Mathematical Model for Scheduling Operations in Cascaded Continuous Processing Units

by

Sumit Kumar Bose
Doctoral Student, IIM Calcutta, Diamond Harbour Road Joka P.O. Kolkata – 700 104, India

&

Subir Bhattacharya
Professor, IIM Calcutta, Diamond Harbour Road Joka P.O. Kolkata – 700 104, India
Mathematical model for scheduling operations in cascaded continuous processing units

Subir Bhattacharya*, Sumit Kumar Bose
Indian Institute of Management Calcutta, Joka, Kolkata -700104, India

Abstract
The scenario under consideration involves $n$ cascaded continuous processing units responsible for processing $m$ product lines. Each product line needs to be processed by all the units in the same sequence, and has dedicated finite capacity storage tanks before and after every processing unit. A unit can process only one product line at a time. Inputs for all the product lines arrive continuously and simultaneously on the input side of the first unit in the sequence. There are multiple intermediate due dates for the final products. An optimal schedule for the units calls for a trade-off among spillage costs, upliftment failure penalties and changeover costs. A mathematical model is developed for the purpose and the resulting MINLP is linearized using standard techniques. The MILP has been tested using GAMS for three units and three product lines as encountered in a refinery situation. The model could output optimal schedules for a ten day scheduling horizon within reasonable time.

Keywords: Scheduling, Continuous processing units, Discrete optimization, Process industry

1 Introduction
Continuous processing units are quite common in refineries and other chemical processing industries. In continuous processing units, input streams for a product line are fed in continuously at one end and the output streams flow out simultaneously from the other end. A typical example would be a fractionating column. In contrast, in a batch processing unit, inputs in right amount and proportion are fed into the unit, ‘treated’ for a fixed amount of time in non-preemptive style, and outputs are taken out after the complete batch gets processed. In both the cases, the unit may be responsible for processing a range of products. In this situation, the unit works in a ‘blocked-out’ fashion, i.e., at any point of time it processes only one product line or stream. Input streams for the other product lines flowing in at the same point of time from

* Author for correspondence. Tel.: +91-33-24678300 Extn-409;
Email Address: subir@iimcal.ac.in
some upstream units have to be either stored in tanks for future processing or they have to be ‘spilled’ to lower valued products if there is no space in the tanks. Scheduling of ‘blocked-out’ batch processing units addresses the issue of determining the optimal sequence in which the product batches should be taken up for processing. In a continuous processing unit, one product line can be processed in number of stretches interleaved with the processing of other streams. The length of each stretch of each product line is a decision variable. The problem of scheduling a continuous processing unit involves deciding the number and durations of the stretches for each product, and interleaving these production stretches of the various product lines in an optimal manner. The authors encountered this problem while scheduling operations of the units responsible for producing lube oils in a refinery. Scheduling problems in batch processing industries have received good deal of attention in the literature. However, interest in scheduling units in continuous processing industry is more recent. This paper proposes a mathematical model for optimal scheduling of cascaded continuous processing units separated by fixed capacity buffers with multiple intermediate due dates for the finished products. The initial MINLP formulation was converted to MILP and tested using GAMS. The model could output optimal schedules for ten day horizons within reasonable time.

2 Problem Scenario

The set of units under consideration consists of a sequence of $n$ ‘blocked-out’ processing units, $U_1, U_2 \ldots U_n$, buffered by fixed capacity storage tanks to hold intermediate products (Figure 1). Each unit can process only one product line (also referred to as stream) at a time. Each product line has its exclusive set of tanks before and after every processing unit. We consider $m$ product lines, $P_1, P_2 \ldots P_m$, each of which requires to be processed by the $n$ units in the same sequence. The processing capacity (known as the feedrate measured in MtPD, Metric tons Per Day) of a unit for a product line is considered fixed, but varies from product to product. The processing of a unit involves splitting the feed with the help of reagents into intermediate streams according to a yield percentage fixed for a product line for the unit. The intermediate streams that are relevant for the final products under consideration get deposited in the tanks on the output side. What happens to the other output streams is beyond the purview of this discussion. For our purpose, if $p_{ij}$ is the yield percentage corresponding to product $P_i$ in unit
$U_j$, then $x$ unit of input to $U_j$ gives rise to $x \cdot p_{ij}$ units of output from $U_j$. Thus, each unit processes one stream at a time, taking its input from the corresponding input tank if it has enough stock and depositing the output into corresponding output tank if it has enough room (ullage). The presence of intermediate storage tanks obviates the need for the units to process the same product in tandem.

