Korean J. Chem. Eng., **18**(4), 422-427 (2001)

MILP Scheduling Model for Multipurpose Batch Processes Considering Various Intermediate Storage Policies

Ho-Kyung Lee, Sang Beom Kim*, Euy Soo Lee and In-Beum Lee†**

Dept. of Chemical Engineering, POSTECH, Pohang 790-784, Korea
*Samsung SDS Co., Ltd., World Tower, 7-25, Shinchun-Dong, Songpa-Gu, Seoul 138-240, Korea
**Dept. of Chemical Engineering, Dongguk University, Seoul 100-715, Korea

Abstract–MILP (Mixed Integer Linear Programming) scheduling models for non-sequential multipurpose batch processes are presented. Operation sequences of products have to be made in each unit differently by considering production route of each product under a given intermediate storage policy to reduce idle time of units and to raise the efficiency of the process. We represent the starting and finishing time of a task in each unit with two coordinates for a given storage policy. One is based on products, and the other is based on operation sequences. Then, using binary variables and logical constraints, we match the variables used in the two coordinates into one. We suggest MILP models considering sequence dependent setup times to guarantee the optimality of the solutions. Two examples are presented to show the effectiveness of the suggested models.

Key words: MILP Model, Scheduling, Non-Sequential Multipurpose Process, Intermediate Storage Policy

INTRODUCTION

Batch processes are economically desirable when a large number of products are made by using similar processing units, but productivity of their processes is heavily dependent on production scheduling [Ku et al., 1987]. During the last decade, scheduling problems for multiproduct batch processes have received considerable attention, and many papers were presented for completion time algorithm and mathematical models for optimal scheduling considering various intermediate storage policies such as UIS (Unlimited Intermediate Storage), NIS (No Intermediate Storage), FIS (Finite Intermediate Storage), ZW (Zero Wait), MIS (Mixed Intermediate Storage) policies which are used to raise efficiencies of processes [Ku and Karimi, 1988; Rajagopalan and Karimi, 1989; Jung et al., 1994; Kim et al., 1996; Ko et al., 2000].

However, in spite of their importance, optimal scheduling problems for multipurpose processes have received relatively little attention during the same period because it is very difficult to build a mathematical model and to solve it. Though a few papers for short-term scheduling of multipurpose processes or pipeless processes were presented [Pinto and Grossmann, 1995; Kim et al., 1997; Bok and Park, 1998; Moon and Hrymak, 1999], they did not consider various storage policies except UIS policy.

We suggest an optimal scheduling model for FIS policy [Kim et al., 2000] and extend it to various intermediate storage polices (UIS, ZW, NIS). The aim of this paper is to present mathematical models for optimal scheduling problems of multipurpose batch processes and to obtain optimal schedules that are able to minimize makespan under various storage policies. We treat non-sequential multipurpose processes as examples in which the production routes of some products may be opposite directions.

†To whom correspondence should be addressed.
E-mail: iblee@postech.ac.kr

This paper is organized as follows. In the second section, the problem that we intend to treat and our approach to the problem are given. In third section, the mathematical models for optimal scheduling of non-sequential multipurpose processes under various intermediate storage policies are proposed and then two examples are presented to show the effectiveness of the proposed models.

PROBLEM DEFINITION

We treat scheduling problems of non-sequential multipurpose batch processes considering various intermediate storage policies. Fig. 1 shows an example process that we will consider. Compared to multiproduct processes in which all products have the same production paths, in this process, each product has its own production path, and the direction of flow of product B is opposite to the direction of flow of product A. In Fig. 2(a), a Gantt chart is presented when this process is operated in the multiproduct-like manner, which means all products have the same processing sequence in all units. Because the difference of production routes between products is not considered in this operation, there is much idle time in each unit and process efficiency is very low.

In non-sequential multipurpose batch processes like this, if the process is operated in this manner, while the former units process the forward direction flows, the latter units have inevitable idle time due to opposite direction flows. Consequently, in order to reduce idle time of units and to raise process efficiency, we have to make

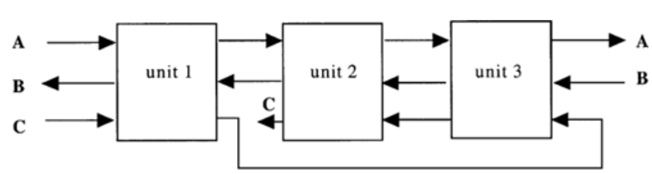

Fig. 1. Illustrative example.

