in the proposed formulation. It should be highlighted that although there is a *fixed number of event points for all units* the starting and final times of different event points for different units *can be different.*

To clarify this point let's consider the following example involving three tasks that can take place in three different units as it is the case for the motivating example in our paper, Figure 16.

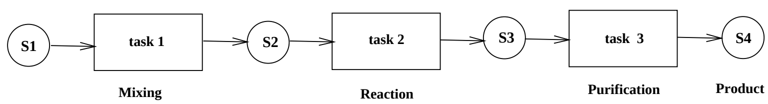

Figure 16: State Task Network representation for Example A1

The tasks and units however, have the characteristics shown in Table 11. In addition an initial amount of intermediate $s2$ equal to 10 te is available.

Units	**Capacity**	**Suitability**	**Processing Time**
Unit 1	10	task1	10.0
Unit 2	2	task2	2.0
Unit 3	2	task3	2.0
States	**Storage Capacity**	**Initial Amount**	**Price**
State 1	Unlimited	Unlimited	0.0
State 2	10	10.0	0.0
State 3	10	0.0	0.0
State 4	Unlimited	0.0	1.0

Table 11: Data for Example A1

Considering a time horizon of 20 hrs, one might think that the optimal schedule in the one shown in Figure 17, where the mixer produces 10 te of material $s2$ at the event point n_0 and the reactor and purifier subsequently simply process this amount of material and produce 2 te of product $s4$. However, using our proposed formulation the optimal schedule is obtained shown in Figure 18 that results in the production of 28 te of product $s4$. More specifically, note that simultaneously with the starting of the task 1 in the mixer the reactor starts processing the available initial amount of material $s2$ at t=0 and then the purifier follows as soon as the first batch of material $s3$ has been produced at t=2. After the mixer finished its first batch at time t=10, the reactor starts the processing of this amount of material. What should be emphasized here is that this time point corresponds to the event

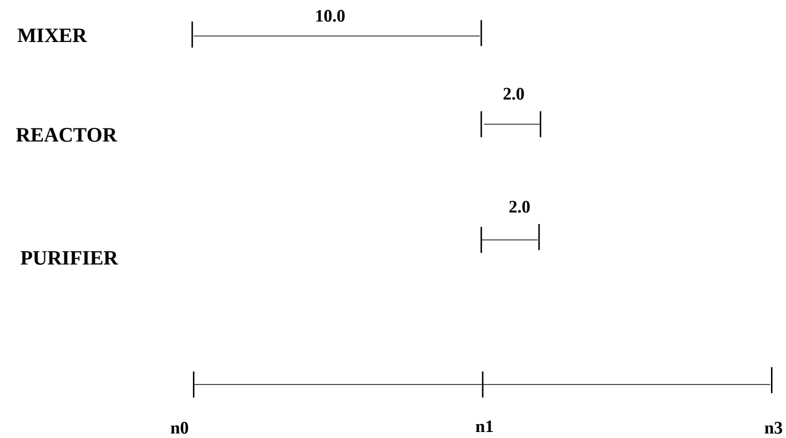

Figure 17: Suboptimal Gantt Chart for Example A1

point n_5 and *not* the event point n_1 as in the schedule shown in Figure 17, which corresponds to the suboptimal solution. To further clarify this point the detailed values of $wv(i,n)$ for the optimal schedule are provided in Table 12.

Optimal Solution ($wv(i,n)$)											
Task	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}
1	0	0	0	0	1	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	1	1	1	1	0

Table 12: Results for Example A1

Note also that the timing constraints for different tasks in different units, and in particular the *"mixing-reactions"* constraints, *do not* introduce any unnecessary restrictions to the problem since they are enforced only in the cases that *(a) the mixing is required for the subsequent reaction step posed by the material balances and (b) the reaction is performed in a subsequent event point of that of the mixing task.* For this example therefore, these event points are the (n_4), (n_5) and *not* the (n_0), (n_1) as it is the case for the schedule shown in Figure 17.

A.2. Feasibility issues

Although the detailed presentation regarding the incorporation of storage requirements is presented in section 2.1 of part II, a brief discussion of this issue is also included here considering a small example involving storage considerations.