The finished products coming out of the last unit ($U_n$) also get stored in fixed capacity tanks to be uplifted according to some pre-specified upliftment schedule. A product can be uplifted several times by specified quantities during the scheduling horizon. Penalty is incurred if the required amount of a finished product is not ready on the specified due date. Shortfall in one upliftment of a product cannot be compensated by providing more during the next upliftment of the same product.

The inputs for the product lines arrive at the input tanks of Unit $U_1$ simultaneously at constant rates. The rates at which these inputs arrive depend on factors that are beyond the control of this block of units. Each of these streams gets stored in a fixed capacity tank if there is ullage in it; otherwise it spills, and is downgraded to lower valued products. Note that spillage is not allowed for intermediate or finished products. In essence, spillage can occur only at the input of $U_1$ because the inputs for all the product lines are arriving simultaneously and also because these units have no control on the rates at which these inputs arrive. Spillage implies opportunity lost, and hence, has a penalty associated with each unit of input stream spilt. Spillage can be reduced by quick changeovers. But a changeover from one product line to another has its associated cost and time. Thus, we have three factors to balance – the per unit spillage penalty for the input stream of each product line, the cost for changeovers, and the per unit penalty for failing to meet the upliftment schedule. Schedules of the units try to balance
these factors by processing each product line in number stretches during the scheduling horizon interleaved with the processing of other streams. As mentioned earlier, the length of a stretch is not fixed and each such stretch is called *run length*. Figure 2 is an example of possible run lengths in a unit considering three product lines or streams. Longer stretches of run are preferred to reduce changeover cost while shorter stretches may bring down the spillage penalty and/or the upliftment failure penalty.

![Figure 2: Example of runlengths in a unit](image)

The scheduling problem aims to find out an optimal mix of product sequences of varying run lengths so as to minimize the total cost/penalty.

### 3 Literature Review

Issues related to scheduling of operations in batch processing units have been discussed extensively in the literature. Overviews of planning and scheduling in process industries can be obtained in [1] and [2]. Review of the techniques available for optimizing the production schedules in the process industries is available in [3] and [4]. Grossman et al. ([5]) has suggested mixed integer mathematical programming methods for scheduling batch processes. [6] discusses the ways to reduce the computational complexity of the various mixed integer optimization models proposed in the literature. Kondili et al. in their seminal paper ([7]) propose ‘state task network’ representation for batch processes, and formulate a discrete time MILP for handling different kinds of scheduling problems arising in batch processing plants. The computational issues related to the MILP formulation presented by Kondili et al. are
discussed in [8]. Pantelides in [9] has introduced ‘resource task network’ (RTN) for process representation and has developed a discrete time optimization model for scheduling of batch plants based on RTN concept. [10] is a summary of the various assignment and sequencing models used for scheduling in process industry. Various scheduling objectives and performance evaluation criteria in the context of the chemical process industry have been discussed in [11]. Continuous time formulations for the short term scheduling problem in batch processes have been proposed in [12] and [13] among others. Compared to batch processing industry, interest in scheduling problems for continuous processing units is rather recent. Ierapetritou et al. ([14] and [15]) have formulated mathematical models for continuous processes taking into account multiple intermediate due dates for the finished products. [16] builds a mathematical model for cyclic scheduling of continuous processing plants with parallel units and decaying performance. Alle and Pinto ([17]) has developed a mathematical model for cyclic scheduling of multi product multistage continuous processing plants. The short term scheduling of refinery operations and mixed integer models based on continuous time formulations have been discussed in [18]. To the best of our knowledge there is no effective solution procedure in the literature to deal with the scheduling problem in multi stage, multi product continuous facilities with finite intermediate storage, multiple upliftment dates and simultaneous arrival of inputs. This work is an attempt to advance the research in this direction.

4 Mathematical Model

Given the scenario and a scheduling horizon of $H$ time periods, the aim is to determine the streams (product lines) to be processed in each time period in each unit so that total cost/penalty is minimized. A product $i$ taken up for processing in unit $j$ should be processed for at least $b$ consecutive time periods. This restriction is imposed to efficiently utilize the minimum amount of reagents that need to be added when changing over to product $i$. It may be recalled that the number of consecutive time periods during which a product is processed (run) in a unit is called the run length. Thus, run length of a product in a unit must be greater than or equal to the minimum run length $b$. For simplicity, we assume that the streams running in the units at the time period just before the start of the scheduling horizon have completed their minimum run length. That is, the units are free to changeover and can therefore take up streams
for processing different from the one being processed. This is not a restriction of the proposed model and, as will be evident later, can easily be relaxed. Changing over from one product to another requires some set up time. However, again for simplicity, we consider the changeover to be immediate. The changeover cost depends on the unit but is independent of the streams involved. Product-wise feedrates and yield percentages for the units are fixed data. The arrival rates of the inputs in front of (on the left of) $U_1$ and the upliftment schedule for the scheduling period are known at the time of planning the units’ schedule. The same product can be uplifted several times by different quantities during the given $H$ time periods. The upliftment schedule may specify upliftment of more than one product at the same time period.