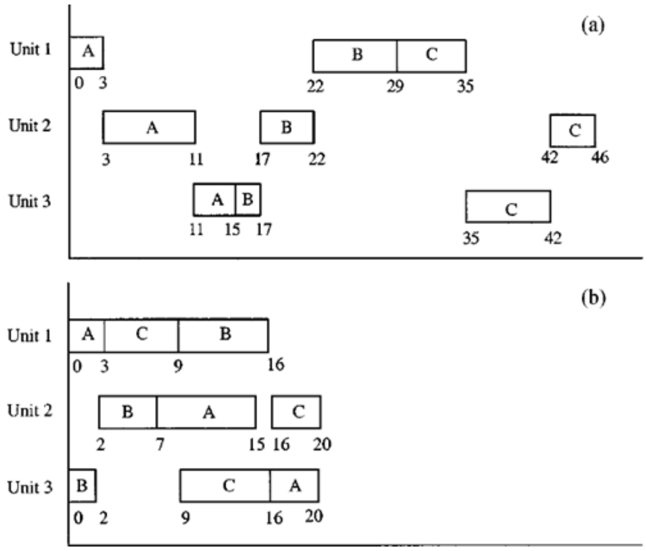

Fig. 2. (a) Schedule for multiproduct-like manner (the same production sequence (A→B→C in all units). (b) Optimal schedule for Fig. 1 under UIS policy.

operation sequences of products in each unit differently from one another like Fig. 2(b) by considering production route of each product and a given intermediate storage policy. Fig. 2(b) represents the optimal schedule for this process under UIS policy.

The problem we will treat in this paper is as follows.

1. Given

(1) Recipe of each product sequence of units that each product has to pass through.

(2) Processing time, transfer time, setup time for processing of each product in a unit. (All data are deterministic during scheduling)

(3) An intermediate storage policy among UIS, NIS, FIS and ZW.

2. Determine

The optimal processing sequence of products in each unit to minimize makespan.

The basic concept of our approach to this problem is as follows.

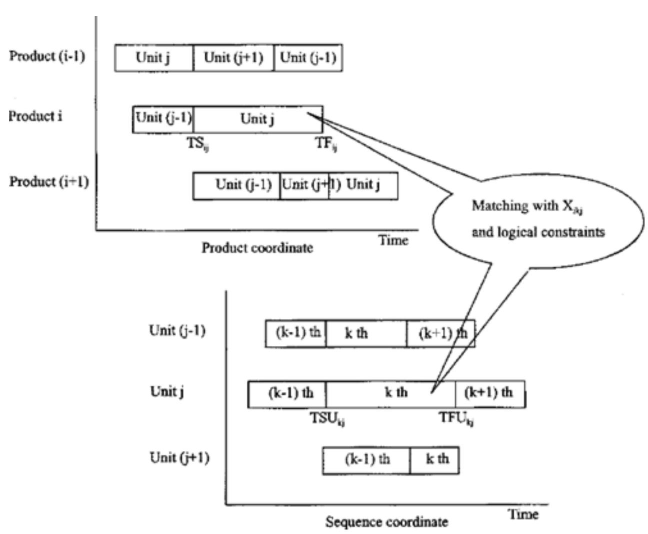

Fig. 3. Basic concept of approach used.

This approach is modified from the two coordinate representation method proposed by Pinto and Grossmann [1995]. First, we represented the starting and finishing time of a task in each unit with two coordinates for a given storage policy. One coordinate is based on products, and the other is based on operation sequences. Then, we matched the variables used in two coordinates into one with binary variables and logical constraints as shown in Fig. 3.

MILP MODELS

In this section, we propose MILP models for optimal scheduling problems of non-sequential multipurpose batch processes that have one unit per processing stage. Since we have suggested the MILP scheduling model for FIS policy [Kim et al., 2000], we present the MILP scheduling models for the UIS, ZW, and NIS policies in this section.

For the formulation of the model, binary variables X_{ikj} used in assignment constraints are defined as follows:

$$X_{ikj}=\begin{cases}1 \text{ if product is processed at kth sequence in unit j}\\ 0 \text{ otherwise}\end{cases}$$

The models are composed of the following constraints.

