15th Triennial World Congress, Barcelona, Spain

Short-Term Scheduling of Multipurpose Batch Plants Based on mSTN

†Chen Changling[1], Zong Xue Jun[2], Shao Huihe[1], Liu Changling[1]

[1] *Institute of Automation, Shanghai Jiao Tong University, Shanghai, 200030, P. R. China*
[2] *Shenyang Institute of Chemical Technology, Shenyang, 110021, P. R. China*

Abstract: This paper presents a mixed integer linear programming (MILP) mathematic model for short-term scheduling of multipurpose batch plants based on the representation of Maximum State Task Network (mSTN) for process recipe. When the model is built, the assignment of tasks and units to event points is handled through one set of binary variables and the continuous-time representation is used. This model can accommodate dedicated, multipurpose, and shared storage modes and can tackle all storage policies. In order to improve the computational efficiency, some effective techniques are developed to cut down the size of the proposed model based on the characteristics of batch plants and the model. Examples reveal the effectiveness and applicability of the model and improvement techniques. *Copyright © 2002 IFAC*

Keywords: short-term scheduling; mSTN; multipurpose batch plant; MILP

1. INTRODUCTION

Multipurpose batch plants are general facilities where a wide variety of products can be manufactured by sharing available equipment. The operation flexibility inherent in such plants is what makes them popular in situation where the product demands fluctuate with time. However, this characteristic also challenges the operator to schedule the production activity accounting for the available resources (Rippin, 1993). The effective scheduling of batch plants can not only increase customer services, lower inventories, and reduce needs for excess capacities but also provide mechanistic understandings of processes and potential of identifying key data (Pekney and Reklaitis, 1998). So, research on scheduling of multipurpose batch plants has substantial engineering and theoretical significance.

Mixed-integer linear programming (MILP) has become one of important approaches to batch plant scheduling for its good quality solution and fair usability and extensibility (Pekney and Reklaitis, 1998). However, the MILP model based on uniform discrete-time representation (Kondili et al., 1993; Papageoriou and Pantelides, 1996; Rodrigues and Rodrigues, 2000) suffers from the following drawbacks (Shah, 1998): (1) The time representation is approximate; (2) The model comprises a large number of binary variables and linear constraints when the problem of industrial size is considered. (3) It is difficult to model operation where the processing time is dependent on the batch size. Hence, the continuous-time representation is widely used in research of batch process scheduling nowadays. Schilling and Pantelides (1996) presented a scheduling model for multipurpose batch plants based on continuous-time representation and resource task network (RTN). The model reveals a large integrality gap and Schilling and Pantelides used a special branch and bound technique to solve the problem. Ierapetritou and Floudas (1998a,b; 1999) proposed MILP models for batch, semicontinuous and continuous plants accounting for due dates of products and intermediates based on state-task network (STN). However, because of the shortcomings inherent in STN and RTN (Barbosa-Povoa and Macchietto,1994), the scheduling models based on them may give ambiguous scheduling results.

In general, the plant utilization and production rate can be enhanced by sufficient utilizing the storage capacity of all equipment. However, there is no clear distinction between the processing and storage equipment. On the basis of the way the storage equipment are used for holding materials, the storage modes can be put into three classes, namely dedicated, multipurpose, and shared storage (Papageoriou and Pantelides, 1996). However, the storage modes are not well dealt with in existing scheduling documents.

In this paper, an MILP mathematic model is proposed for scheduling of multipurpose batch plants based on the representation of Maximal State Task Network (mSTN) for process recipe. When the model is built, the assignment of tasks and units is handled through one set of binary variables and the continuous-time representation is used. Moreover, the storage modes of equipment are accounted for. The paper is organized as follows. The mSTN is introduced and the problem statement of this paper is given in Section 2. Section 3 presents the mathematic model. Afterwards, some techniques are proposed to reduce the size of the model in section 4. In section 5, some examples are studied to illustrate the

Financial support from Project "973" of the State Key Fundamental Research under grant G1998030415.
†Corresponding author.
E-mail address: clchen72@sina.com

effectiveness and applicability of the model and the improvement techniques.

2. BATCH PLANT REPRESENTATION AND PROBLEM STATEMENT

STN and RTN, as traditional recipe representations for batch plants, leave two ambiguities (Crooks, 1992). First, transfers of materials between any equipment are assumed to be possible. In fact, there are no connections between some equipment. Second, it is not clear whether material produced at end of a task is actually residing in the vessel allocated to the task, in a separate storage tank, or in a downstream vessel carrying out subsequent task.