To facilitate our discussion, we categorize the storage tanks into certain levels. The tanks to the left of $U_1$ belong to level 0 and the tanks immediately after a unit $U_k$ are said to be in level $k$ (see figure 1).

### 4.1 Notations

We will use the following sets, parameters and variables in our formulation.

*Table 1: Notations for the mathematical formulation*

<table>
<thead>
<tr>
<th>Sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>Stream identifier, $i \in [1,2,\ldots,m]$</td>
</tr>
<tr>
<td>$j$</td>
<td>Unit identifier, $j \in [1,2,\ldots,n]$</td>
</tr>
<tr>
<td>$t$</td>
<td>Time period, $t \in [1,2,\ldots,H]$</td>
</tr>
<tr>
<td>$l$</td>
<td>Level of tanks, $l \in [0,1,2,\ldots,n]$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_i$</td>
<td>Per unit spillage penalty of stream $i$</td>
</tr>
<tr>
<td>$a_i$</td>
<td>Rate of arrival of input stream $i$ in front of $U_1$</td>
</tr>
<tr>
<td>$C_{il}$</td>
<td>Capacity of tank corresponding to stream $i$ at level $l$</td>
</tr>
<tr>
<td>$q_j$</td>
<td>Cost of changeover from one stream to another in unit $j$</td>
</tr>
<tr>
<td>$u_i$</td>
<td>Per unit penalty for unfulfilled demand of stream $i$</td>
</tr>
<tr>
<td>$p_{ij}$</td>
<td>Percentage yield of stream $i$ in unit $j$</td>
</tr>
<tr>
<td>$d_{it}$</td>
<td>Demand of stream $i$ at the beginning of period $t$</td>
</tr>
<tr>
<td>$A_j$</td>
<td>Shutdown cost of unit $j$</td>
</tr>
<tr>
<td>$f_{ij}$</td>
<td>Feed rate of stream $i$ in unit $j$</td>
</tr>
<tr>
<td>$b$</td>
<td>Minimum runlength</td>
</tr>
</tbody>
</table>
Variables

\( S_{ijt} \): Amount of material in tank corresponding to stream \( i \) at level \( l \) at the beginning of time period \( t \)

\[
Y_{ijt} = \begin{cases} 
1 & \text{if stream } i \text{ is processed on unit } j \text{ in time period } t \\
0 & \text{otherwise}
\end{cases}
\]

\[
X_{it} = \begin{cases} 
1 & \text{if } S_{0it} + a_i > C_{io} \text{ and } Y_{ijt} = 0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
V_{ijt} = \begin{cases} 
1 & \text{if } Y_{ijt} = 1 \text{ and } Y_{ijt+1} = 0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
W_{it} = \begin{cases} 
1 & \text{if } (d_{it} - S_{int}) > 0 \\
0 & \text{otherwise}
\end{cases}
\]

\( Y_{ijt} \)'s are the binary decision variables. \( Y_{ijt} \) takes the value of 1 if stream \( i \) is processed in unit \( j \) during time period \( t \), otherwise it is 0. The binary variables \( X_{it} \), \( V_{ijt} \) and \( W_{it} \) represent spillage, changeover and upliftment failures respectively during time period \( t \). If stream \( i \) spills during time period \( t \), then \( X_{it} \) is 1, otherwise 0. The value of \( V_{ijt} \) is 1 if stream \( i \) is processed in unit \( j \) during time period \( t \) but is not processed during time period \( t+1 \), indicating changeover to some other stream \( i' \) in period \( t+1 \). Likewise, \( W_{it} \) equals 1 when the amount of stream \( i \) delivered falls short of the quantity \( d_{it} \) demanded in time period \( t \).

### 4.2 Constraints

#### 4.2.1 Storage constraints for level 0 tanks

Inputs for all the streams flow in simultaneously to level 0 tanks. An incoming stream collects in the designated tank and waits to be processed by \( U_1 \). If the tank gets filled up, the excess amount spills. For feasibility reasons, \( f_{rl} \) is much higher than \( a_i \) for all streams (see table 2 in section 6 for real-life data). The stock in the tank corresponding to stream \( i \) at the start of the time period \( t+1 \) is equal to the stock at the start of time period \( t \) adjusted by the amount arriving in period \( t \) minus the withdrawal that takes place in case the stream is processed by unit 1 in period \( t \). The expression for updating level 0 tank stocks is given by:
\[ S_{i_{t+1}} = X_i C_{i_0} + (1 - X_i)(S_{i_t} + a_i) - f_{i_1} Y_{i_1t} \quad \forall i, t \] (1)