(a) Assignment constraints:

$$\sum_{k\in K_j} X_{ikj}=1 \qquad \forall j,\ i\in I_j \qquad (1)$$

$$\sum_{i\in I_j} X_{ikj}=1 \qquad \forall j,\ k\in K_j \qquad (2)$$

Eqs. (1) and (2) are assignment constraints. Eq. (1) represents that product i is assigned to only one position k in the processing sequences assigned to unit j. Here K_j means a set of sequences that are assigned to unit j and I_j means a set of products processed in unit j. Eq. (2) describes that a sequence in unit j is assigned to only one product.

In addition to Eqs. (1) and (2), we define new binary variables Y_{likj} used to keep this model linear in spite of considering sequence-dependent setup times between products in each unit [Byun et al., 1997].

$$Y_{likj}=\begin{cases}1 \text{ if product } l \text{ is followed by product i which is processed at kth sequence in unit j}\\ 0 \text{ otherwise}\end{cases}$$

(b) Constraints for linearity of models:

$$Y_{likj}\geq X_{l(k-1)j}+X_{ikj}-1 \qquad \forall j,\ (l,i)\in I_j,\ k\in K_j \qquad (3)$$

$$Y_{likj}\leq (X_{l(k-1)j}+X_{ikj})/2 \qquad \forall j,\ (l,i)\in I_j,\ k\in K_j \qquad (4)$$

Eqs. (3) and (4) represent the relation with binary variables X_{ikj} and Y_{likj}. By Eqs. (3) and (4), if product l is followed by product i which is processed in kth sequence in unit j, then the value of Y_{likj} is one. Otherwise, the value of Y_{likj} is zero.

Equation sets in (a) and (b) are common constraints used for all intermediate storage policies. To obtain the optimal schedules under various intermediate storage policies, additional constraints that express the characteristics of each storage policy are needed. They are formulated by two coordinate representation method.

1. MILP Model for UIS Policy

(c) Time constraints based on sequence coordinate:

$$TFU_{kj}=TSU_{kj}+\sum_{i\in I_j}a_{i(\_j)_i}X_{ikj}+\sum_{i\in I_j}(t_{ij}+a_{ij})X_{ikj} \qquad \forall j,\ k\in K_j \qquad (5)$$

$$TSU_{kj}\geq TFU_{(k-1)j}+\sum_{l\in I_j}\sum_{i\in I_j}S_{lij}Y_{likj} \qquad \forall j,\ k\in K_j \qquad (6)$$

Eq. (5) represents the relation of the starting and finishing time of kth task processed in unit j. Since there is enough storage capacity between units under UIS policy, the product which is finished processing in a unit can go to the next unit in the recipe or storage without holding in the present unit. So Eq. (5) is equality form. Eq. (6) means that kth task in unit j can start after completion of the previous task and setup needed between the tasks. Especially, compared with the MINLP models by Jung et al. [1994] and Kim et al. [1996], Eq. (6) is the linear form through eliminating quadratic terms for sequence-dependent setup times.

(d) Constraints for calculation of makespan:

$$LFT\geq TFU_{\bar{k}j} \qquad \forall j \qquad (7)$$

$$EST\leq TSU_{kj} \qquad \forall j \qquad (8)$$

The objective function we would like to optimize is the makespan, which is the latest finishing time (LFT) among all tasks minus the earliest starting time (EST) of all tasks. These constraints are commonly used for all other storage polices.

(e) Time constraints based on product coordinate:

$$TF_{ij}=TS_{ij}+a_{i(\_j)_i}+t_{ij}+a_{ij} \qquad \forall i,\ j\in J_i \qquad (9)$$

$$TS_{i(j\_)_i}\geq TF_{ij}-a_{ij} \qquad \forall i,\ j\in J_i \qquad (10)$$

Eq. (9) represents the relation of the starting and finishing time of product i processed in unit j. These constraints have the same meaning as Eq. (5) that is represented by sequence coordinate. Eq. (10) means that product i can start its processing in unit j after completion of processing in the previous unit in recipe.

(f) Time matching constraints:

$$-U(1-X_{ikj})\leq TS_{ij}-TSU_{kj}\leq U(1-X_{ikj}) \qquad \forall j,\ i\in I_j,\ k\in K_j \qquad (11)$$

$$-U(1-X_{ikj})\leq TF_{ij}-TFU_{kj}\leq U(1-X_{ikj}) \qquad \forall j,\ i\in I_j,\ k\in K_j \qquad (12)$$

Eqs. (11) and (12) are logical constraints used to match the time variables represented by two coordinates into one. When product i is processed at kth sequence in unit j, that is, the value of X_{ikj} is unity, these constraints can affect the model. Otherwise, they are relaxed due to large number U. For UIS policy, we may use only either Eq. (11) or Eq. (12) because relations between starting and finishing time of a task are equality constraints.