The example is described by the STN of Figure 19. Tasks i_1, i_4 can be performed in unit 1, task i_2 is performed in unit 2, whereas tasks i_3, i_5 can be performed in unit 3. All units

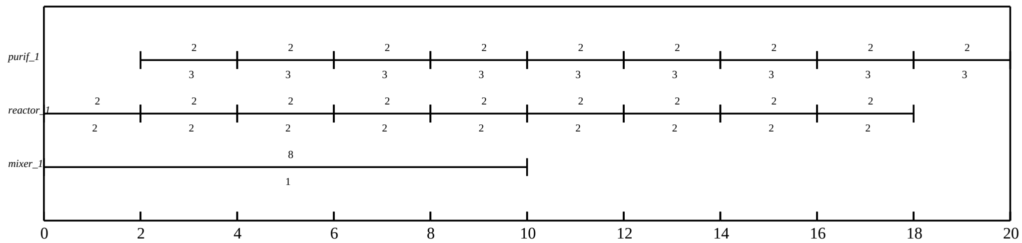

Figure 18: Gantt Chart for Example A1

have the same capacity of 10 te. The consideration of storage limitations for the intermediate states s_2 and s_3 requires the incorporation of two additional units, j_4 and j_5 the storage tanks for the storage of materials s2 and s3, respectively and two additional tasks task i_6, i_7 that correspond to the storage tasks.

To illustrate the capabilities of the proposed formulation three different cases are considered here. In the first case, two storage tanks with 10te capacity are considered for the storage of intermediates (s2, s3). $Tank_1$ is used for storing material s3 whereas the $tank_2$ for the storage of state s2. As explained in detail in section 2.1 of part II, the storage tasks are considered as batch tasks. Here, for simplicity of the presentation we assume that $\alpha_{i_7,j_5} = 0.0$ and $\beta_{i_7,j_5} = 0.0$ whereas for the storage task of state s3, (i.e., task i_6), $\alpha_{i_6,j_4} = 0.5$ and $\beta_{i_6,j_4} = 0.0$. In other words, we are not considering storage limitations for state s2 but we are considering such limitations for state s3 for which we assume that storage task is an additional batch task with fixed processing time. Note that the assumption of fixed processing time can be relaxed as presented in section 2.1 of part II. The scheduling problem of maximizing the production of all different products s4, s7, s8 is then solved using the proposed formulation. The optimal schedule shown in Figure 20 is obtained with an objective of 30 units.

In the second case, the capacity of the storage tank 1 used for storing state s3 is reduced to 8 te. In this case the solution of the proposed formulation results in the optimal schedule shown in Figure 21 with an objective of 28 units.

In the third case, the storage capacity of both storage tanks are considered to be zero. In this case the proposed formulation results in the optimal schedule shown in Figure 22 with an objective of 20 units.

Note that the proposed formulation results in the optimal schedule for all the different cases of limited storage capacity for the intermediate s3. The exact timing constraints that need to be considered are explained in great detail in section 2.1 of part II of this paper.