If there is no spillage from the tank corresponding to the stream \( i \) in time period \( t \) \( (X_i = 0) \) and if stream \( i \) is also not processed by the unit in the period \( t \) \( (Y_{i_{1t}} = 0) \) then it reduces to \( S_{i_{t+1}} = S_{i_{tt}} + a_i \) indicating an increase in the stock position of \( i \) in the tank by an amount \( a_i \). If, however stream \( i \) is processed in unit 1 during time period \( t \) \( (Y_{i_{1t}} = 1) \), then \( X_i \) must be 0 (since \( a_i >> f_{i_1} \) ) and the equation reduces to \( S_{i_{t+1}} = S_{i_{tt}} + a_i - f_{i_1} \). If stream \( i \) has spilt during time period \( t \), then \( X_i = 1 \) and \( Y_{i_{1t}} = 0 \) and the stock at \( t+1 \) becomes \( S_{i_{t+1}} = C_{i_0} \). Note that situations with \( X_i = 1 \) and \( Y_{i_{1t}} = 1 \) cannot occur because of the way \( X_i \) has been defined.

4.2.2 Storage constraints for level 1 to level \( n-1 \) tanks

The stocks in the intermediate tanks in levels 1 to \( n-1 \) can be updated using the following:

\[ S_{il_{t+1}} = S_{il_t} + f_{j_l} p_{j} Y_{jlt} - f_{j_{l+1}} Y_{j_{l+1}t} \quad \forall i, t, l = 1, \ldots, n-1; \ j = l \] (2)

The stock at the start of time period \( t+1 \), \( S_{il_t} \), depends on the opening stock at time period \( t \), build up of stock by outflow \( (f_{j_l} * p_{j}) \) from the preceding unit if stream \( i \) has been processed by the unit in time period \( t \), and withdrawal of stock \( (f_{j_{l+1}}) \) by the next unit if it has processed stream \( i \) in time period \( t \).

4.2.3 Storage constraints for level \( n \) tanks

Stocks at the finished product tanks need to be updated by a separate expression to account for the upliftments and the possible shortfalls.

\[ S_{in_{t+1}} = (1-W_i)(S_{in_t} - d_i) + f_{in} p_{in} Y_{in} \quad \forall i, t \] (3)

Upliftment of product \( i \), if demanded in period \( t \), is met at the start of the period \( t \) and hence, production if any, of stream \( i \) during period \( t \) is not available to meet the demand. If there is a shortfall \( (S_{in_t} < d_i) \), then \( W_i \) becomes 1 and the first term becomes 0; otherwise \( W_i \) is 0 and
the stock is depleted by \( d_{it} \). If there is no upliftment of stream at time period \( t \), then both \( d_{it} \) and \( W_{it} \) are 0 and the first term reduces to \( S_{int} \).

### 4.2.4 Storage capacity constraints

The stock in any tank at any point of time cannot exceed its capacity and hence

\[
0 \leq S_{it} \leq C_{il} \quad \forall i, l, t
\]  

(4)

### 4.2.5 Changeover constraints

At any point of time, a unit can changeover to a maximum of one other stream. Thus,

\[
\sum_i V_{ijt} \leq 1 \quad \forall j, t
\]  

(5)

Definition of \( V_{ijt} \) in section 4.1 requires defining the following set of constraints:

\[
V_{ijt} \geq Y_{ijt} - Y_{ijt+1} Y_{ijt} \quad \forall i, j, t
\]

\[
V_{ijt} \leq \frac{(Y_{ijt} - Y_{ijt+1} + 1)}{2} \quad \forall i, j, t
\]  

(6)

These constraints force \( V_{ijt} \) to 1, when \( Y_{ijt} \) is 1 and \( Y_{ijt+1} \) is 0. For all other combinations of \( Y_{ijt} \) and \( Y_{ijt+1} \), the binary variable \( V_{ijt} \) assumes a value of 0. It may be noted that we count the changeover on the last period of the previous run. Changeover from stream \( i' \) to \( i \) is taken into account on the last period of stream \( i' \). So \( Y_{ijt} = 0 \) and \( Y_{ijt+1} = 1 \) need not be counted again.