The problem to minimize makespan (MS), subject to above constraints, leads to the following MILP model for UIS policy:

Objective function Min MS=LFT−EST

Subject to (1)-(12).

2. MILP Model for ZW Policy

In ZW policy, the product must be transferred to the next unit in the recipe immediately after completion of task in a unit. So Eq. (10) has to be changed into equality form, which was an inequality form in UIS policy.

$$TS_{i(j\_)_i}=TF_{ij}-a_{ij} \qquad \forall i,\ j\in J_i \qquad (13)$$

Other constraints needed are the same as in the UIS case. MILP model for ZW policy is as follows:

Objective function Min MS=LFT−EST

Subject to (1)-(9), (11)-(13)

3. MILP Model for NIS Policy

In NIS policy, if the next unit in the recipe is busy for another processing, the product must be held in the present unit after completion until the next unit is available. In order to consider this case, we have to change Eqs. (5) and (9) from equality forms to inequality forms like Eqs. (14) and (15).

$$TFU_{kj}\geq TSU_{kj}\sum_{i\in I_j}a_{i(\_j)_i}X_{ikj}+\sum_{i\in I_j}(t_{ij}+a_{ij})X_{ikj} \qquad \forall j,\ k\in K_j \qquad (14)$$

$$TF_{ij}\geq TS_{ij}+a_{i(\_j)_i}+t_{ij}+a_{ij} \qquad \forall i,\ j\in J_i \qquad (15)$$

Moreover, since there is no storage between units, there is no difference between the starting time in unit j and the finishing time in the previous unit in recipe. So Eq. (10) is changed to equality form like Eq. (13).

Finally, in order to check the availability of the next unit in recipe, we should insert another constraint set (16).

$$TF_{ij}\geq TFU_{(k-1)(j\_)_i}+\sum_{l\in I_{(j\_)_i}}S_{li(j\_)_i}Y_{lik(j\_)_i}+a_{ij}-U(1-X_{ik(j\_)_i})$$

$$\forall i,\ j\in J_i,\ k\in K_{(j\_)_i} \qquad (16)$$

These equations mean that if we would like to finish processing for product i in unit j, the unit $(j\_)_i$, which is the next unit in the recipe, must be available. Eq. (16) also has logical constraints. So when the value of $X_{ik(j\_)_i}$ is unity, these constraints are activated, otherwise they are relaxed due to large number U. MILP model for NIS policy is as follows:

Objective function Min MS=LFT−EST

Subject to (1)-(4), (6)-(8), (11)-(16)

EXAMPLES

Two examples are treated to show the effectiveness of the models. The MILP solver we used to solve the problems is GAMS/OSL and H/W is IBM RS/6000 (model 350).

1. Example 1

Fig. 4 shows a non-sequential multipurpose process that produces four products with four units. To obtain the optimal schedule that

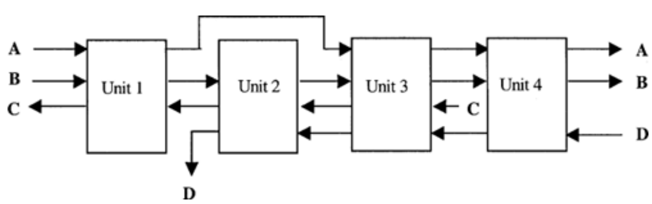

Fig. 4. Schematic diagram for example 1.

Table 1. Data for illustrative example

Product \ Unit	Unit 1	Unit 2	Unit 3
	Processing times		
A	3	8	4
B	7	5	2
C	6	4	7

Table 2. Data for example 1

Product \ Unit	Unit 1	Unit 2	Unit 3	Unit 4
	Processing times			
A	15		8	12
B	10	20	5	13
C	20	7	9	
D		7	17	5

Table 3. Optimal sequence in each unit for example 1

	Unit 1	Unit 2	Unit 3	Unit 4
UIS	B-A-C	C-B-D	C-D-A-B	D-A-B
ZW	B-C-A	C-B-D	C-D-B-A	D-B-A
NIS	A-B-C	C-B-D	C-A-D-B	D-A-B
FIS	B-A-C	B-C-D	C-D-B-A	D-B-A