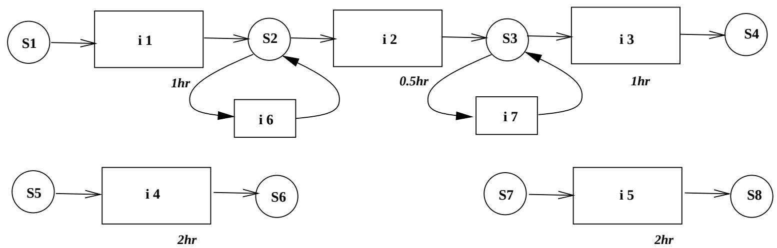

Figure 19: State Task Network representation for Example A2

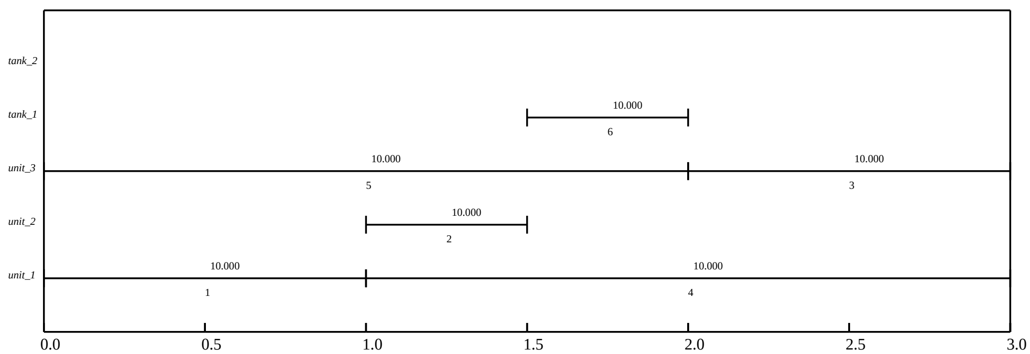

Figure 20: Gantt Chart for Example A2, storage tanks with 10te capacity

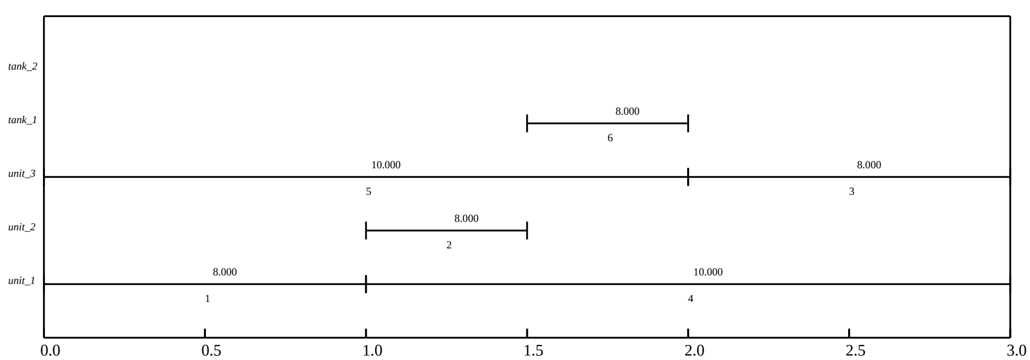

Figure 21: Gantt Chart for Example A2, storage tanks with 8te capacity

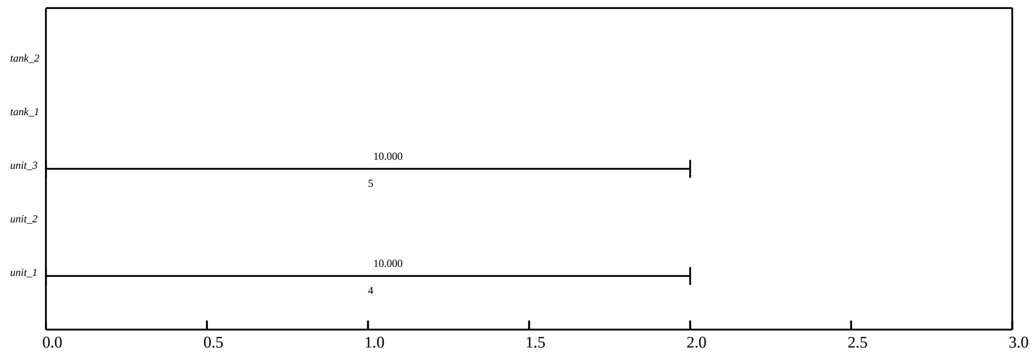

Figure 22: Gantt Chart for Example A2, storage tanks with 0te capacity

A.3. Number of Event Points

Although there is no generic procedure for the determination of the optimal number of event points, an additional example is presented here to illustrate the utilization of the simple procedure suggested in section 5.4 of this paper and to demonstrate the advances of the proposed formulation compared to the existing ones.

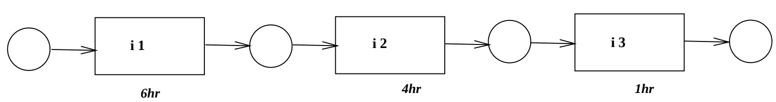

Figure 23: State Task Network representation Example A3

Let us consider Example A3, shown in Figure 23, where each task can take place in a different equipment item with capacity $10te$. The optimal schedule that maximizes the production of the final product 4, for the case of 3 event points is shown in Figure 24 and has an objective of 2 units. Considering 4 event points, the proposed formulation **identifies a better solution** shown in Figure 25 with objective of 4 units.