In view of the above shortcomings, Crooks (1992) extended STN to a more general recipe representation for batch plants, namely maximal state task network (mSTN), accounting for connections between equipment. Five different types of nodes characterise this representation: $\text{eTask}_{i,j}$ (e.g. $\text{eTask}_{T1/1a}$)-processing/auxiliary tasks i which can be perform in unit j; $\text{eState}_{s,j}$ (e.g. $\text{eState}_{S3/1a}$)– state s which can be stored in a vessel j; iState and oState – which sever uniquely to identify the origin and location of the material and are characterised by zero capacity; tTask_{π} – transfer task which introduced between two state instances if there is a direct link between the units associated with these states. For example, a batch plant shown in Fig. 1 can be represented by mSTN in Fig. 2. It is noted that if a material produced by an eTask in a unit can be stored in a dedicate storage unit, it is unnecessary to store the material in any other unit. This is because the capacity of dedicate storage units are usually great enough in practice. Moreover, because units 2a and 2b are not suitable to store material produced by $\text{eTask}_{T1/1a}$ (i.e., there is no possibility of transferring material produced by $\text{eTask}_{T1/1a}$ between units 2a and 2b), the material produced by $\text{eTask}_{T1/1a}$ can only be store in unit 1a.

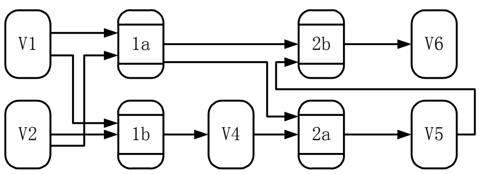

Fig. 1 Plant flowsheet

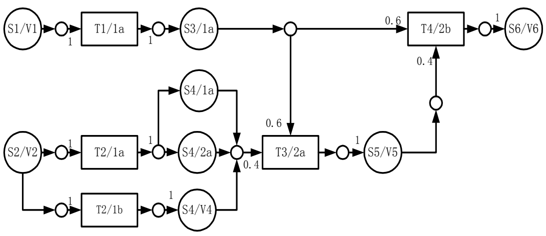

Fig. 2 Maximum state task network

It is noted that mSTN can be used to check the redundant allocation of eTasks to units. For instance, in the above example plant unit V4 is not suitable to hold the material produced by $\text{eTask}_{T1/1b}$, so eTask T1 can not be allocated to unit 1b. If STN is used to depict this batch plant, error can never be avoided. However, the scheduling based on mSTN is more complicated than that based on STN. The reason is that when developing the scheduling model base on mSTN the connections between equipment have to be taken into consideration. So far, there have been few documents based on mSTN. Barbosa-Povoa and Macchietto (1994) proposed a model based on mSTN using uniform discrete time representation, and the storage models and storage policies are not well tackled.

According to the above definition of mSTN, the problem statement can be given as follows: Assume: (1) non-preemptive processing – once started, processing activity must be processed until completion; (2) material transfer times are neglected; (3) once an eTask is completed, the materials go to downstream storage tank. Given (1) scheduling time horizon; (2) the flowsheet of batch plant (including the connections between units, processing steps, and proportion of each material consumed or produced by any eTask); (3) the parameters (including the capacity, mean value of processing time) of any eTask processing in a unit; (4) the parameter (capacity) of a storage tank; (5) the available resources and market demands. Determine: (1) sequence of the eTasks processed in the same units; (2) processing time of eTasks; (3) timing of individual eTask; (4) selection of resources in storage units to execute tasks and determination of unit to store the product and intermediates accounting for connections.

3. MATHEMATIC MODEL

In this part, an MILP mathematic model for short-term scheduling of multipurpose batch plant is developed. The model can be charactetised by the following basic ideas:

Model Based on mSTN. The proposed model is based on mSTN and it can give more detailed scheduling results without ambiguity in contrast with that based on STN and RTN.

Continuous-Time Representation. The proposed mathematic model is based on continuous-time representation. It requires initial consideration of necessary number of event points corresponding to beginning of eTasks. The timings of these event points are determined by optimization.

Handling Assignment of Units and eTasks through One Set of Binary Variable. If eTask i is performed in unit j at event point n , $W_{i,j,n}=1$; otherwise it is 0.

Variable Processing Time. Processing times vary with the amount of the materials being processed.