Table 4. Statistical result for example 1

	0-1 variable	Continuous variable	Constraint	Makespan	CPU (sec)
UIS	52	58	196	59	0.43
ZW	52	58	196	71	0.43
NIS	52	58	317	63	0.85
FIS	68	73	361	60	1.24

Table 5. Production paths for example 2

	Flow characters	Production paths
P1	Forward	U1→U4→U5→U6
P2	Backward	U3→U2→U1
P3		U3→U5→U2

minimizes makespan under each storage policy, we applied the MILP models to this example. Table 2 shows the data of processing time and we assume that there is only one storage in unit 3 for FIS policy. Tables 3 and 4 show numerical results.

2. Example 2

This example treats a non-sequential multipurpose process that produces six products with six units. We consider the processing time, transfer time, and setup time in units (or in storage). Table 5 shows the production route of each product, and other time data are listed in Table 6.

For FIS policy, we assume that there is only one storage tank in unit 4. Tables 7 and 8 show numerical results. The computational results indicate that we can solve large-sized problems that are composed of about 200 binary variables and 1500 constraints by using

Table 6. Data for example 2

• Processing time

Product \ Unit	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
P1	6			10	17	4
P2	8	15	5			
P3		20	8		13	
P4			9	5	6	
P5	15		11	9		7
P6		13		5		10

• Transfer time

Product \ Unit	Unit 0	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
P1	1	1			2	2	1
P2	2	1	3	1			
P3	1		1	2		1	
P4	2			1	2	2	
P5	1	2		2	3		1
P6	1		1		1		1

• Setup time

Product \ Unit	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Storage	Unit 6
P1.P2	1						
P1.P3					1	2	
P1.P4				1	2	1	
P1.P5	2			2			1
P1.P6				1			2
P2.P1	1						
P2.P3		2	2				
P2.P4			1				
P2.P5	2		1				
P2.P6		3					
P3.P1					2	1	
P3.P2		1	3				
P3.P4			1		2	1	
P3.P5			3				
P3.P6		2					
P4.P1				1	1	2	
P4.P2			1				
P4.P3			1		2	1	
P4.P5			2	2			
P4.P6				3			
P5.P1	2			2			2
P5.P2	2		2				
P5.P3			2				
P5.P4			1	1			
P5.P6				2			3
P6.P1				1			1
P6.P2		1					
P6.P3		1					
P6.P4				3			
P6.P5				2			2

Table 7. Optimal sequence in each unit for example 2

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
UIS	P1-P2-P5	P2-P6-P3	P2-P3-P4-P5	P5-P6-P1-P4	P3-P1-P4	P6-P5-P1
ZW	P2-P1-P5	P2-P3-P6	P2-P3-P4-P5	P5-P4-P1-P6	P3-P4-P1	P5-P6-P1
NIS	P1-P2-P5	P2-P3-P6	P2-P3-P4-P5	P5-P1-P6-P4	P3-P1-P4	P5-P6-P1
FIS	P1-P2-P5	P2-P3-P6	P2-P3-P5-P4	P5-P1-P6-P4	P3-P1-P4	P5-P6-P1

Table 8. Statistical result for example 2

	0-1 Variable	Continuous variable	Constraint	Makespan	CPU (sec)
UIS	188	687	1553	70	2.19
ZW	188	687	1553	80	2.16
NIS	188	687	1585	78	2.15
FIS	240	726	1898	73	3.47

these models within reasonable CPU times.

CONCLUSION

We suggest MILP models for optimal scheduling of non-sequential multipurpose batch processes under various intermediate storage policies. To formulate this problem, we represented the starting and finishing time of a task in each unit with two coordinates. Then, using binary variables and logical constraints, we matched the variables used in two coordinates into one. The main contribution of our study is that we present mathematical models for multipurpose scheduling problems under various storage policies that have not been considered in previous studies. Additionally, compared with Jung et al. [1994] and Kim et al. [1996] using MINLP models for formulation of multiproduct scheduling problems, we suggest MILP models considering sequence-dependent set up times to guarantee the optimality of the solutions.

ACKNOWLEDGEMENT

This work was supported by KOSEF (Grant No. 1999-1-307-002-3).

NOMENCLATURE

Indices

l, i : products
j : units
$(\_j)_i$: a previous unit to unit j in the production path for product i
$(j\_)_i$: a next unit to unit j in the production path for product i
j' : storage included in unit j
k : sequences

Sets

I : set of products
J : set of units
K : set of sequences
I_j : set of products processed in unit j
J_i : set of units used to process product i
K_j : set of sequences assigned in unit j ($|I_j|=|K_j|$)

Parameters/Variables

a_{ij} : transfer time of product i from unit j to other units
EST : the earliest starting time of all tasks
LFT : the latest finishing time of all tasks
MS : makespan
S_{lij} : setup time when product l is followed by product i in unit j
SC_{lij} : setup time when product l is followed by product i in storage j
$SX_{ikj'}$: 1 if product i is stored at kth sequence in storage j' included in unit j, otherwise 0
SY_{likj} : 1 if product l is followed by product i which is stored at kth sequence in storage j' included in unit j, otherwise 0
t_{ij} : processing time of product i in unit j
TF_{ij} : finishing time of product i in unit j in product coordinate
TFU_{kj} : finishing time of kth sequence in unit j in unit coordinate
TS_{ij} : starting time of product i in unit j in product coordinate
TSU_{kj} : starting time of kth sequence in unit j in unit coordinate
X_{ijk} : 1 if product i is processed at kth sequence in unit j, otherwise 0
Y_{likj} : 1 if product l is followed by product i which is processed at kth sequence in unit j, otherwise 0

REFERENCES

Bok, J.-K. and Park, S., "Continuous-Time Modeling for Short-Term Scheduling of Multipurpose Pipeless Plants," *Ind. Eng. Chem. Res.*, **37**, 3652 (1998).
Byun, S. H., Lee, H. P., Kim, K. S. and Yoo, K. O., "Mixed Integer Linear Model for the Scheduling of Multiproduct Batch Processes with Transfer and Setup Times," *HWAHAK KONGHAK*, **35**, 877 (1997).
Jung, J. H., Lee, H., Yang, D. R. and Lee, I., "Completion Times and Optimal Scheduling for Serial Multi-Product Processes with Transfer and Setup Times in Zero Wait Policy," *Computers & Chemical Engineering*, **18**(6), 537 (1994).
Kim, H., Jung, J. H., Kim, M. and Lee, I., "A Hybrid System of Heuristics and Simulated Annealing for Large Size Scheduling Problems of Batch Processes with Pipeless Intermediate Storage," *Korean J. Chem. Eng.*, **14**, 225 (1997).
Kim, M. S., Jung, J. H. and Lee, I., "Optimal Scheduling of Multiproduct Batch Processes for Various Intermediate Storage Policies," *Ind. Eng. Chem. Res.*, **35**, 4058 (1996).
Kim, S. B., Lee, H., Lee, I., Lee, E. S. and Lee, B. S., "Scheduling of Non-Sequential Multipurpose Batch Processes Under Finite Intermediate Storage Policy," *Comp. and Chem. Engng*, **24**, 1603 (2000).
Ko, D., Na, S., Moon, I. and Lee, I., "Development of a Batch Manager for Dynamic Scheduling and Process Management in Multiproduct Batch Processes," *Korean J. Chem. Eng.*, **17**, 27 (2000).
Ku, H. M. and Karimi, I. A., "Scheduling in Serial Multiproduct Batch Processes with Finite Interstage Storage: A Mixed Integer Linear

Program Formulation," *Ind. Eng. Chem. Res.*, **27**, 1840 (1988).
Ku, H. M., Rajagopalan, D. and Karimi, I. A., "Scheduling in Batch Processes," *Chem. Eng. Prog.*, **Aug.**, 35 (1988).
Moon, S. and Hrymak, A. N., "Mixed-Integer Linear Programming Model for Short-Term Scheduling of a Special Class of Multipurpose Batch Plants," *Ind. Eng. Chem. Res.*, **38**, 2144 (1999).
Pinto, J. M. and Grossmann, I. E., "A Continuous Time Mixed Integer Linear Programming model for Short Term Scheduling of Multistage Batch Plants," *Ind. Eng. Chem. Res.*, **34**, 3037 (1995).
Rajagopalan, D. and Karimi, I. A., "Completion Times in Serial Mixed Storage Multi-Product Processes with Transfer and Setup Times," *Computers & Chemical Engineering*, **13**(1/2), 175 (1989).
Voudouris, V. T. and Grossmann, I. E., "MILP Model for Scheduling and Design of a Special Class of Multipurpose Batch Plants," *Computers & Chemical Engineering*, **20**(11), 1335 (1996).