Different formulations[9;11] require the use of two more additional points to determine the same optimal schedule as shown graphically in Figure 26.

This distinctive advantage of the proposed formulation is based on the fact that the starting and final times of an event point *could be different depending on the unit* as shown in Figure 27.

Consideration of additional event points does not lead to further improvement of the objective function. The optimal schedule and detailed results of the proposed formulation are illustrated in Figures 24, 25, 28 and Tables 13, 14, 15, for 3,4, and 5 event points.

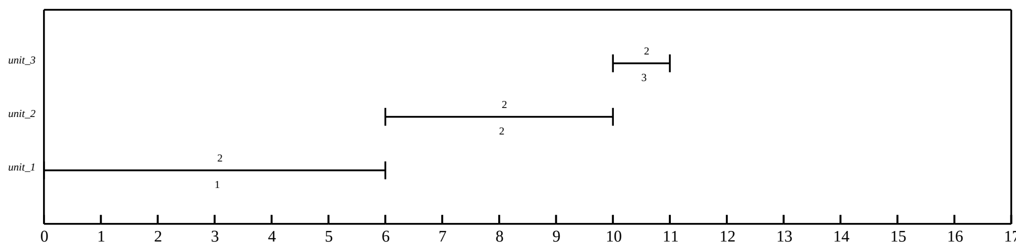

Figure 24: Gantt Chart for Example A3 with 3 event points

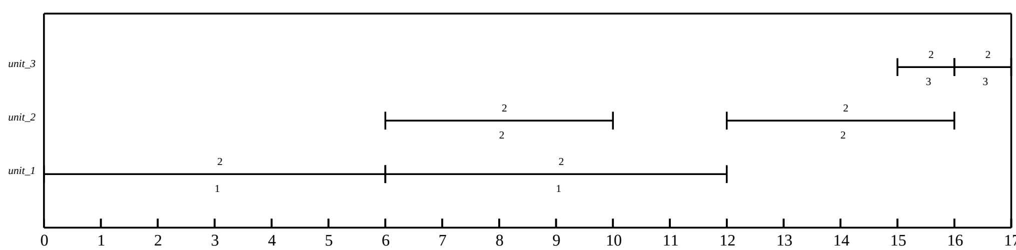

Figure 25: Gantt Chart for Example A3 with 4 event points

Optimal Solution $(wv(i,n))$				
Task	n_0	n_1	n_2	n_3
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0