Accommodating Multiplicate Storage Modes and Storage Policies. The model proposed in this paper can accommodate dedicated, multipurpose, and shared storage modes. Moreover, the model can deal with all storage policies, i.e. UIS (Unlimited Intermediate Storage), FIS (Finite Intermediate Storage), NIS (No Intermediate Storage) and ZW (Zero Wait) storage policy.

Before introducing the proposed mathematic scheduling models for multipurpose batch plants, the following sets and parameters are defined firstly.

Subindices

i,i' = processing tasks (eTasks); j, j' = units;
n = event point; $\overline{n}$ = the last event point;
s = eState;

Sets

I = set of processing tasks; J = set of units;
N = set of event points; S = set of eStates;
S_i^{in} / S_i^{out} = set of eStates which store materials produced/consumed by eTask i

$I_j\,/\,S_j$ = set of processing/storage tasks which can be performed in unit j;
$J_i\,/\,J_s$ = set of units which are suitable for performing processing task i / storing material in eState s;
I_s^p / I_s^c = set of eTasks which produce/consume material in eState s;

Parameters

$C_{s,j}^{\max}$ = maximum storage capacity of unit j when storing the material in eState s;
$V_{i,j}^{\max}\,/\,V_{i,j}^{\min}$ = maximum / minimum capacity of unit j when processing task i;
$\rho_{s,i}^{p}\,/\,\rho_{s,i}^{c}$ = proportion of material in eState s produced/ consumed by processing task i;
$\alpha_{i,j}\,/\,\beta_{i,j}$ = constant/variable term of processing time of eTask i in unit j;
$\Pr_s$ = price of material in eState s;
H = scheduling time horizon;
r_s = market demand of material in eState s;

Variables

$W_{i,j,n}$ = binary variable denoting that unit j is allocated to performing eTask i at event point n;
$Y_{s,j,n}$ = binary variable denoting that unit j is assigned to storing material of eState s at event point n;
$B_{i,j,n}$ = amount of material which starts undertaking processing task i in unit j at event point n;
$ST_{s,j,n}$ = amount of material in eState s which is stored in unit j at event point n;
$D_{s,j,n}$ = amount of material in eState s in unit j delivered to market at event point n;
$Ts_{j,n}/Te_{j,n}$ = time that unit j start/complete performing an eTask at event point n;
$Tsi_{s,n}$ = minimum time that the eTasks in set I_s^c can be started at event point n.
$O_{s,j,i,j',n}\,/\,P_{s,j,i,j',n}$ = the amount of material in eState s in unit j which is consumed/produced by eTask i in unit j' at event point n

It is noted that only one eState corresponds to each material in the mSTN. However, a material can be stored in several units accounting for connections. If a material is stored in a unit, a storage task occurs.

On the basis of the idea and nomenclature, the proposed model can be formulated as follows:

Allocation Constraints

$$\sum_{i\in I_j} W_{i,j,n} + \sum_{s\in S_j} Y_{s,j,n} \le 1 \quad \forall j\in J, n\in N \qquad (1)$$

$$\sum_{j'\in J_s} Y_{s,j',n+1} - W_{i,j,n} \ge 0$$
$$\forall i\in I, j\in J, s\in S_i^{in}, n\in N \qquad (2)$$

Constraints in eq (1) express that unit j can undertake at most one processing task or storage task at any event point. Constraints in eq (2) state that if eTask i is performed in unit j at event point n, each material produced by this eTask has to be held in at least one unit at event point $n+1$.

Capacity Constraints

$$V_{i,j}^{\min} W_{i,j,n} \le B_{i,j,n} \le V_{i,j}^{\max} W_{i,j,n} \quad \forall i\in I, j\in J, n\in N \qquad (3)$$

Eq (3) states that if eTask i is performed in unit j at event point n, the amount of materials is bounded by maximum and minimum capacity of that unit; otherwise, it has to equal 0.

Storage Constraints

$$ST_{s,j,n} \le C_{s,j}^{\max} Y_{s,j,n} \qquad \forall s\in S, j\in J_s, n\in N \qquad (4)$$

Eq (4) expresses that if unit j is allocated to storing material in eState s at event point n, the amount can't exceed the maximum storage capacity.