### 4.2.6 Minimum run length constraints

A unit \( j \) that picks up processing of stream \( i \) cannot changeover to any other stream before stream \( i \) has been processed for \( b \) consecutive periods, the minimum run length. So we stipulate:

\[
\sum_{t'=t}^{t-b+1} Y_{ijt'} \geq bV_{ijt} \quad \forall t, j, i
\]  

(7)

By this constraint we do not allow \( V_{ijt} \) to assume a value of 1 until stream \( i \) has been processed for at least \( b \) periods including period \( t \). A run length can be greater than \( b \) as illustrated in
figure 2 and the actual length of runs for a particular stream $i$ in a unit $j$ will be dictated by the terms in the objective function and the rest of the constraints.

4.2.7 **Unit allocation constraints**

At most one stream can be assigned to unit $j$ for processing during period $t$. So

$$\sum_i Y_{ji} \leq 1 \quad \forall j, t \quad (8)$$

4.2.8 **Spillage constraints**

Definition of the binary variable $X_i$ in section 4.1 requires defining the following set of constraints:

$$-(S_{0i} + a_i - f_i * Y_{i1} - C_{i0}) \leq M * (1 - X_i) \quad \forall i, t$$

$$(S_{0i} + a_i - f_i * Y_{i1} - C_{i0}) \leq M * X_i \quad \forall i, t \quad (9)$$

where $M$ is a sufficiently large positive number. When $Y_{i1}$ is 1 then the term $S_{0i} + a_i - f_i - C_{i0}$ is negative (as $a_i < f_i$) so that only the first equation becomes relevant and forces $X_i$ to be 0. If however, $Y_{i1}$ is 0 and the term $S_{0i} + a_i - C_{i0}$ is positive indicating spillage, then only the second equation is relevant and forces $X_i$ to become 1.

4.2.9 **Demand constraints**

$W_i$ should be 1 when the availability of finished product $i$ at period $t$ is less than upliftment requirement at $t$, otherwise it is 0. We achieve this by specifying

$$-(d_i - S_{int}) \leq M * (1 - W_i) \quad \forall i, t$$

$$(d_i - S_{int}) \leq M * W_i \quad \forall i, t \quad (10)$$

$S_{int} \geq d_i$ indicates fulfillment of requirement. Only the first equation constrains the problem and $W_i$ takes the value 0. $S_{int} < d_i$ indicates partial fulfillment. Only the second equation becomes relevant and $W_i$ then takes the value 1.
4.3 Objective function

The objective is to develop a schedule that would lead to the least overall cost. The objective function to be minimized is given by:

\[
\text{Min } \sum_i \sum_j r_i (S_{io} + a_i - C_{i0} \cdot X_{it} + \sum_i \sum_j q_j (\sum_i V_{ij}) + \sum_i \sum_j u_i (d_{ia} - S_{ina}) W_{it} + \\
\sum_i \sum_j (1 - \sum_i Y_{ij}) A_j
\]  \tag{11}

The first term gives the cost of spillage, the second term represents the cost of changeover and the third term takes care of the penalties for upliftment failure. Shutdown of the units is prohibitively expensive as it involves considerable time and money to restore the unit to its proper working condition. Therefore every effort is made to prevent shutdown of units due to non-availability of stock or ullage of any of the streams. Even then, situations might arise where shutdown of a unit may be unavoidable and is therefore taken care of in the model by the last term in the equation of the objective function. The cost for shutdown is very high so that in a situation with large number of alternatives, the trade off is essentially among the first three terms of the objective function.

5 Model Simplification

Many of the constraints in the above formulation have nonlinear terms and the model is thus an MINLP model. In most of the cases non linearity involve bilinear products of continuous and binary variables. Equation (6), however, involves bilinear product of two binary variables. Employing standard linearization techniques it is possible to remove all the non-linearity from the proposed formulation. Bilinear products of binary and continuous variables are removed from the formulation by employing linearization techniques presented in [19 and 20]. A new nonnegative continuous variable \( \tilde{S}_{it} \) is defined to represent the bilinear term \( X_{it} \cdot S_{it0} \) in equation (1) and an additional set of constraints is imposed on the formulation.

\[ M \cdot X_{it} \geq \tilde{S}_{it} \geq S_{it0} + M \cdot X_{it} - M \quad \forall i, t \]  \tag{12}

\[ S_{it0} \geq \tilde{S}_{it} \quad \forall i, t \]  \tag{13}

Similarly a new nonnegative continuous variable \( \hat{S}_{it} \) is defined to represent the bilinear term \( W_{it} \cdot S_{it0} \) in equation (3) and an additional set of constraints is imposed.
Applying techniques of pure integer polynomial programming, the nonlinear 0-1 constraint represented by equation (6) is reduced to an equivalent 0-1 linear constraint. A new non-negative continuous variable $\bar{Y}_{ijt}$ is defined to represent the cross-product $Y_{ijt} \ast Y_{ijt}$ and the following set of three constraints is added:

\begin{align}
Y_{ijt+1} + Y_{ijt} - \bar{Y}_{ijt} &\leq 1 \quad \forall i, j, t \tag{16} \\
\bar{Y}_{ijt} &\leq Y_{ijt} \quad \forall i, j, t \tag{17} \\
\bar{Y}_{ijt} &\leq Y_{ijt+1} \quad \forall i, j, t \tag{18}
\end{align}