Table 13: Results for Example A3 with 3 event points

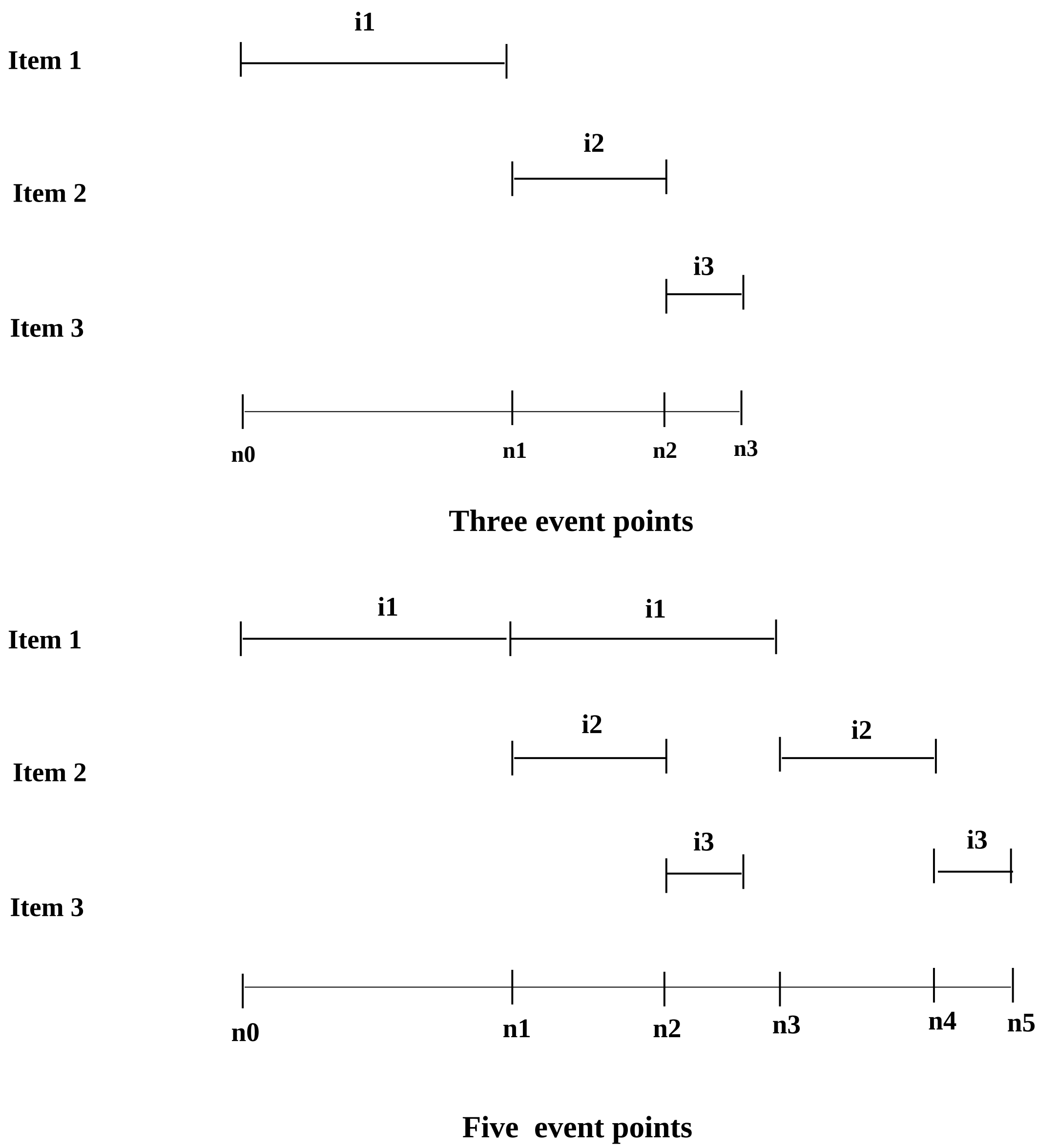

Figure 26: Event points consideration in existing scheduling formulations

Optimal Solution ($wv(i,n)$)					
Task	n_0	n_1	n_2	n_3	n_4
1	1	1	0	0	0
2	0	1	1	0	0
3	0	0	1	1	0

Table 14: Results for Example A3 with 4 event points

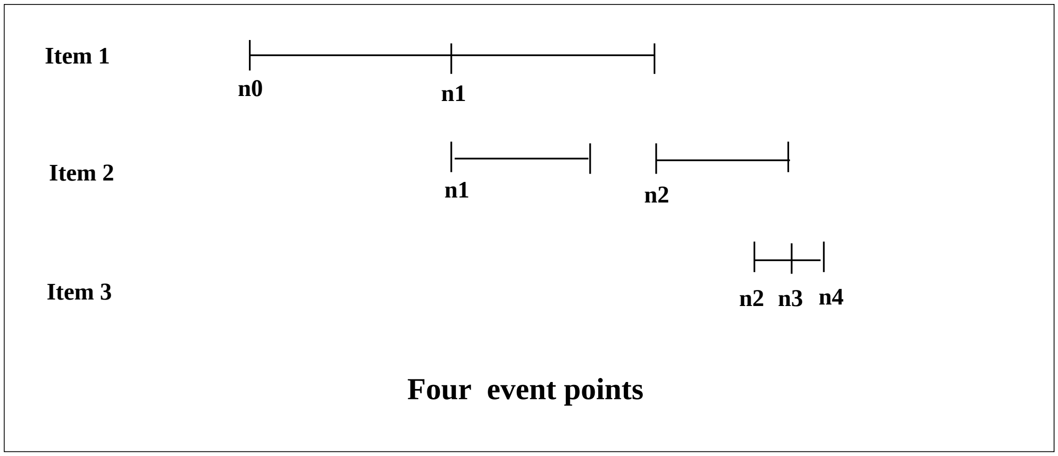

Figure 27: Event points consideration in our proposed formulation

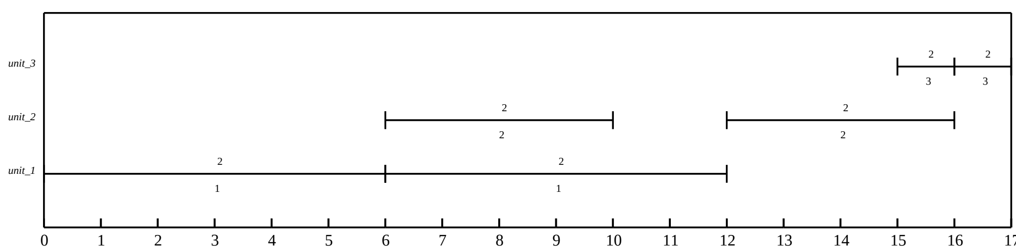

Figure 28: Gantt Chart for Example A3 with 5 event points

Optimal Solution ($wv(i,n)$)						
Task	n_0	n_1	n_2	n_3	n_4	n_5
1	1	1	0	0	0	0
2	0	1	1	1	0	0
3	0	0	1	1	0	0

Table 15: Results for Example A3 with 5 event points

References

(1) Reklaitis, G. Overview of Scheduling and Planning of Batch Process Operations. In *Presented at NATO Advanced Study Institute - Batch Process Systems Engineering, Antalya, Turkey* 1992 .

(2) Pantelides, C. C. Unified frameworks for the optimal proces planning and scheduling. In *Proc. Second Conf. on Foundations of Computer Aided Operations* 1994 253–274.

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

(4) Shah, N.; Pantelides, C. C.; Sargent, R. A General Algorithm for Short-Term Scheduling of Batch Operations - II. Computational Issues. *Comp. Chem. Engng.* **1993**, *17*, 229.

(5) Mockus, L.; Reklaitis, G. Mathematical Programming Formulation for Scheduling of Batch Operations using nonuniform time discretization. *AIChE annual meeting, San Francisco, California* **1994**, No. 235d.

(6) Mockus, L.; Reklaitis, G. A New Global Optimization Algorithm for Batch Process Scheduling. In *State of the Art in Global Optimization*; Floudas, C.; Pardalos, P., Eds.; Kluwer Academic Publishers, 1996, p 521–538.

(7) Pinto, J.; Grossmann, I. Optimal Cyclic Scheduling of Multistage Continuous Multiproduct Plants. *Comp. Chem. Engng.* **1994**, *18*, 797.

(8) Pinto, J.; Grossmann, I. A Continuous Time Mixed Integer Linear Programming Model for Short Term Scheduling of Multistage Batch Plants. *Ind. Eng. Chem. Res.* **1995**, *34*, 3037.

(9) Schilling, G.; Pantelides, C. C. A Simple Continuous-time Process Scheduling Formulation and a Novel Solution Algorithm. *Comp. Chem. Engng.* **1996**, *20*, S1221.

(10) Zhang, X. *Algorithms for optimal scheduling using nonlinear models*; PhD Thesis, University of London, 1995.

(11) Zhang, X.; Sargent, R. The optimal operation of mixed production facilities - general formulation and some solution approaches for the solution. In *Proc. of 5th Intl. Symp. on Process Systems Engineering, Kyongju, Korea* 1994 171–177.

(12) Karimi, I.; McDonald, C. Planning and Scheduling of Parallel Semi-Continuous Processes - Part II : Short-Term Scheduling. *Ind. Eng. Chem. Res.* **1997**, *36*, 2701.

(13) McDonald, C.; Karimi, I. Planning and Scheduling of Parallel Semi-Continuous Processes - Part I : Production Planning. *Ind. Eng. Chem. Res.* **1997**, *36*, 2691.

(14) Ierapetritou, M. G.; Floudas, C. Effective Continuous-Time Formulation for Short-Term Scheduling: II. Continuous and Semi-continuous Processes. *Ind. Eng. Chem. Res.* **1998**, this issue.