Material Balances

$$ST_{s,j,n} = ST_{s,j,n-1} - D_{s,j,n} + \sum_{i\in I_s^p}\sum_{j'\in J_i} P_{s,j,i,j',n-1}$$
$$+ \sum_{i\in I_s^c}\sum_{j\in J_i} O_{s,j,i,j',n}$$
$$\forall s\in S, j\in J_s, n\in N\setminus\{1\} \qquad (5)$$

$$ST_{s,j,1} = STin_{s,j} + \sum_{i\in I_s^c}\sum_{j'\in J_i} O_{s,j,i,j',1} \quad \forall s\in S, j\in J_s \qquad (6)$$

$$ST_{s,j,\bar{n}+1} = ST_{s,j,\bar{n}} - D_{s,j,\bar{n}+1} + \sum_{i\in I_s^p}\sum_{j'\in J_i} P_{s,j,i,j',\bar{n}}$$
$$\forall s\in S, j\in J_s \qquad (7)$$

$$P_{s,j,i,j',n} \le Y_{s,j,n} V_{i,j'}^{\max} \quad \forall s\in S, j\in J_s, i\in I_s^p, j'\in J_i, n\in N \qquad (8)$$

$$\sum_{s\in S_i^{out}}\sum_{j'\in J_s} O_{s,j',i,j,n} = \rho_{s,i}^c B_{i,j,n}$$
$$\forall s\in S, i\in I, j\in J_i, n\in N \qquad (9)$$

$$\sum_{s\in S_i^{in}}\sum_{j'\in J_s} P_{s,j',i,j,n} = \rho_{s,i}^p B_{i,j,n}$$
$$\forall s\in S, i\in I, j\in J_i, n\in N \qquad (10)$$

where $\rho_{s,i}^p, P_{s,i,n} \ge 0$ and $\rho_{s,i}^c, O_{s,i,n} \le 0$. According to constraint in eq (5), the amount of material in eState s stored in unit j at event point n is equal to that at event point $n-1$ adjusted by any amount produced or consumed between the event point $n-1$ and n and the amount required by market at event point n within the time horizon. However, at event point 1, the material in eState s in unit j can only be consumed, so that constraints in eq (6) is formulated, where $STin_{s,j}$ is the initial amount of material at beginning of time horizon. Because the intermediates and products produced by eTasks starting at event point n can only be obtained at event point $n+1$, hence the constraints in eq (7) have to be comprised in the model. Constraints in eq (8) express that only if unit j is allocated to storing material in eState s at event point n, the material produced by eTask i in unit j' can be transfered to eState s in unit j. Constraints in eq (9) (eq (10)) state that the amount of materials consumed (produced) by eTask i has to be equal to that provided (consumed) by eStates.

Demand Constraints

$$\sum_{n\in N\cup\{\bar{n}+1\}}\sum_{j\in J_s} D_{s,j,n} \ge r_s \quad \forall s\in S \qquad (11)$$

These constraints represent that the amount of material produced within scheduling time horizon has to be greater than or equal to market demands.

Duration Constraints of Processing Tasks

$$Te_{j,n} \ge Ts_{j,n} + \sum_{i\in I_j} \alpha_{i,j} W_{i,j,n} + \sum_{i\in I_j} \beta_{i,j} B_{i,j,n}$$
$$\forall j\in J, n\in N \qquad (12)$$

Constraints in eq (12) express the duration constraints when eTask i is performed in unit j. At event point n, if no processing task is performed in unit j ($W_{i,j,n}=0, \forall i\in I$), then $Te_{j,n} \ge Ts_{j,n}$; if eTask i is performed in unit j ($W_{i,j,n}=1$), $Te_{j,n} \ge Ts_{j,n} + \alpha_{i,j} + \beta_{i,j} B_{i,j,n}$. As stated above, processing time is related to batch size. Assume that the mean value of processing time is $P_{i,j}$, the coefficient can be computed as follows (Ierapetritou and Floudas, 1998):

$$\alpha_{i,j} = \tfrac{2}{3} P_{i,j} \qquad \beta_{i,j} = \frac{\tfrac{2}{3} P_{i,j}}{V_{i,j}^{\max} - V_{i,j}^{\min}} \qquad (13)$$

ETasks in the Same Unit

$$Ts_{j,n+1} - Te_{j,n} \geq 0 \quad \forall j \in J, n \in N \setminus \{\overline{n}\} \tag{14}$$

These constraints express that the time of unit j starting an eTask at event point $n+1$ must be greater than or equal to the time of the same unit completing an eTask at event point n.