A close look at the model would reveal that the changeover costs depend on the units but do not take into account the sequence dependencies. In other words, the streams involved in the changeover do not influence the cost. So we propose to represent changeovers by $V_{jt}$ rather than $V_{ijt}$. The value of $V_{jt}$ is 1 if there is a changeover in unit $j$ at time $t$, 0 otherwise. Then the pair of equations in equation (6) of the original model needs to be replaced by equation (19) as:

\begin{equation}
V_{jt} = \sum_i Y_{ijt} - \sum_i (Y_{ijt} \ast Y_{ijt+1}) \quad \forall j, t \tag{19}
\end{equation}

The above equation would then need to be reformulated using the $\bar{Y}_{ijt}$ variables as

\begin{equation}
V_{jt} = \sum_i Y_{ijt} - \sum_i \bar{Y}_{ijt} \quad \forall j, t \tag{20}
\end{equation}

Equation (20) along with equations (16), (17) and (18) then redefine the changeover constraint. Additionally, the minimum run length constraints in equation (7) also need to be remodeled as:

\begin{equation}
\sum_{t'=t}^{t+b-1} Y_{ijt'} \geq bV_{jt}Y_{ijt} \quad \forall i, j, t \tag{21}
\end{equation}

Further computational gains can be obtained by taking note of the fact that the value of $V_{jt}$ in equation (20) depends solely on the binary variable $Y_{ijt}$. The structure of the equation (20) is such that $V_{jt}$, even if it is declared continuous, can assume values nothing other than 1 or 0. Thus, declaring $V_{jt}$ variables to be continuous rather than binary results in significant
computational gains. Using linearization techniques, the term $V_{jt}Y_{ijt}$ in equation (21) needs to be replaced by the continuous variables $\bar{V}_{ijt}$ and the following constraints need to be added:

\[
M \ast Y_{ijt} \geq \bar{V}_{ijt} \geq V_{jt} + M \ast Y_{at} - M \quad \forall i, j, t
\]  

(22)

\[
V_{jt} \geq \bar{V}_{ijt} \quad \forall i, j, t
\]  

(23)

The resulting final formulation is a mixed integer linear programming problem and would be given as:

Minimize

\[
Z = \sum \sum r \left( S_{it} + a_i X_{at} - C_{it} X_{at} \right) + \sum \sum q_j (V_{jt}) + \sum \sum u_i (d_{at} W_{at} - \tilde{S}_{it}) + \sum \sum (1 - \sum j Y_{ijt}) A_j
\]  

(24)

subject to:

\[
M \ast X_{at} \geq \tilde{S}_{at} \geq S_{10r} + M \ast X_{at} - M \quad \forall i, t
\]  

(25)

\[
S_{10r} \geq \tilde{S}_{it} \quad \forall i, t
\]  

(26)

\[
M \ast W_{it} \geq \tilde{S}_{it} \geq S_{at} + M \ast W_{at} - M \quad \forall i, t
\]  

(27)

\[
S_{at} \geq \tilde{S}_{it} \quad \forall i, t
\]  

(28)

\[
-(S_{10r} + a_i - f_i \ast Y_{rlt} - C_{it}) \leq M \ast (1 - X_{at}) \quad \forall i, t
\]  

(29)

\[
(S_{10r} + a_i - f_i \ast Y_{rlt} - C_{it}) \leq M \ast X_{at} \quad \forall i, t
\]  

(29)

\[
V_{jt} = \sum_i Y_{ijt} - \sum_i \bar{Y}_{ijt} \quad \forall j, t
\]  

(30)

\[
Y_{ijt} + Y_{ijt} - \bar{Y}_{ijt} \leq 1 \quad \forall i, j, t
\]  

(31)

\[
\bar{Y}_{ijt} \leq Y_{ijt} \quad \forall i, j, t
\]  

(32)

\[
\bar{Y}_{ijt} \leq Y_{ijt+1} \quad \forall i, j, t
\]  

(33)

\[
-(d_{at} - S_{int}) \leq M \ast (1 - W_{at}) \quad \forall i, t
\]  

(34)

\[
(d_{at} - S_{int}) \leq M \ast W_{at} \quad \forall i, t
\]  

(35)

\[
\sum_{i=1}^{r-1} Y_{ijt} \geq b \bar{V}_{ijt} \quad \forall i, j, t
\]  

(36)

\[
M \ast Y_{ijt} \geq \bar{V}_{ijt} \geq V_{jt} + M \ast Y_{at} - M \quad \forall i, j, t
\]  

(36)
\[ V_{jt} \geq \bar{V}_{jt} \quad \forall i, j, t \]  
\[ S_{i0t+1} = X_{it} C_{i0} + S_{i0t} + a_i - \bar{S}_{it} - a_i * X_{it} - f_{it} Y_{it} \quad \forall i, t \]  
\[ S_{ilt+1} = S_{ilt} + f_{ij} P_{ij} Y_{ij} - f_{ij+1} Y_{ij+1t} \quad \forall i, t, l = 1, \ldots, n-1; j = l \]  
\[ S_{int+1} = S_{int} - d_{it} - \hat{S}_{it} + W_{it} * d_{it} + f_{in} P_{in} Y_{in} \quad \forall i, t \]  
\[ 0 \leq S_{ilt} \leq C_{tl} \quad \forall i, l, t \]  
\[ V_{jt} \leq 1 \quad \forall j, t \]  
\[ \sum_{i} Y_{ij} \leq 1 \quad \forall j, t \]  

The above MILP model can be solved using any of the standard packages.

6 Computational Experience

The proposed MILP formulation was modeled in GAMS and was run using the XA solver embedded in GAMS on a 2.4 GHz Pentium 4 workstation with 1 GB RAM. The fixed data (Table 2) used for the test cases are real data collected from a refinery situation involving three product lines \( m = 3 \) and three cascaded processing units \( n = 3 \). The per unit spillage costs and the per unit upliftment failure penalties have been masked. However, their relative importances have been reflected in the values used. The tank capacities are assumed to be 10000 Mt for easy comparison of the input data for the cases considered.

The model has been tried out on 13 cases. For each case, the scheduling horizon was 10 time periods (day 1 to day 10). The input of a case consists of the initial stock positions at all the four levels, the upliftment schedule and information about the streams running in the units on the day just before the start of the scheduling horizon (day 0). The last information is required to account for changeover(s) at the start of horizon, i.e., if a unit chooses to run a stream on day 1 different from the stream running on day 0, a changeover is counted. As mentioned earlier, for simplicity, we have assumed that the streams running on day 0 have completed their minimum runlengths. The model can easily take care of incomplete minimum runlengths by noting the number of days for which the last stream has been running on a unit \( j \). For example, if the unit has started processing a stream on day 0 itself then \( V_{j0} \) and \( V_{ji} \) can be forced to 0 so...
that changeover is not possible before day 2 (with \( b = 3 \)). The initial tank stocks taken as inputs are stocks at the beginning of day 0.

Table 2: Fixed data

| Number of Units: 3 \( \{U_1; \text{FEU}; U_2; \text{SDU}; U_3; \text{HFU}\} \) |
| Number of Streams: 3 \( \{1: \text{IO}; 2: \text{HO}; 3: \text{DAO}\} \) |
| Minimum Run Length \( (b) \): 3 |
| Capacity (Mt): 10000 (For all streams for all levels) |
| Scheduling Horizon(\( H \)): 10 |
| Changeover cost \( (q_j) \): 20 (Same for all \( j \)) |

<table>
<thead>
<tr>
<th>Stream</th>
<th>Input Rate ( (a_i) ) (MtPD)</th>
<th>Feed Rate ( (f_{ij}) ) (MtPD)</th>
<th>Yield Percentage ( (p_{ij}) )</th>
<th>Spill Penalty ( (r_i) )</th>
<th>Upliftment Penalty ( (u_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>620</td>
<td>1250</td>
<td>578</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>255</td>
<td>1030</td>
<td>576</td>
<td>50</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>310</td>
<td>1050</td>
<td>521</td>
<td>65</td>
<td>14</td>
</tr>
</tbody>
</table>

To generate the test cases, we first designed a base case as given in table 3. The inputs of the base case are chosen in a manner so as to require collaboration among the units to come up with the least cost schedule. A close look at table 3 data would reveal that the initial tank stocks at level 0 at the start of day 0 are such that some spillage is inevitable. \( U_1 \) processed stream 1 on day 0. The input rate column in table 2 and the level 0 stocks in table 3 considered together show that streams 1, 2 and 3, if not processed in \( U_1 \) from day 1 onwards, would start spilling on day 6, day 3 and day 1 respectively.