Different eTasks in Different Units

It can be found in Fig. 2 that at event point $n+1$, if eTask T4/2b consumes the material produced by T1/1a and T3/2a at event point n, the time of unit 2b starting T4 at event point $n+1$ should be greater than or equal to the time of unit 1a completing T1 and unit 2a completing T3. Such time constraints can be formulated as follows:

$$Tsi_{s,n+1} - Te_{j,n} \geq -H(1 - W_{i,j,n})$$
$$\forall s \in S, i \in I_s^p, j \in J_i, n \in N \setminus \{\overline{n}\} \tag{15}$$

$$Tsi_{s,n+1} \geq Tsi_{s,n} \quad \forall s \in S, n \in N \tag{16}$$

$$Ts_{j',n+1} - Tsi_{s,n+1} \geq -H(1 - W_{i',j',n+1})$$
$$\forall s \in S, i' \in I_s^c, j' \in J_{i'}, n \in N \setminus \{\overline{n}\} \tag{17}$$

where $Tsi_{s,n}$ denotes the minimum time when the eTasks in set I_s^c can be started at event point n. Constraints in eq (15) state that $Tsi_{s,n+1}$ should be greater than or equal to $Te_{j,n}$ if unit j is allocated to performing eTask i which produces the material in eState s at event point n ($W_{i,j,n} = 1$). Eq (16) is needed to impose the monotonicity in $Tsi_{s,n}$. Constraints in eq (17) specify that $Ts_{j',n+1}$ is greater than or equal to $Tsi_{i',n+1}$ if unit j' is assigned to performing eTask i' which consumes the material in eState s at event point $n+1$.

Constraints in eq (5)-(10) and (15)-(17) also guarantee the materials consumed by eTasks can't exceed the available amounts at any time. Eq (5)-(10) guarantee the quantitative relation of material in eState s between event point n and $n+1$, while Eq (15)-(17) impose the constraints that the starting time of eTasks in set I_s^c at event point $n+1$ is later than the end time of eTasks in set I_s^p at event point n.

Duration Constraints of Storage Tasks

$$Ts_{j,n} - Te_{j',n-1} \leq H(2 - Y_{s,j,n} - W_{i,j',n-1})$$
$$s \in S, j \in J_s, i \in I_s^p, j' \in J_i, n \in N \setminus \{1\} \tag{18}$$

$$Te_{j,n} - Ts_{j',n+1} \geq -H(2 - Y_{s,j,n} - W_{i,j',n+1})$$
$$s \in S, j \in J_s, i \in I_s^c, j' \in J_i, n \in N \setminus \{\overline{n}\} \tag{19}$$

Eq (18) and (19) are additional constraints corresponding to the starting and finishing timings of storage tasks. At event point n, if unit j is allocated to holding material in eState s produced by processing task $i \in I_s^p$ in unit j' at event point $n-1$, then $Ts_{j,n}$ should be smaller than or equal to $Te_{j',n-1}$. On the other hand, $Te_{j,n}$ should be greater than $Ts_{j',n+1}$ if unit j is assigned to holding material in eState s at event point n ($Y_{s,j,n} = 1$) and unit j' is allocated to performing processing task i at event point $n+1$ ($W_{i,j',n+1} = 1$).

Scheduling Time Horizon Constraints

$$Te_{j,\overline{n}} \leq H \quad \forall j \in J \tag{20}$$

The time horizon constraints represent that the finishing time of last event point in each unit is less than or equal to the scheduling time horizon.

Objective Function

$$\sum_{s \in S} \sum_{j \in J_s} \sum_{n \in N \cup \{\overline{n}+1\}} \Pr_s D_{s,j,n} \tag{21}$$

The objective shown in eq (21) is to maximize production in term of profit within scheduling time horizon.

Remark

The model proposed above can accommodates UIS, FIS and NIS storage policies. However, it can't accommodate ZW storage policy. For this purpose, eq (15) and (17) are reformulated as follows:

$$Tsi_{s,n+1} - Te_{j,n} \geq -H(1 - W_{i,j,n})$$
$$\forall s \in S, i \in I_s^p, j \in J_i, n \in N \setminus \{\overline{n}\}$$

$$Tsi_{s,n+1} - Te_{j,n} \leq H(1 - W_{i,j,n})$$
$$\forall s \in S, i \in I_s^p, j \in J_i, n \in N \setminus \{\overline{n}\}$$

$$Ts_{j',n+1} - Tsi_{s,n+1} \geq -H(1 - W_{i',j',n+1})$$
$$\forall s \in S, i' \in I_s^c, j' \in J_{i'}, n \in N \setminus \{\overline{n}\}$$

$$Ts_{j',n+1} - Tsi_{s,n+1} \leq H(1 - W_{i',j',n+1})$$
$$\forall s \in S, i' \in I_s^c, j' \in J_{i'}, n \in N \setminus \{\overline{n}\}$$

4. REDUCTION OF MODEL SIZE

The model proposed in Section 3 is simple. It comprises less binary variables, continuous variables and constraints than that of any other existing model based on STN, RTN and mSTN. However, the model size can be further reduced by using the following techniques.

In batch plants, some dedicated storage maybe exist. In such a case, it is unnecessary to treat the eState as a storage task. This can get reduction in the number of event points, and save the allocation and storage timing constraints involving the dedicated storage tank. Moreover, the storage constraints in eq (4) can be reformulated as follows:

$$ST_{s,j,n} \leq C_{s,j}^{\max} \quad \forall s \in S, j \in J_s, n \in N \tag{4'}$$

and the constrains on material balance in eq (5)-(10) can be simplified as follows:

$$ST_{s,j,n} = ST_{s,j,n-1} - D_{s,j,n} + \sum_{i \in I_s^p} \rho_{s,i}^p \sum_{j' \in J_i} B_{i,j',n-1}$$
$$+ \sum_{i \in I_s^c} \rho_{s,i}^c \sum_{j' \in J_i} B_{i,j',n}$$
$$\forall s \in S, j \in J_s, n \in N \setminus \{1\} \tag{5'}$$

$$ST_{s,j,1} = STin_{s,j} + \sum_{i \in I_s^c} \rho_{s,i}^c \sum_{j' \in J_i} B_{i,j',1}$$
$$\forall s \in S, j \in J_s \tag{6'}$$

$$ST_{s,j,\overline{n}+1} = ST_{s,j,\overline{n}} - D_{s,j,\overline{n}+1} + \sum_{i \in I_s^p} \rho_{s,i}^p \sum_{j' \in J_i} B_{i,j',\overline{n}}$$
$$\forall s \in S, j \in J_s \tag{7'}$$

Assume that the intermediates are zero at beginning of scheduling time horizon. On such condition, it can be found in Fig. 2 that T1/1a is probably performed whereas T3/2a and T4/2b are surely inactive because the materials are zero. Following the above approach, the minimum event point at which a processing task i may be performed can be determined. *Assume that intermediates remain in plant as few as possible at end of scheduling time horizon*. Based on such assumption, it can be found in Fig. 2 that at the last event point $\overline{n}$, T4/2b is probably performed whereas T1/1a are certainly inactive. Following the above approach, the maximum event point at which the given processing task may be performed can be determined. Once we have determine the minimum and maximum event points between which a processing task may be performed, we can also determine the minimum and maximum event points between which an eState may be active.

It is noted that using the approaches stated above has no effect on the optimality of model. Moreover, the short-term scheduling is driven by customer orders, so the two assumptions made above are reasonable in most situations.

5. NUMERIC RESULTS AND DISCUSSION

In this section, the above mathematic formulation is applied to two examples, and some numeric results and discussion are presented. LP Proc (SAS Institute INC., 1989), the MILP solver of SAS 8.0, is used to solve the following examples on a 733 MHz, 128 MB Pentrium Ⅲ PC. Moreover, the branch and bound technique is employed to solve all the scheduling problems in this paper.

Example 1

As stated in section 2, mSTN is an extended representation for batch plants. Hence, the mathematic formulation proposed in this paper can deal with the scheduling problems of batch plants represented by STN.

Num.	*eState*	Storage Capacity	Initial amount	Price
1	Feed A	Unlimited	Unlimited	0.0
2	Feed B	Unlimited	Unlimited	0.0
3	Feed C	Unlimited	Unlimited	0.0
4	Hot A	100	0.0	0.0
5	IntAB	200	0.0	0.0
6	IntBC	150	0.0	0.0
7	Impure E	200	0.0	0.0
8	Product 1	Unlimited	0.0	10.0
9	Product 2	Unlimited	0.0	10.0

Table 1 Data of eStates for Example 1

Num.	Unit	Capacity	Suitability	Mean Proc. Time
1	Heater	100	Heating	1.0
2	Reactor 1	50	Reaction 1,2,3	2.0, 2.0, 1.0
3	Reactor 2	80	Reaction 1,2,3	2.0, 2.0, 1.0
4	Still	200	Separation	Product 1 1
				Product 2 2

Table 2 Data for eTasks of example 1

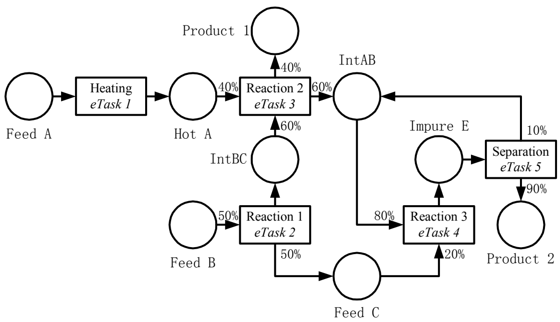

Fig. 3 STN representation for example 1

Consider the multipurpose batch plant shown in Fig. 3. This example is proposed by Kondili et al. (1993) and studied by Schilling and Pantelides (1996) and Ierapetritou and Floudas (1998a). Data for the example are shown in Table 1 and Table 2. Information including storage capacity as well as initial amount and price of each eState is provided in Table 1. The suitability, mean processing time and processing capacity of each unit are reported in Table 2.

The proposed formulation is used to solve this scheduling problem for four and five event points. The Model sizes and computational requirements are presented in Table 3. For four event points, the model involves 32 binary variables, 139 continuous variables, 186 constraints. Solving the scheduling problem only requires 0.23 CPU seconds. The optimal objective function corresponds to 1498.19 within the scheduling time horizon of 8h. The gantt chart for the optimal scheduling for four event points is shown in Fig. 4. The number in the chart denotes "eTask, unit, event point". For instance, the number "2,2,1" in the first chart of reactor 1 indicates that unit 2 is allocated to performing eTask 2 at event point 1.

Fig. 4 Gantt chart for example 1

Model	Event Points	0-1 Var.	Cont. Var.	Const.	Obj. Fun.	CPU Time
Proposed	4	32	139	186	1498.19	0.23s
Approach	4*	18*	111*	130*	1498.19*	0.18s*
	5	40	170	231	1498.19	5.94s
	5*	26*	142*	171*	1498.19*	2.48s*
Ierape_	5	40	260	374	1503.15	0.65s
tritou	6	48	310	465	1503.18	16.86s
Schilling	6	130	386	587	1488.05	

Table 3 Model sizes and computational requirements for example 1

In order to improve the computational performance of the model, the techniques proposed in Section 4 are used to solve the problem for four and five event points. The model sizes and computational requirements are also reported in Table 3. It can be found in Table 3 that the techniques can not only improve the computational performance, but also have no effect on the optimality of the problem. For example, when the techniques proposed in Section 4 are used for five event points, the model comprises 26 binary variables, 142 continuous variables and 171 constraints. In the same case, the model without the improvement techniques involves 40 binary variables, 170 continuous variables and 231 constraints. The computing time of the later model is 2.4 times as long as that of the former one, and the optimal solutions are identical.

Table 3 also reports the results of the proposed formulation compared with the results found in the literature for this example. It can be found that the proposed formulation needs less event points and its size is much smaller than the models proposed by Schilling and Pantelides (1996) and Ierapetritou and Floudas (1998a). However, the optimal solution is a little smaller than that of the model proposed by Ierapetritou and Floudas, and is much larger than that of the model proposed by Schilling and Pantelides. In fact, a scheduling model is a compromise between the utilization rate of resource and computational performance. The scheduling model with high computational performance and low utilization rate of resource would lose the favor of user while the model with low computational performance and high utilization rate of resource has no applicability when the scheduling problem of industrial size is considered. From the above two aspects, the proposed model is effective.

Example 2

This example involves the production of two products through five processing tasks. The mSTN representation for process recipe is shown in Fig. 2. Information about batch plant is shown in Table 4. It can be found that the plant involves 5 dedicated storage units (V1, V2, V4, V5 and V6), two dedicated processing unit (1b and 2b) and two multipurpose unit (1a and 2a) which can perform processing tasks and storage tasks.

For six event points, the proposed formulation involves 48 binary variables, 206 continuous variables and 333 constraints. Solving the problem requires 7.11s. The optimal objective function corresponds to 1850.51 within the time horizon 16h. The gantt chart for optimal scheduling for six event points is shown in Fig. 5. The characters in the chart denotes "Processing (Storage) Task/unit, event point"; gray rectangle represents processing task whereas white rectangle represents storage task. For example, "T1/1a,1" in the first gray rectangle of unit 1a indicates that processing unit 1a is allocated to performing eTask 1 at event point 1.

Unit	Suita-bility	Capa_city	Initial Amount	Proc. Time	Price
V1	S1/V1	+∞	+∞		
V2	S2/V2	+∞	+∞		
V4	S4/V4	55	0.0		
V5	S5/V5	30	0.0		10
V6	S6/V6	100	0.0		15
1a	T1/1a	20-40		2	
	T2/1a	25-40		2	
	S3/1a	45	0.0		
	S4/1a	45	0.0		
1b	T2/1b	15-45		2	
2a	T3/2a	10-50		4	
	S4/2a	50	0.0		
2b	T4/2b	20-55		4	

Table 4 Data for example 2

Fig. 5 Gantt chart for example 2

The improvement techniques introduced in Section 4 are also applied to this example. For six event points, the model involves 43 binary variables, 189 continuous variables and 301 constraints. Solving the problem only requires 5.67s. Apparently, in contrast with the model without improvement techniques, the improved model reveals better computational performance.

6. CONCLUSION

A new MILP formulation for short-term scheduling of multipurpose batch plant has been presented. It is based on continuous-time domain representation and more general process representation, namely mSTN. The model gives more detailed optimal scheduling without ambiguities in contrast with the model based on STN or RTN. When the model is built, assignment of units and eTasks to event points is represented by one set of binary variables. The size of the model is smaller than the existing model based on STN, RTN and mSTN. The model can accommodate multiplicate storage modes and all storage policies.

In order to reduce the size of the proposed model, some techniques are introduced. These techniques can not only save the number of binary variables, continuous variables and constraints, but also cut down the computational efforts. Furthermore, optimal solution can be derived from the model embedding these techniques.

Two examples have been tackled successfully by solving MILP models in a reasonable short computing time. The computation shows that the model reveals better computational performance than the existing models based on STN, RTN and mSTN.

REFERENCES

Barbosa-Povoa, A.P. and S. Macchietto (1994). Detailed design of multipurpose plants. *Computers & Chemical Engineering*, Vol. 18, 1013-1042

Crooks, C. (1992), Synthesis of operating procedures for chemical plants, *Ph.D Thesis,* University of London.

Ierapetritou, M.G. and C.A. Floudas (1998a). Effective continuous-time for short-term scheduling. 1., *Ind. Eng. Chem. Res.*, 37, 4341-4359.

Ierapetritou, M.G. and C.A. Floudas (1998b). Effective continuous-time for short-term scheduling. 2. *Ind. Eng. Chem. Res.*, Vol.37, 4360-4374

Ierapetritou, M.G. and C.A. Floudas (1999). Effective continuous-time for short-term scheduling. 3. Multiple intermediate due dates, Ind. Eng. Chem. Res., Vol.37, 3446-3461.

Kondili E., C.C. Pantelides and R.W.H. Sargent (1993). A general algorithm for short-term scheduling for batch operations-I, *Computers & Chemical Engineering*, 17, 211-227

Papageoriou, L.G., and C.C. Pantelides (1996). Optimal campaign planning/scheduling of multipurpose batch/semicontinuous pants. 1. *Ind. Eng. Chem. Res.* 35, 488-509.

Pekney, J.F. and G.V. Reklaitis (1998). Towards the convergences of theory and practice: a technology guide for scheduling/planning methodology. *AIChE Symposium Series 320*, 94, 75-90

Rippin, D.W.T. (1993). Batch process systems engineering: a retrospective and prospective review. *Computers and Chemical Engineering*, 17, s1-s13

Rodrigues, M.T., L.G. Latre, and L.C. Rodrigues (2000). Production planning using time windows for short-term multipurpose batch plants scheduling problems. *Ind. Eng. Chem. Res.*, 39, 3823-3834

SAS Institute Inc (1989). *SAS/OR User's Guide, Version 6, 1st ed*. SAS Institue Inc.: Cary, NC.

Schilling G. and C.C. Pantelides (1996). A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers & Chemical Engineering*, s1221-s1226

Shah, N. (1998). Single and multisite planning and scheduling: current status and future challenges. *AIChE Symposium Series 320,* 94, 75-90