Table 3: Base case input data

<table>
<thead>
<tr>
<th>Tank Stock Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streams</td>
</tr>
<tr>
<td>Streams</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheduled Uplifements (Mt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

| Last Run: (Unit: Stream) : \( (U_1; 1) \); \( (U_2; 2) \); \( (U_3; 3) \) |

For the problem with 3 streams, 3 processing units and 10 time periods, the formulation involves 150 binary variables – 90 \( Y_{ij} \) s, 30 \( W_i \) s and 30 \( X_a \) s. However, a significant number of
$W_i$ and $X_i$ variables can be assigned a priori. As the due dates are known, $W_i$ can be fixed to 0 for the periods in which there is no upliftment of stream $i$. Additionally, since the level of the tank stock at the beginning of the scheduling horizon is known, it is possible to determine the time from the start of the scheduling horizon when the streams will start spilling if not processed. Therefore, for all periods prior to the period when stream $i$ may start spilling $X_i$ can be assigned 0. The optimal solution for the base case data was found in about one and half hours after 9940393 iterations. The objective function value is 5374.72. Figure 3 is the Gantt chart of the Units’ schedules as suggested by the model.

Table 4 summarizes our experiences with 12 other case instances. These cases were generated by varying the input parameters of the base case. Each case was obtained by altering the initial stock positions of level 0 tanks and/or the upliftment schedule. The cases represent all possible combinations of four different level 0 initial stock positions and three different upliftment schedules. For all the cases the time limit was set to 8 hrs. The solutions have been rounded to second decimal place.
Table 4: Model output for 12 case instances

<table>
<thead>
<tr>
<th>Case No.</th>
<th>Stock Position at start of day 0 ('00 Mt)</th>
<th>Scheduled Upliftments ('00 Mt)</th>
<th>Final Solution Cost</th>
<th>Iteration count</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Levels</td>
<td>Day</td>
<td>Streams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>50 30 20 10</td>
<td>10</td>
<td>22 40 15</td>
<td>4194.77</td>
<td>142091</td>
</tr>
<tr>
<td>1</td>
<td>70 50 50 10</td>
<td>8</td>
<td>- 15 -</td>
<td>3672.23</td>
<td>58527</td>
</tr>
<tr>
<td>2</td>
<td>65 50 50 20</td>
<td>10</td>
<td>- 22 -</td>
<td>3692.23</td>
<td>749124</td>
</tr>
<tr>
<td>3</td>
<td>50 30 20 10</td>
<td>6</td>
<td>- 15 -</td>
<td>3692.23</td>
<td>749124</td>
</tr>
<tr>
<td>4</td>
<td>90 50 50 10</td>
<td>10</td>
<td>22 40 15</td>
<td>4214.77</td>
<td>382365</td>
</tr>
<tr>
<td>5</td>
<td>65 50 50 20</td>
<td>8</td>
<td>- 15 -</td>
<td>3692.23</td>
<td>985739</td>
</tr>
<tr>
<td>6</td>
<td>90 50 50 10</td>
<td>6</td>
<td>- 15 -</td>
<td>3712.23</td>
<td>1397379</td>
</tr>
<tr>
<td>7</td>
<td>65 50 50 20</td>
<td>10</td>
<td>22 40 15</td>
<td>4214.77</td>
<td>22242643</td>
</tr>
<tr>
<td>8</td>
<td>70 50 50 10</td>
<td>8</td>
<td>- 15 -</td>
<td>3692.23</td>
<td>45341922</td>
</tr>
<tr>
<td>9</td>
<td>65 50 50 20</td>
<td>6</td>
<td>- 15 -</td>
<td>3712.23</td>
<td>59067853</td>
</tr>
<tr>
<td>10</td>
<td>50 30 20 10</td>
<td>10</td>
<td>22 40 15</td>
<td>4194.77</td>
<td>97080</td>
</tr>
<tr>
<td>11</td>
<td>50 50 50 20</td>
<td>8</td>
<td>- 15 -</td>
<td>3672.23</td>
<td>329198</td>
</tr>
<tr>
<td>12</td>
<td>50 30 20 10</td>
<td>6</td>
<td>- 15 -</td>
<td>3692.23</td>
<td>105501</td>
</tr>
</tbody>
</table>

## Conclusion

In this paper we have presented a MILP formulation for a scheduling problem encountered in many petrochemical industries deploying continuous processing units. The proposed discrete time model could solve scheduling problems involving 3 units and 3 product lines for a horizon of 10 time periods. The model needs to be improved to tackle problems with longer scheduling
horizons. The model also needs to be extended to subsume changeover time and sequence dependent changeover costs. Efforts are on to represent the problem as a State Task Network ([7]). A continuous time formulation of the problem is also on the anvil. We hope that the proposed model will elicit interest among researchers working in related areas to extend and modify it in order to suit the requirements of the planners in continuous processing industries.

Reference:


