Petri-Net Based Binary Integer Programs for Automatic Synthesis of Batch Operating Procedures

Jeng-Wen Lai, Chuei-Tin Chang,* and Shyh-Hong Hwang

Department of Chemical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, Republic of China

Traditionally, the task of conjecturing the operation steps in batch processes is carried out manually on an ad hoc basis. This approach is often time-consuming in industrial applications, and furthermore, the resulting recipes may be error-prone. The aim of this paper is thus to develop a systematic strategy to generate the optimal operation procedures with the Petri-net based binary integer programs (BIPs). Specifically, the system net consists of three types of standard subnets, i.e., the path modules, the equipment modules, and the process modules. The logic constraints in the corresponding BIP are formulated mainly for the purpose of describing the token movements in the system net. Additional constraints are also incorporated to enhance solution efficiency. The specific actions in the optimal operating procedure can be determined by solving this integer program. Two distinct classes of operation modes can be identified: (1) stage-based operating procedures and (2) time-based operating procedures. Several realistic examples are provided in this paper to demonstrate the feasibility of the proposed strategy.

1. Introduction

The batch operations considered in this work can be characterized more specifically as material transfer. Moving materials from one unit to another via interconnecting pipelines should be considered as a basic operation performed routinely in every batch chemical plant. Such operations are essential to almost all batch processes, e.g., mixing, filtration, distillation, extraction, reaction, energy transfer, and even cleaning.[1,2] Traditionally, the tasks of finding all possible material-transfer routes and then synthesizing the corresponding operating procedures are carried out manually on an ad hoc basis. For a complex industrial plant, the demand of these tasks for time and effort may be overwhelming and the resulting recipes are often error-prone. Thus, in order to relieve work load and also to enhance operation performance, it is highly desirable to develop a systematic strategy to conjecture the needed operation steps correctly and efficiently.

In a pioneering work, Rivas and Rudd[3] proposed a method for the synthesis of failure-safe procedures to help the operators make proper decisions during emergency situations. A valve operation sequence can be quickly determined to reach the given operation objective. O'Shima[4] handled this problem with a more efficient solution technique. The author developed the algorithms for finding the routes between the given starting and terminating points of a material stream and also for evaluating the flow state in each unit along the stream. The operating procedures were then synthesized on the basis of these algorithms. Foulkes et al.[5] represented the states of fragments in a plant structure with a series of condition lists. They utilized a combination of artificial intelligence techniques, pattern matching, and path search algorithms to identify all feasible routes for transferring a designated material from one storage tank to another in the plant. Uthgenannt[6] used digraph models to describe the network of interconnected process equipments. The material transfer routes and the required operating procedures can be obtained using a graph search method. A two-tier planning methodology was developed by Li et al.[7] The operating path was determined in the top tier with nonlinear programming models, while the primitive operation steps were synthesized in the bottom tier using a generic model-based reasoning method. Finally, the symbolic model verifier (SMV) was adopted by Kim and Moon[8] to generate safe operation steps for the multipurpose batch processes.

Although interesting results have been generated in the aforementioned studies, these methods are still not mature enough for the multitask applications in practice. One of the main reasons is due to the inherent deficiencies in their modeling tools. To this end, notice that a formal definition of the terminology, models, and functionality of industrial batch control systems has already been published in the ISA standard ISA-S88.01.[9] It was shown that a sequential function chart (SFC) is suitable for representing the hierarchical procedural model specified in this standard.[10] Notice also that an SFC is essentially derived from the basic concepts of Petri net, and a large collection of extensions are already available for enhancing the descriptive power of the latter model.[11] It is therefore logical to develop a Petri-net based methodology for generating the batch operating procedures automatically. A few related studies can be found in the recent literature. For example, a hierarchical control structure was proposed by Ferrarini and Piroddi,[12] with a supervisor module coordinating several slave controllers. The supervisor was represented with Petri net, whereas the slaves were represented by the sequential function chart. A design methodology for the logic control systems of batch processes was then developed accordingly. On the other hand, the approach taken in Chou and Chang[1] and also in Wang et al.[2] can be regarded as another example. A systematic procedure was suggested in these works to generate all feasible operating procedures for realizing the given material-transfer tasks according to the reachability tree of Petri-net model.

The Petri-net based approaches mentioned above suffer mainly two drawbacks in realistic applications, i.e., (1) the optimality of a selected feasible procedure cannot be guaranteed and (2) the time schedule for carrying out the material-transfer tasks must be given in advance. To solve these problems, the

* To whom correspondence should be addressed. Tel.: 886-6-275-7575 ext 62663. Fax: 886-6-234-4496. E-mail: ctchang@mail.ncku.edu.tw.

10.1021/ie0610400 CCC: $37.00
Published on Web 03/27/2007

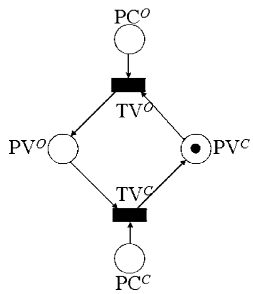

Figure 1. Standard valve model.

emphases of the present study are placed upon issues concerning the synthesis of optimal operating procedures and also their implementation schedule to achieve any collection of given material-transfer tasks on the basis of Petri-net models. For illustration purposes, the standard Petri nets are first developed in section 2 to represent the detailed operation actions, the material-transfer paths, and the corresponding variations in process conditions. A systematic procedure is also presented to construct the system model by assembling these components hierarchically. Binary integer programs (BIPs) can then be formulated accordingly to generate either stage-based or time-based operating procedures. The detailed model constraints and objective functions of these two types of BIP models are outlined in sections 3 and 4, respectively. Finally, the summaries of application studies concerning a large pipeline network and a beer filtration plant are provided in the last section.

2. Hierarchical Petri-Net Model

The mathematical representation of the *ordinary* Petri net is provided by Peterson.[13] A detailed review of the Petri-net elements and the transition enabling and firing rules can be found in the work of David and Alla.[11] On the basis of these fundamental developments, a systematic approach has been proposed by Wang and Chang[14,15] to construct Petri nets for modeling the batch operations. In particular, a system model can be assembled with a hierarchy of four different levels of components. In any existing batch process, the first-level component (which is usually a programmable logic controller or a human operator) is used to execute the operating steps specified in a recipe on the basis of a predetermined time schedule or a set of sensor measurements. Its actions alter the states of valves, pumps, and compressors in the second level. The states of these components in turn determine the process configuration and, consequently, the operation mode and equipment condition of each process unit in the third level. Finally, these process states are monitored via sensors in the last level, which may or may not be utilized as the basis for further controller actions.

For the purpose of recipe synthesis, it is obvious that the first-level components cannot be included in the Petri-net model since the operating procedures are unavailable. Moreover, to simplify BIP formulation, the sensor models are ignored in this work by assuming that the process conditions and their measurements are always identical. The remaining component Petri nets are briefly described as follows.

2.1. Representation of Equipment Operations. In order to generate the specific operation steps to realize the material-transfer tasks, the system Petri net must contain the second-level component models. The standard Petri-net representation of a valve can be found in Figure 1. Here, the places PV^O and PV^C denote two opposite valve positions respectively, i.e., open and close. The transitions $TV^O$and TV^Crepresent the valve-switching actions from PV^C to PV^O and vice versa. The input place PC^O of the transitions TV^Ocan be interpreted as the valve-opening command issued by a programmable logic controller or an operator, and the place PC^C can be considered as the demand for valve-closing action.

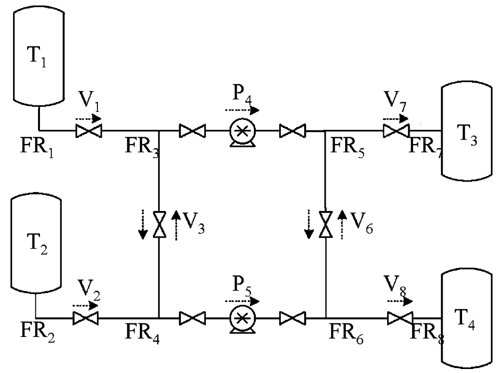

Figure 2. Typical pipeline network (example 1).

On the other hand, since the standard operating procedure (SOP) of a pump (or compressor) and its isolation valves can be regarded as a well-established industrial practice, e.g., see the work of Karassik and McGuire,[16] the detailed steps in SOP are not described in the corresponding equipment model for the sake of simplicity. Thus, the Petri net presented in Figure 1 is also used in the present study to model power-generating systems. In this case, the places PV^O and PV^C represent two opposite states, i.e., on and off, of the system, respectively. The transitions TV^O and TV^C can be regarded as a series of standard actions to turn on and off the pump/compressor system.

2.2. Representation of Material-Transfer Paths. The most critical issue in modeling a *process configuration* is concerned with division of the given system into distinct components. The concept of *fragments*[5] is adopted in this work for such a purpose. In particular, a fragment is defined as a collection of pipeline branches and/or process units separated from other fragments (or the environment) by valves, pumps, compressors, and other means of flow blockages in the piping and instrumentation diagram (P&ID). Let us consider Figure 2 as an example. Eight fragments can be identified according to this criterion, i.e., FR_1–FR_8. Notice that every pump and its isolation valves are viewed as one *lumped* power-generating system and this system is treated as a flow blockage if it is turned off. Notice also that, in many industrial plants, the pipeline networks contain dead branches. These branches are usually separated from the atmosphere by blanks, slip plates, and/or closed and locked valves. According to the definition given above, every dead branch and its connecting branches can still be viewed as a single fragment as long as no flow blockages can be found inside this fragment.

For illustration convenience, let us first examine the simplest fragment structure, i.e., a pipe branch isolated by an inlet valve and an outlet valve (see Figure 3a). In this case, the flow in either valve is allowed only in one direction. The corresponding Petri-net model is presented in Figure 3b. The place FR in this model is used to reflect the fragment state. More specifically, a token entering such a place denotes the condition that fluid is delivered to the corresponding fragment from an upstream source fragment. Notice that places PV_1^O and PV_2^O represent respectively the "open" positions of the upstream and downstream valves. Thus, the transitions CN_1 and CN_2 can be interpreted respectively as the events that establish the corresponding connections. On the other hand, if both valves in Figure 3a permit bidirectional material transfer, the fragment

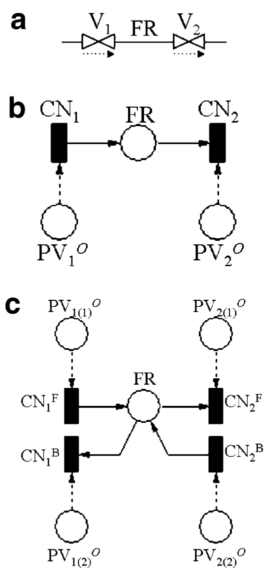

Figure 3. (a) Basic structure of a piping fragment with two single-direction valves. (b) Petri-net model of a basic fragment with two single-direction valves. (c) Petri-net model of a basic fragment with two bidirection valves.

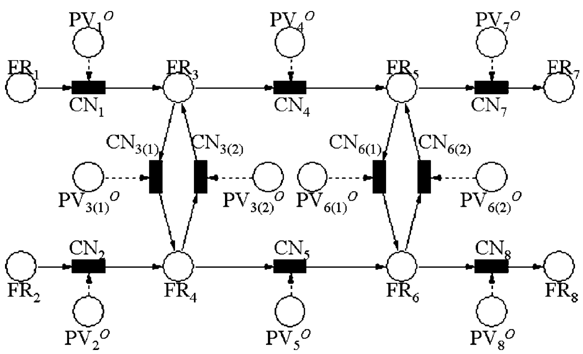

Figure 4. Path model of the example network in Figure 2.

model depicted in Figure 3b should be changed to the one shown in Figure 3c. Notice that each transition in the former Petri net is now replaced with two transitions to denote the material transfer actions to and from the fragment FR via the corresponding valve. Finally, it should be noted that all mass-transfer paths can be found in a Petri net assembled by connecting the fragment models according to the process configuration. This net is referred to as the *path model* in this study. For example, the path model presented in Figure 4 can be built on the basis of the P&ID in Figure 2.

2.3. Representation of the Process Units. As mentioned previously, a fragment may consist of not only pipelines but also *process units*. For example, each of the source or sink fragments in Figure 2 (i.e., FR_1, FR_2, FR_7, and FR_8) contains a piping branch as well as a storage tank. Any material-transfer operation inevitably changes the states of the process units located on the path between source and sink. If the operation goal is to control or manipulate the unit states, the corresponding places reflecting the fragment states in the path model should be replaced with the more detailed process models. A generalized version of this process model is presented in Figure 5. Here, the places $S_{i,1}$, $S_{i,2}$, ..., $S_{i,n}$ represent the states of the process unit located in fragment FR_i, and $W_{i,1}$, $W_{i,2}$, ..., $W_{i,n}$ can be viewed as the operation modes established by their connections with the adjacent fragments. Specific examples of this model will be presented later in the case studies.

2.4. System Representation. The Petri-net representation of the entire system is constructed in the present study by connecting the aforementioned equipment models and process models to the path model. This model-building approach may cause a problem in succinctly describing the operation steps of a bidirectional valve. Notice that the corresponding connection status between two adjacent fragments is in fact characterized with two distinct places in the path model (e.g., see $PV^{O}_{3(1)}$ and $PV^{O}_{3(2)}$ in Figure 4). Consequently, it is necessary to attach two separate valve models to these places respectively according to the proposed modeling method. As a result, the places representing the valve positions and also control commands are duplicated. This problem can be circumvented by imposing extra constraints in the integer program to reconcile the conflicting operation decisions which may be generated in the optimal solution. A detailed explanation of this technique will be given in the next section.

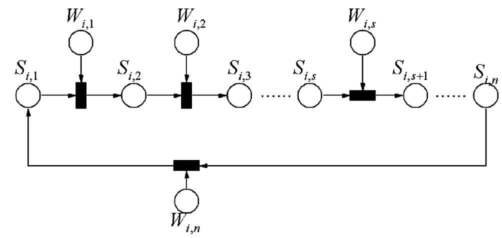

Figure 5. General petri-net representation of a process unit.

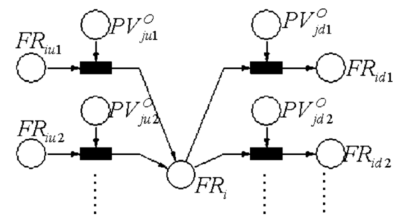

Figure 6. Petri-net representation of a fragment (FR_i) and its upstream and downstream connections.

3. Stage-Based Operating Procedures

For illustration convenience, a simplified version of the recipe-synthesis problem is considered here. Let us temporarily assume that the elapsed times of all material-transfer tasks are identical and thus can be ignored in the model formulation. The operation steps needed to perform any task are expected to be completed within a standard time period, which is referred to as a *stage* in this paper, and multiple tasks are allowed to be carried out in a single stage. In other words, only the implementation *order* of stages and the operation actions required in each stage are generated with the present approach.

3.1. Path Constraints. Logic constraints can be written to characterize the movements of tokens in the path model. In particular, two different types of binary variables, i.e., x_i and y^{O}_{j}, are adopted to represent the token numbers in places representing the fragment states (FR_i) and connection status (PV^{O}_{j}), respectively. To facilitate explanation of the constraint formulation, let us consider the generalized fragment model presented in Figure 6. The causal relations between the fragment state of FR_i and those of its downstream and upstream fragments can be translated into the following two inequality constraints respectively according to the formulation techniques developed by Raman and Grossmann,[17] i.e.,

$$(1 - x_i) + (1 - y^{O}_{jd}) + x_{id} \geq 1 \quad jd \in \mathbf{JD}_i \quad id \in \mathbf{ID}_{jd} \quad (1)$$

$$(1 - x_i) + (1 - y^{O}_{ju}) + x_{iu} \geq 1 \quad ju \in \mathbf{JU}_i \quad iu \in \mathbf{IU}_{ju} \quad (2)$$

where x_i, x_{id}, $x_{iu} \in \{0,1\}$; y^{O}_{jd}, $y^{O}_{ju} \in \{0, 1\}$; $\mathbf{JD}_i = \{jd_1, jd_2, ...\}$; $\mathbf{JU}_i = \{ju_1, ju_2, ...\}$; $\mathbf{ID}_{jd_k} = \{id_k\}$; $\mathbf{IU}_{ju_{k'}} = \{iu_{k'}\}$. In other words, the elements of set $\mathbf{JD}_i$ are used to distinguish the places representing the downstream connection states of fragment FR_i, i.e., $\mathrm{PV}^{\mathrm{O}}_{jd1}$, $\mathrm{PV}^{\mathrm{O}}_{jd2}$, ..., and the elements of set $\mathbf{JU}_i$ are for the places representing the upstream connection states of fragment FR_i, i.e., $\mathrm{PV}^{\mathrm{O}}_{ju1}$, $\mathrm{PV}^{\mathrm{O}}_{ju2}$, ...; $i\mathrm{d}_k$ and $i\mathrm{u}_{k'}$ denote the indices of the places representing the states of the kth downstream fragment and the k'th upstream fragment, respectively.

It is obvious that, other than the source and sink, there should be exactly one downstream connection and one upstream connection for any fragment FR_i on a material-transfer path. The corresponding logic constraints can be expressed as

$$(1 - x_i) + \sum_{j\mathrm{d}\in\mathbf{JD}_i} y^{\mathrm{O}}_{j\mathrm{d}} \geq 1 \tag{3}$$

$$(1 - x_i) + \sum_{j\in\mathbf{JU}_i} y^{\mathrm{O}}_{j\mathrm{u}} \geq 1 \tag{4}$$

$$\sum_{j\mathrm{d}\in\mathbf{JD}_i} y^{\mathrm{O}}_{j\mathrm{d}} \leq 1 \tag{5}$$

$$\sum_{j\in\mathbf{JU}_i} y^{\mathrm{O}}_{j\mathrm{u}} \leq 1 \tag{6}$$

Notice that constraints 1−6 must be imposed upon all fragments except that (1), (3), and (5) cannot be used to describe the flow connections of sinks and (2), (4), and (6) are not applicable in the case of sources.

3.2. Operation Constraints. The aforementioned constraints can be used only to characterize the material-transfer paths. For the purpose of generating actual operation steps, it is still necessary to incorporate additional constraints that represent the operation actions of valves, pumps, and compressors. These constraints can be derived on the basis of the Petri-net model given in Figure 1. Again, the token number in every place is represented with a distinct binary variable. Let us use the binary variables y^{O}_j and y^{C}_j to respectively denote the token numbers in the two places reflecting the states of equipment j, and use z^{O}_j and z^{C}_j to denote the token numbers in places representing the corresponding control commands. The token movements in the equipment model can therefore be described as follows:

$$\begin{aligned}(1 - y^{\mathrm{OI}}_j) + (1 - z^{\mathrm{C}}_j) + y^{\mathrm{C}}_j &\geq 1\\ (1 - y^{\mathrm{OI}}_j) + z^{\mathrm{C}}_j + y^{\mathrm{O}}_j &\geq 1\\ (1 - y^{\mathrm{CI}}_j) + (1 - z^{\mathrm{O}}_j) + y^{\mathrm{O}}_j &\geq 1\\ (1 - y^{\mathrm{CI}}_j) + z^{\mathrm{O}}_j + y^{\mathrm{C}}_j &\geq 1\end{aligned} \tag{7}$$

In the above constraints, $y^{\mathrm{OI}}_j$and y^{CI}_j represent respectively the initial values of y^{O}_j and y^{C}_j. The first constraint is equivalent to the logic statement that, if the initial equipment state is open ($y^{\mathrm{OI}}_j = 1$) and a "close" command is issued ($z^{\mathrm{C}}_j = 1$), then the resulting state should be closed ($y^{\mathrm{C}}_j = 1$). On the other hand, if the close command is *not* given ($z^{\mathrm{C}}_j = 0$) under the same initial condition ($y^{\mathrm{OI}}_j = 1$), the equipment should remain at its open state ($y^{\mathrm{O}}_j = 1$). Notice that the third and fourth constraints can be interpreted in a similar way and thus are not repeated for the sake of brevity. It is assumed in the proposed model that, other than the source valves and the pumps, the initial states of all other valves should remain unchanged from those in the previous stage. On the other hand, it is also reasonable to institute the routine practices of closing all opened *source* valves and switching off all running pumps at the end of each operation stage. In other words, the corresponding binary variables y^{OI}_j values should be set to 0 and y^{CI}_j values should be 1 in all operation stages.

In addition to the constraints given above, it is necessary to impose the following auxiliary constraints to enhance search efficiency:

(1) It is obvious that the state of any equipment must be unique, i.e.,

$$y^{\mathrm{O}}_j + y^{\mathrm{C}}_j = 1 \tag{8}$$

(2) If an equipment is at its open (or closed) state initially, then it is meaningless to execute the operation step to open (or close) the same equipment. The following constraints are adopted to prevent such possibilities

$$\begin{aligned}y^{\mathrm{OI}}_j + z^{\mathrm{O}}_j &\leq 1\\ y^{\mathrm{CI}}_j + z^{\mathrm{C}}_j &\leq 1\end{aligned} \tag{9}$$

(3) In order to ensure practical applicability, it is assumed that every piece of level-two equipment (except the source valves and power-generating devices) can be operated at most once. Thus, the corresponding constraints can be expressed as

$$z^{\mathrm{O}}_j + z^{\mathrm{C}}_j \leq 1 \tag{10}$$

(4) As mentioned before, the actions to close source valves and to switch off running pumps are assumed to be the routine steps performed at the end of each operation stage. The implied restrictions of this assumption can be written as

$$(1 - y^{\mathrm{O}}_j) + z^{\mathrm{C}}_j = 1 \quad j \in \mathbf{JP} \cup \mathbf{JS} \tag{11}$$

where **JP** and **JS** denote respectively the sets of all power-generating units and source valves.

On the other hand, notice that the equipment model of a bidirectional valve is built with two standard equipment models. Extra constraints are thus needed to reconcile the conflicting control commands resulting from this modeling practice. In particular, the two corresponding fictitious valves cannot be both open, i.e.,

$$y^{\mathrm{O}}_{j(1)} + y^{\mathrm{O}}_{j(2)} \leq 1 \tag{12}$$

Thus, the control commands to open or close these two fictitious valves should not be issued at the same time, i.e.,

$$\begin{aligned}z^{\mathrm{O}}_{j(1)} + z^{\mathrm{O}}_{j(2)} &\leq 1\\ z^{\mathrm{C}}_{j(1)} + z^{\mathrm{C}}_{j(2)} &\leq 1\end{aligned} \tag{13}$$

Furthermore, notice that all possible states of a bidirectional valve can be classified according to the states of two fictitious valves, i.e.

- State 1 $y^{\mathrm{O}}_{j(1)} = 0$, $y^{\mathrm{C}}_{j(1)} = 0$, $y^{\mathrm{O}}_{j(2)} = 0$, $y^{\mathrm{C}}_{j(2)} = 0$;
- State 2 $y^{\mathrm{O}}_{j(1)} = 1$, $y^{\mathrm{C}}_{j(1)} = 0$, $y^{\mathrm{O}}_{j(2)} = 0$, $y^{\mathrm{C}}_{j(2)} = 0$;
- State 3 $y^{\mathrm{O}}_{j(1)} = 0$, $y^{\mathrm{C}}_{j(1)} = 0$, $y^{\mathrm{O}}_{j(2)} = 1$, $y^{\mathrm{C}}_{j(2)} = 0$.

State 1 is associated with the closed position of a bidirectional valve, while states 2 and 3 both correspond to the open position. The flows corresponding to the latter two states are opposite in direction and marked as (1) and (2), respectively. Due to the fact that a bidirectional valve can be considered to be closed as

Table 1. Correspondence between the Fictitious and Actual Commands of a Bidirectional Valve

$z^{O}_{j(1)}$	$z^{C}_{j(1)}$	$z^{O}_{j(2)}$	$z^{C}_{j(2)}$	u^{O}_{j}	u^{C}_{j}
1	0	0	0	1	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	0	1
1	0	0	1	0	0
0	1	1	0	0	0
0	0	0	0	0	0

long as the flow in either direction is blocked, one can conclude that

$$y^{C}_{j(1)} = y^{C}_{j(2)} = y^{C}_{j} \tag{14}$$

Thus, eq 8 can be rewritten as

$$y^{O}_{j(1)} + y^{O}_{j(2)} + y^{C}_{j} = 1 \tag{15}$$

The transition from one state to another can be realized by manipulating the fictitious valves. It is clear that the maximum number of such transitions is six. To resolve the conflicting control commands required in these processes, let us use the binary variables u^{O}_{j} and u^{C}_{j} to represent respectively the *actual* control signals for opening and closing valve j. The correspondence between the fictitious and actual commands of a bidirectional valve is summarized in Table 1. Notice that, due to the need to satisfy the constraints in (13), not all combinations are included in this table. The fictitious commands listed in the first two rows of Table 1 are adopted to change from state 1 to state 2 and vice versa, while those in rows 3 and 4 can be used to activate the forward and backward transitions between states 1 and 3, respectively. Since there is only one action required in each of the above four fictitious commands, the corresponding actual command should be the same. These logic relations can be expressed as

$$\begin{aligned}
(1 - z^{O}_{j(1)}) + z^{C}_{j(2)} + u^{O}_{j} &\geq 1 \\
(1 - z^{C}_{j(1)}) + z^{O}_{j(2)} + u^{C}_{j} &\geq 1 \\
(1 - z^{O}_{j(2)}) + z^{C}_{j(1)} + u^{O}_{j} &\geq 1 \\
(1 - z^{C}_{j(2)}) + z^{O}_{j(1)} + u^{C}_{j} &\geq 1
\end{aligned} \tag{16}$$

Notice that not all binary variables are included in these constraints. This is due to the fact that the values (0) of the missing variables can be directly inferred from constraints 10 and 13.

The fictitious commands in the fifth and sixth rows of Table 1 represent two separate sets of operation steps needed to change the two-valve system state from 3 to 2 and 2 to 3, respectively. However, if either set of operation steps are carried out in practice, the bidirectional valve is required to be first opened and then closed or vice versa. This implies that the actual valve position is unchanged and thus no real actions should be taken. Following are the inequality constraints representing the inference rules given in row 5:

$$\begin{aligned}
(1 - z^{O}_{j(1)}) + (1 - z^{C}_{j(2)}) + (1 - u^{O}_{j}) &\geq 1 \\
(1 - z^{O}_{j(1)}) + (1 - z^{C}_{j(2)}) + (1 - u^{C}_{j}) &\geq 1
\end{aligned} \tag{17}$$

The constraints used to describe the logic in row 6 can be written as

$$\begin{aligned}
(1 - z^{C}_{j(1)}) + (1 - z^{O}_{j(2)}) + (1 - u^{C}_{j}) &\geq 1 \\
(1 - z^{C}_{j(1)}) + (1 - z^{O}_{j(2)}) + (1 - u^{O}_{j}) &\geq 1
\end{aligned} \tag{18}$$

Notice that the last row in Table 1 is associated with the possibility that the valve remains in its original state. In this situation, there should not be any actual action either. The corresponding constraints are the following:

$$\begin{aligned}
z^{O}_{j(1)} + z^{O}_{j(2)} + z^{C}_{j(1)} + z^{C}_{j(2)} + (1 - u^{O}_{j}) &\geq 1 \\
z^{O}_{j(1)} + z^{O}_{j(2)} + z^{C}_{j(1)} + z^{C}_{j(2)} + (1 - u^{C}_{j}) &\geq 1
\end{aligned} \tag{19}$$

Finally, to facilitate consistent model formulation, the binary variables associated with the actual control commands of the single-directional valves and power-generating systems are also expressed with the same notations, i.e.,

$$\begin{aligned}
u^{O}_{j} &= z^{O}_{j} \\
u^{C}_{j} &= z^{C}_{j}
\end{aligned} \tag{20}$$

3.3. Goal Constraints. Besides the above-mentioned constraints, there are still needs to incorporate additional ones for the purpose of achieving the operation goals. The simplest goal is to perform a single material-transfer task between a pair of given source and sink fragments. For example, let us consider the system presented in Figure 2 and the task of transporting material from tank T_1 to tank T_4. In this case, the binary variables representing the fragment states of $FR_1$and FR_8 should both be 1, while those associated with the other source and sink fragments should be set to 0, i.e.,

$$\begin{aligned}
x_1 = x_8 &= 1 \\
x_2 = x_7 &= 0
\end{aligned} \tag{21}$$

Notice that, since the path constraints 3−6 have already been included in the integer program, one of the conditions given in the first part of eq 21 (i.e., $x_1 = 1$or $x_8 = 1$) can in fact be neglected.

If there is a need to perform multiple tasks in a *single* stage, then a second subscript r can be added to the variables in path constraints given in eqs 1−4 to distinguish the corresponding material-transfer routes. Specifically, x_i (the token number in FR_i) and y^{O}_{j} (the token number in PV^{O}_{j}) can be replaced by $x_{i,r}$ and $\varsigma^{O}_{j,r}$. Since there is a one-to-one correspondence between a path and its source (or sink), the latter is used to identify the former in the present study. In other words, the subscript r is both the label of a source fragment and that of the corresponding path. Let us again consider the system in Figure 2 and two material-transfer tasks: (a) $T_1 \rightarrow T_7$ and (b) $T_2 \rightarrow T_8$. The corresponding goal constraints are

$$\begin{aligned}
x_{1,b} = x_{2,a} &= 0 \\
x_{7,a} = x_{8,b} &= 1
\end{aligned} \tag{22}$$

Notice that it is not necessary to stipulate the states of the source fragments here.

Since a fragment cannot be shared by more than one path and a level-two component can cause the material flow along only one path, the following constraints must be valid

$$\sum_{r} x_{i,r} \leq 1 \tag{23}$$

To avoid creating too many new variables and new operation constraints, the variables $\varsigma^{O}_{j,r}$ can be related to the equipment states with the following equation

$$\sum_r \varsigma^{O}_{j,r} = y^{O}_j \tag{24}$$

As a result, there is no need to introduce the extra subscript r into the binary variables for characterizing the operation steps. More specifically, the variables y^{O}_j, y^{C}_j, z^{O}_j, z^{C}_j, u^{O}_j, and u^{C}_j in eqs 5−20 should remain unchanged in multitask applications.

In multitask *multistage* applications, a third subscript t must be added to all variables in the path and operation constraints. Let us consider the problem of synthesizing operation steps for transporting material via separate routes in the pipeline network in Figure 2 according to a *given* order, e.g., (1) $T_1 \rightarrow T_4$ and (2) $T_2 \rightarrow T_3$. The corresponding goal constraints can be written as

$$\begin{aligned} x_{1,b,t} &= x_{2,a,t} = 0 \\ x_{8,a,1} &= x_{7,b,2} = 1 \end{aligned} \tag{25}$$

Notice that it is also possible to determine the execution order of operation stages and the detailed steps in each stage simultaneously with a BIP model. For example, let us consider four material-transfer tasks in the pipeline network in Figure 2, i.e. $T_1 \rightarrow T_3$, $T_2 \rightarrow T_4$, $T_1 \rightarrow T_4$, and $T_2 \rightarrow T_3$. If the implementation order of these tasks is not specified a priori, one can formulate the goal constraints as follows

$$\begin{gathered} x_{1,b,t} = x_{2,a,t} = 0 \\ \sum_{t=1}^{H} x_{7,a,t} = \sum_{t=1}^{H} x_{8,a,t} = \sum_{t=1}^{H} x_{7,b,t} = \sum_{t=1}^{H} x_{8,b,t} = 1 \end{gathered} \tag{26}$$

where, H is a sufficiently large positive integer. Finally, notice that it may not be necessary to specify the source and sink of every material-transfer path in certain multistage and multitask operations. For example, if the operation goal is to clean the entire pipeline network by moving detergent through *every* fragment, then the following constraint should be imposed upon *all* fragments:

$$\sum_{t=1}^{H} x_{i,t} \geq 1 \tag{27}$$

Notice that, in this constraint, the subscript r of the binary variable is dropped. This is due to our assumption that all source tanks are filled with detergent and there is no need to stipulate a definite source fragment for every material-transfer route. Consequently, subscript r in the corresponding path and operation constraints must also be removed from the BIP model in the cleaning applications.

3.4. Objective Functions. An objective function is required in the formulation of any mathematical program. A reasonable choice may be

$$\min_{\mathbf{U}^{O},\mathbf{U}^{C}} \sum_t \sum_r \sum_i x_{i,r,t} \tag{28}$$

where

$$\mathbf{U}^{O} = \begin{bmatrix} u^{O}_{1,1} & u^{O}_{1,2} & \cdots & u^{O}_{1,t} & \cdots & u^{O}_{1,H} \\ u^{O}_{2,1} & u^{O}_{2,2} & \cdots & u^{O}_{2,t} & \cdots & u^{O}_{2,H} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ u^{O}_{j,1} & u^{O}_{j,2} & \cdots & u^{O}_{j,t} & \cdots & u^{O}_{j,H} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \end{bmatrix}$$

$$\mathbf{U}^{C} = \begin{bmatrix} u^{C}_{1,1} & u^{C}_{1,2} & \cdots & u^{C}_{1,t} & \cdots & u^{C}_{1,H} \\ u^{C}_{2,1} & u^{C}_{2,2} & \cdots & u^{C}_{2,t} & \cdots & u^{C}_{2,H} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ u^{C}_{j,1} & u^{C}_{j,2} & \cdots & u^{C}_{j,t} & \cdots & u^{C}_{j,H} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \end{bmatrix}$$

This operation objective is essentially to minimize the total path length, i.e., the total number of fragments embedded in all material-transfer routes. Another candidate chosen for the present study is

$$\min_{\mathbf{U}^{O},\mathbf{U}^{C}} \left[\sum_t \sum_j u^{C}_{j,t} + \sum_t \sum_j u^{C}_{j,t} \right] \tag{29}$$

This alternative objective is to minimize the total number of operation steps.

3.5. Example 1. Let us again consider the batch plant given in Figure 2 and the corresponding path model presented in Figure 4. By connecting the equipment models to places PV^{O}_1−PV^{O}_8, the Petri-net representation of the entire system can be constructed. The material-transfer path and operation constraints can be formulated on the basis of this Petri net. The optimal operating procedure can then be generated by solving a BIP model built by incorporating these and the goal constraints with a given objective function. On the basis of the assumption that the initial states of all valves and pumps in the present example are closed and off, respectively, the following results can be obtained with the CPLEX module in GAMS.

Case 1. If we simply want to accomplish a single material-transfer task from FR_1 to FR_8, the path and operation constraints in the BIP model can be formulated according to eqs 1−20, while the goal constraints are given in eq 21. The corresponding integer program was solved to minimize the path length and step number. In both cases, the same solutions have been generated. The optimal route was found to be

$$FR_1 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_6 \rightarrow FR_8$$

The detailed operation steps are listed in Table 2.

Case 2. Let us next consider the multitask multistage material-transfer operation described by the goal constraints in eq 25. To construct the BIP model for this case, it is necessary to first introduce an additional subscript r to variable x_i and replace y^{O}_j with $\varsigma^{O}_{j,r}$ in eqs 1−4. Another subscript t should then be added to the variables in these modified path constraints and also eqs 5−20, 23, and 24. Using either path length or step number as the objective function, the following optimal routes can be identified with the BIP model, i.e.,

Stage 1: $FR_1 \rightarrow FR_3 \rightarrow FR_4 \rightarrow FR_6 \rightarrow FR_8$

Stage 2: $FR_2 \rightarrow FR_4 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_7$

Two slightly different operating procedures were generated according to the aforementioned objective functions. The one with the fewest operation steps is presented in Table 3. Notice that, if the operation objective is to minimize path length, the

Table 2. Stage-Based Operating Procedure for Case 1 of Example 1

stage/step	operation actions
1/1	open valves V_1, V_6, and V_8 switch on pump P_4
1/2	switch off pump P_4 close valve V_1

Table 3. Stage-Based Operating Procedure Achieving the Fewest Steps for Case 2 of Example 1

stage/step	operation actions
1/1	open valves V_1, V_3, and V_8 switch on pump P_5
1/2	switch off pump P_5 close valve V_1
2/1	open valves V_2 and V_7 switch on pump P_4
2/2	switch off pump P_4 close valve V_2

Table 4. Stage-Based Operating Procedure Achieving the Fewest Steps for Case 3 of Example 1

stage/step	operation actions
1/1	open valves V_1 and V_7 switch on pump P_4
1/2	switch off pump P_4 close valve V_1
2/1	open valves V_2 and V_8 switch on pump P_5
2/2	switch off pump P_5 close valve V_2
3/1	open valves V_2 and V_3 switch on pump P_4
3/2	switch off pump P_4 close valve V_2
4/1	open valve V_1 switch on pump P_5
4/2	switch off pump P_5 close valve V_1

solution of the BIP model may contain extra steps which do not affect the outcome of the material-transfer operation.

Case 3. The requirements of material-transfer operation in this case can be represented by the goal constraints given in eq 26. The corresponding path and operation constraints are essentially the same as those used in Case 2. The value of H adopted in the BIP model is 4. If the objective function is step number, then the following routes can be identified from the optimal solution

$$\text{Stage 1:}\quad FR_1 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_7$$

$$\text{Stage 2:}\quad FR_2 \rightarrow FR_4 \rightarrow FR_6 \rightarrow FR_8$$

$$\text{Stage 3:}\quad FR_2 \rightarrow FR_4 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_7$$

$$\text{Stage 4:}\quad FR_1 \rightarrow FR_3 \rightarrow FR_4 \rightarrow FR_6 \rightarrow FR_8$$

The corresponding operating procedure is presented in Table 4. Notice that a total of 19 operation steps are taken in this procedure. If the path length is used as the objective function, then the four designated tasks are required to be carried out in a different order via the same set of material-transfer routes, i.e.,

$$\text{Stage 1:}\quad FR_2 \rightarrow FR_4 \rightarrow FR_6 \rightarrow FR_8$$

$$\text{Stage 2:}\quad FR_1 \rightarrow FR_3 \rightarrow FR_4 \rightarrow FR_6 \rightarrow FR_8$$

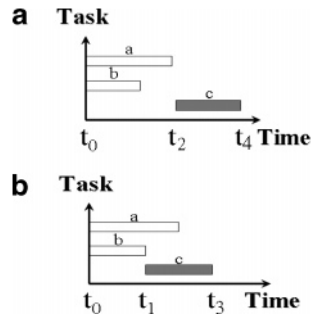

Figure 7. (a) Typical schedule achieved with stage-based operating procedure. (b) Typical schedule achieved with time-based operating procedure.

$$\text{Stage 3:}\quad FR_1 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_7$$

$$\text{Stage 4:}\quad FR_2 \rightarrow FR_4 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_7$$

The total number of steps in the corresponding procedure is increased to 21.

4. Time-Based Operating Procedures

In a stage-based operating procedure, every task is required to be performed within a designated stage and each stage ends only after all its tasks are completed. Since the fragments in a realistic pipeline network may be drastically different in shape, size, and length, the corresponding residence times of the transported material should not be the same and, therefore, the tasks in a stage may end at different times. A typical stage-based schedule for three material-transfer tasks (labeled as *a*, *b*, and *c*) can be found in Figure 7a. Due to the assumption that only paths *b* and *c* are partially overlapped in this system, these three tasks must be carried out in two consecutive stages. Since task *a* ends later than task *b* (i.e., $t_1 < t_2$), task *c* must start at time t_2 and end at t_4 in this schedule. It is obvious that the total operation time can be shortened to t_3 by starting task *c* at an earlier time t_1 according to the Gantt chart given in Figure 7b. In order to be able to produce operating procedures to realize this type of schedule, the description of time must be incorporated into the BIP model. Following is the proposed approach to formulate the model constraints.

4.1. Time-Tracking Mechanisms. The Petri nets described previously can still be used as a template for constructing the time-based BIP models. The only difference is that a delay must be assigned to each transition in this net to represent the residence time (or processing time) associated with its input place. The index *t* in the stage-based binary integer program is now treated as *actual time* in terms of the number of time units in the time-based model. Consequently, the variable $x_{i,r,t}$ (which represents the state of fragment *i* on path *r* in operation stage *t*) is replaced in the present case by a new variable $\sigma_{i,r,t}$ (which represents the state of fragment *i* on path *r* at actual time *t*). The former variable equals 1 in only one single stage, while the latter may assume the value of 1 at several consecutive instances as long as the material-transfer task on path *r* is in progress. In the proposed model, the values of fragment states are controlled with two extra binary variables according to the following equation:

$$\sigma_{i,r,t+1} = \sigma_{i,r,t} + \nu_{i,r,t+1} - \omega_{i,r,t+1} \tag{30}$$

where, $\nu_{i,r,t}$ is used to convert $\sigma_{i,r,t}$ from 0 to 1 and $\omega_{i,r,t}$ is used to convert vice versa. The path constraints in eqs 1−4 are now replaced with two identical sets of inequalities to regulate $\sigma_{i,r,t}$ and $\nu_{i,r,t}$, i.e.,

$$(1-\sigma_{i,r,t})+(1-\varsigma^{O}_{jd,r,t})+\sigma_{id,r,t}\geq 1 \quad jd\in \mathbf{JD}_i \quad id\in \mathbf{ID}_{jd} \tag{31}$$

$$(1-\sigma_{i,r,t})+(1-\varsigma^{O}_{ju,r,t})+\sigma_{iu,r,t}\geq 1 \quad ju\in \mathbf{JU}_i \quad iu\in \mathbf{IU}_{ju} \tag{32}$$

$$(1-\sigma_{i,r,t})+\sum_{jd\in\mathbf{JD}_i}\varsigma^{O}_{jd,r,t}\geq 1 \tag{33}$$

$$(1-\sigma_{i,r,t})+\sum_{j\in\mathbf{JU}_i}\varsigma^{O}_{ju,r,t}\geq 1 \tag{34}$$

$$(1-\nu_{i,r,t})+(1-\varsigma^{O}_{jd,r,t})+\nu_{id,r,t}\geq 1 \quad jd\in \mathbf{JD}_i \quad id\in \mathbf{ID}_{jd} \tag{35}$$

$$(1-\nu_{i,r,t})+(1-\varsigma^{O}_{ju,r,t})+\nu_{iu,r,t}\geq 1 \quad ju\in \mathbf{JU}_i \quad iu\in \mathbf{IU}_{ju} \tag{36}$$

$$(1-\nu_{i,r,t})+\sum_{jd\in\mathbf{JD}_i}\varsigma^{O}_{jd,r,t}\geq 1 \tag{37}$$

$$(1-\nu_{i,r,t})+\sum_{j\in\mathbf{JU}_i}\varsigma^{O}_{ju,r,t}\geq 1 \tag{38}$$

Again, it is necessary to use constraint 24 to combine all fictitious connection states into the corresponding actual equipment state. For the sake of completeness, this constraint is repeated below:

$$\sum_r \varsigma^{O}_{j,r,t}=y^{O}_{j,t} \tag{24$'$}$$

To ensure that exactly one downstream connection and one upstream connection for any fragment FR_i on a material-transfer path, constraints 5 and 6 should be imposed. They are also repeated as follows:

$$\sum_{jd\in\mathbf{JD}_i} y^{O}_{jd,t}\leq 1 \tag{5$'$}$$

$$\sum_{j\in\mathbf{JU}_i} y^{O}_{ju,t}\leq 1 \tag{6$'$}$$

Notice that a material-transfer task begins at the time when a feasible route can be chosen to satisfy all the path constraints mentioned above, i.e., eqs 5′, 6′, 24′, and 31−38. At this initial time (say t_0), all state variables σ_{i,r,t_0} and ν_{i,r,t_0} corresponding to the fragments on the selected path should be converted from 0 to 1 according to eq 30. If the task is still not completed at time t_0+1 (i.e., $\omega_{i,r,t_0+1}=0$), it can be further deduced from eq 30 that the values of ν_{i,r,t_0+1} should all be switched back to 0 while those of σ_{i,r,t_0+1} must remain unchanged at 1. On the basis of the same rationale, it can be concluded that the values of $\sigma_{i,r,t}$ and $\nu_{i,r,t}$ at the later time instances ($t=t_0+2$, t_0+3, ...) should be kept at 1 and 0, respectively, as long as the task is not terminated.

The control of $\omega_{i,r,t}$ values is achieved with still another set of binary variables $\tau_{i,r,t}$, which can be used to determine the total processing time of a material-transfer task via the selected path. In particular, the initial time of the task is recorded with the following equation:

$$\tau_{q,r,t}=\nu_{q,r,t} \quad q\in\mathbf{SC}_r \tag{39}$$

where $\mathbf{SC}_r$ is the set of source fragments of route r. In this study, it is assumed that the mean residence time needed for fluid particles to be transported through every fragment in the pipeline network can be determined in advance and the total processing time of a particular task can be estimated by summing these characteristic times associated with all fragments on the material-transfer path. Thus, the time-based token movements along the selected path can be described as

$$(1-\tau_{i,r,t})+(1-\varsigma^{O}_{jd,r,t})+\tau_{id,r,t+n_i}\geq 1 \quad jd\in\mathbf{JD}_i \quad id\in\mathbf{ID}_{jd} \tag{40}$$

where n_i denotes the mean residence time of fragment FR_i. Notice that the values of $\omega_{i,r,t}$ on route r should all be zero except when the task terminates at time t. For the possible sink fragments,

$$\omega_{q,r,t+n_q}=\tau_{q,r,t} \quad q\in\mathbf{SK}_r \tag{41}$$

where n_q denotes the mean residence time of sink fragment FR_q; $\mathbf{SK}_r$ is the set of all possible sink fragments of route r. For the other fragments on the material-transfer route,

$$\begin{gathered}(1-\sigma_{i,r,t})+(1-\sum_{q\in\mathbf{SK}_r}\omega_{q,r,t+1})+\omega_{i,r,t+1}\geq 1\\ (1-\omega_{i,r,t})+\sum_{q\in\mathbf{SK}_r}\omega_{q,r,t}\geq 1\\ (1-\omega_{i,r,t+1})+\sigma_{i,r,t}\geq 1\end{gathered} \tag{42}$$

These constraints are equivalent to the logic statement that if and only if fragment FR_i is used on path r at time t and the task is terminated at time $t+1$, then the corresponding $\omega_{i,r,t+1}$ must also be set to 1.

Finally, to ensure that none of the fragments can be shared by more than one route at the same time, eq 23 should be rewritten as

$$\sum_r \sigma_{i,r,t}\leq 1 \tag{43}$$

4.2. Time-Based BIP Model. The discussion presented in the above subsection is in a sense a description of the time-based path constraints. It can be observed that they are quite different from their stage-based counterparts. On the other hand, the stage-based operation constraints can be adopted in the time-based integer programs with only minor modifications. First of all, the subscript t should of course be added to eqs 7−20. In the time-based version of constraint 7, $y^{OI}_{j,t}$ and $y^{CI}_{j,t}$ represent respectively the initial values of $y^{O}_{j,t}$ and $y^{C}_{j,t}$ associated with *every* equipment unit in the system. They can be determined from the equipment states at the previous time instance, i.e.,

$$\begin{gathered}y^{OI}_{j,t}=y^{O}_{j,t-1}\\ y^{CI}_{j,t}=y^{C}_{j,t-1}\end{gathered} \tag{44}$$

Since each task may last for a period of time, the routine operation steps described in constraint 11 must be performed at the termination time, i.e.,

$$(1-\sum_r\omega_{i,r,t})+(1-y^{O}_{j,t})+z^{C}_{j,t}=1 \quad j\in\mathbf{JP}\cup\mathbf{JS} \quad i\in\mathbf{IP}_j\cup\mathbf{IS}_j \tag{45}$$

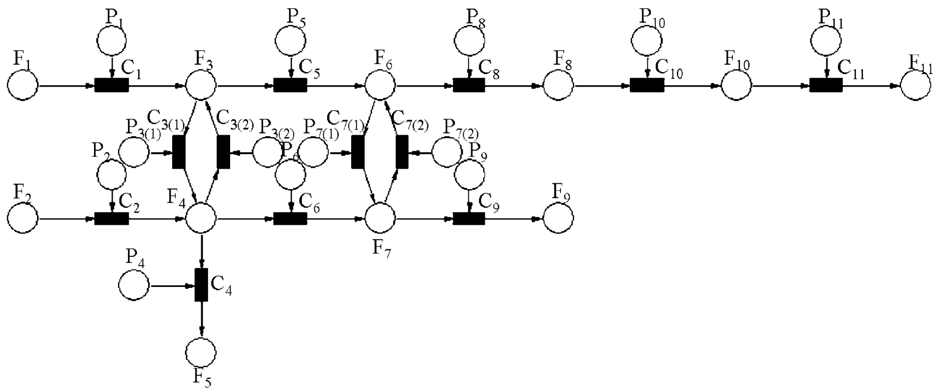

Figure 8. Path model of the pipeline network considered in example 2.

where the element of set $\mathbf{IP}_j$ is the upstream fragment of power-generating unit j while $\mathbf{IS}_j$ represents the set of upstream fragments of source valve j. Also, for the purpose of improving the solution efficiency, let us assume that all operation actions can be performed only at the initial times or termination times of the material-transfer tasks. The corresponding requirements can be written as

$$\sum_r(\nu_{i,r,t} + \omega_{i,r,t}) + (1 - u^{\mathrm{O}}_{j,t}) \geq 1 \quad j \in \mathbf{JP} \cup \mathbf{JV} \quad i \in \mathbf{IP}_j \cup \mathbf{IV}_j$$

$$\sum_r(\nu_{i,r,t} + \omega_{i,r,t}) + (1 - u^{\mathrm{C}}_{j,t}) \geq 1 \quad j \in \mathbf{JP} \cup \mathbf{JV} \quad i \in \mathbf{IP}_j \cup \mathbf{IV}_j \tag{46}$$

where $\mathbf{JV}$ denotes the set of all valves in the pipeline network; $\mathbf{IV}_j$ represents the set of upstream fragment of source valve j.

Since the techniques to formulate goal constraints are the same as those used for constructing the stage-based BIP models, they are not repeated here for the sake of brevity. The objective function presented in eq 29 can be used in the time-based integer program for synthesizing an operating procedure with the fewest steps. The objective function for achieving the shortest path length can be expressed as

$$\min_{\mathbf{U}^{\mathrm{O}},\mathbf{U}^{\mathrm{C}}} \sum_t\sum_r\sum_i \nu_{i,r,t} \tag{47}$$

Finally, it should be noted that a new objective function can also be formulated to generate a procedure for completing the given tasks within the shortest time period. For this purpose, let us introduce the final set of binary variables $\mathcal{f}_t$ to reflect if one or more material-transfer task is taking place at time t, i.e.,

$$(1 - \sigma_{i,r,t}) + \mathcal{f}_t \geq 1 \tag{48}$$

To avoid creating a schedule with one or more interruption periods, the following constraints must also be imposed:

$$(1 - \mathcal{f}_{t+1}) + \mathcal{f}_t \geq 1 \tag{49}$$

Thus, the objective function for yielding the minimum operation time can be written as

$$\min_{\mathbf{U}^{\mathrm{O}},\mathbf{U}^{\mathrm{C}}} \sum_{t=1}^{\mathrm{H}} \mathcal{f}_t \tag{50}$$

4.3. Example 2. Let us consider the path model shown in Figure 8. Notice that F_1-F_{11} are the places associated with fragment states and P_1-P_{11} represent the open states of the corresponding valves. Thus, a complete system model can be constructed simply by attaching the equipment models to P_1-P_{11}. It is assumed that the material flows in the corresponding pipeline network are driven by gravity, and therefore, no power-generating units are required in this system. Finally, it is also assumed that all valves are closed initially.

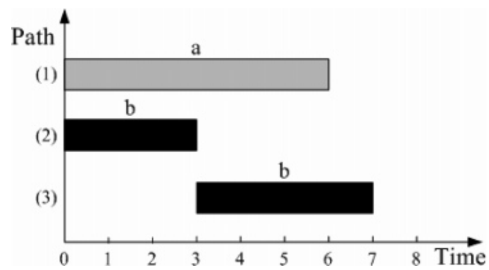

Figure 9. Optimal operation schedule in case 1 of example 2.

Table 5. Time-Based Operating Procedure Achieving the Minimum Time for Case 1 of Example 2

time	operation actions
0	open P_1, P_2, P_4, P_5, P_8, P_{10}, and P_{11}
3	close P_4; open P_6 and P_9
6	close P_1, (P_{10})
7	close P_2, (P_5)

Let us first label the routes originated from fragment F_1 as a and those from F_2 as b. The corresponding goal constraints should be

$$\sigma_{1,b,t} = \sigma_{2,a,t} = 0$$
$$\nu_{1,b,t} = \nu_{2,a,t} = 0 \tag{51}$$

Let us consider the cleaning operation by moving detergent through *every* fragment in the pipeline network. Thus, the additional goal constraints in this example should be

$$\sum_{t=1}^{\mathrm{H}}\sum_r \nu_{i,r,t} \geq 1 \tag{52}$$

Equation 50 is used as the objective function of the BIP model to minimize the total operation time. Two different sets of residence times have been adopted in this model. The resulting operating procedures are presented in the following.

Case 1. A residence time of 1 time unit is assigned to every fragment in this case. The value of time horizon H used in the

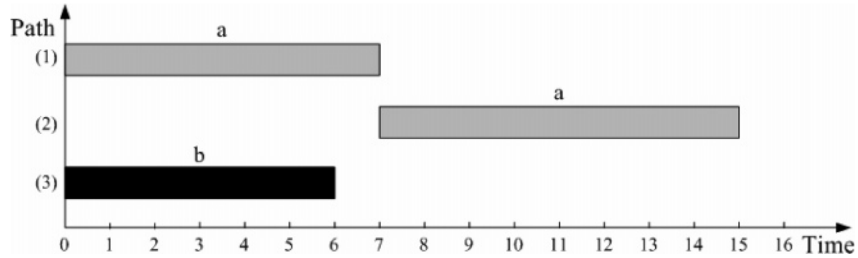

Figure 10. Optimal operation schedule in case 2 of example 2.

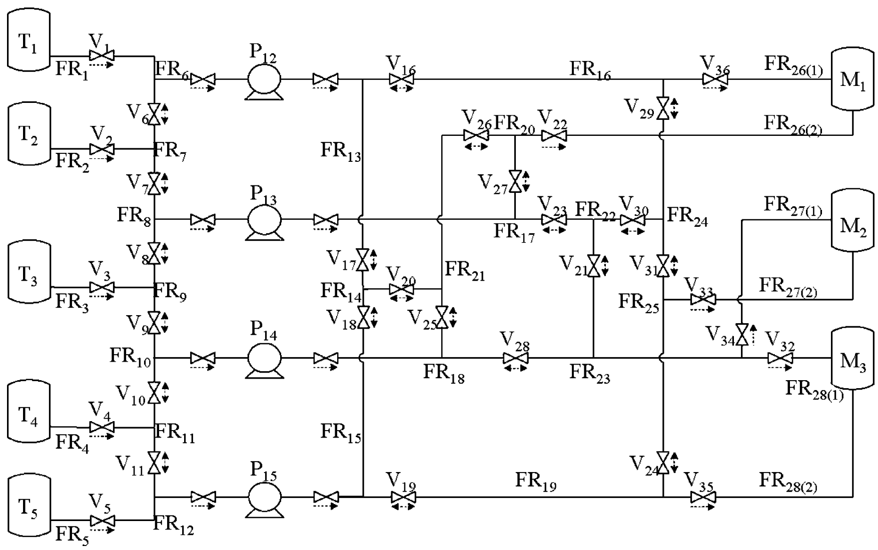

Figure 11. Complex pipeline network (application 1).

BIP model is 10. The material-transfer routes identified from the optimal solution are

$$\text{Path 1:}\quad F_1 \rightarrow F_3 \rightarrow F_6 \rightarrow F_8 \rightarrow F_{10} \rightarrow F_{11}$$

$$\text{Path 2:}\quad F_2 \rightarrow F_4 \rightarrow F_5$$

$$\text{Path 3:}\quad F_2 \rightarrow F_4 \rightarrow F_7 \rightarrow F_9$$

These tasks are required to be executed according to the Gantt chart presented in Figure 9. The detailed operation actions can be found in Table 5. Since the operation objective is to minimize the total cleaning time and the operation actions are considered to be instantaneous, the number of steps identified in the optimal solution may be larger than the minimum. Specifically, although the cleaning operation can be accomplished by following the procedure listed in Table 5, the same purpose can still be achieved without the actions to close P_{10} at time 6 and to close P_5 at time 7. These unnecessary steps are given in the parentheses in Table 5.

Case 2. Let us change the residence times of F_1, F_4, and F_9 to 2, 4, and 3, respectively. The residence times of the remaining fragments are still kept at 1 time unit. With a time horizon of 16, the material-transfer routes can be found to be

$$\text{Path 1:}\quad F_1 \rightarrow F_3 \rightarrow F_6 \rightarrow F_8 \rightarrow F_{10} \rightarrow F_{11}$$

$$\text{Path 2:}\quad F_1 \rightarrow F_3 \rightarrow F_6 \rightarrow F_7 \rightarrow F_9$$

$$\text{Path 3:}\quad F_2 \rightarrow F_4 \rightarrow F_5$$

The corresponding schedule and operating procedure are presented in Figure 10 and Table 6, respectively. Notice that, due to the changes in the residence times, path 2 is selected in the present case to replace path 3 in the previous case in order to minimize operation time. Notice also that the unnecessary steps are also shown in the parentheses in Table 6.

Table 6. Time-Based Operating Procedure Achieving the Minimum Time for Case 2 of Example 2

time	operation actions
0	open P_1, P_2, P_4, P_5, P_8, P_9, P_{10}, and P_{11}
6	close P_2 (P_4); open (P_6)
7	close (P_6), P_8; open P_7
15	close P_1

5. Applications

In order to demonstrate the effectiveness of the proposed approach in realistic applications, the results of two more complex case studies are presented here. The first is concerned with the stage-based operating procedure for cleaning a large pipeline network. The second application involves both stage-based and time-based procedures for operating a beer filtration plant.

5.1. Large Pipeline Network. The pipeline network described in Foulkes et al.[5] is adopted to demonstrate the capability of the proposed method for generating the stage-based procedure. The network contains 8 storage tanks, 36 valves, and 4 pumps (see Figure 11). A total of 5 source fragments, i.e., FR_1-FR_5, 20 internal fragments, i.e., FR_6-FR_{25}, and 6 sink fragments, i.e., $FR_{26(1)}-FR_{28(1)}$ and $FR_{26(2)}-FR_{28(2)}$, can be defined in this system. Since each sink tank in the present system has two inlet pipelines, these two inlets are thus treated as two distinct fragments in this example. By following the construction procedure described previously, the path model of the given system can be obtained (see Figure 12). Notice that the equipment models of the valves and pumps can be attached in a straightforward fashion to this path model. For the sake of

Table 7. Stage-Based Operating Procedure for Cleaning the Large Pipeline Network in Figure 11 with the Shortest Path Length

stage/step	operation actions
1/1	Open valves V_2, V_4, V_5, V_7, V_{10}, V_{16}, V_{18}, V_{20}, V_{22}, V_{23}, V_{26}, V_{28}, V_{30}, V_{31}, V_{32}, V_{33}, and V_{36}
	switch on pumps P_{13}, P_{14}, and P_{15}
1/2	switch off pumps P_{13}, P_{14}, and P_{15}
	close valves V_2, V_4, and V_5
2/1	open valves V_1, V_3, V_5, V_9, V_{19}, V_{34}, and V_{35}
	close valves V_{10}, V_{18}, V_{20}, V_{22}, V_{31}, V_{32}, and V_{33}
	switch on pumps P_{12}, P_{14}, and P_{15}
2/2	switch off pumps P_{12}, P_{14}, and P_{15}
	close valves V_1, V_3, and V_5

Table 8. Stage-Based Operating Procedure for Cleaning the Large Pipeline Network in Figure 11 with the Fewest Operation Steps

stage/step	operation actions
1/1	open valves V_2, V_3, V_5, V_7, V_9, V_{19}, V_{23}, V_{28}, V_{29}, V_{30}, V_{32}, V_{35}, and V_{36}
	switch on pumps P_{13}, P_{14}, and P_{15}
1/2	switch off pumps P_{13}, P_{14}, and P_{15}
	close valves V_2, V_3, and V_5
2/1	open valves V_1, V_3, V_4, V_{11}, V_{17}, V_{20}, V_{22}, V_{24}, V_{26}, V_{33}, and V_{34}
	close valves V_{32} and V_{33}
	switch on pumps P_{12}, P_{14}, and P_{15}
2/2	switch off pumps P_{12}, P_{14}, and P_{15}
	close valves V_1, V_3, and V_4

conciseness, the resulting Petri net is not shown here. Finally, it is assumed that all valves are closed and all pumps are switched off initially.

In this case study, the optimal cleaning procedures were synthesized with the proposed integer program on the basis of the objective functions specified in eqs 28 and 29. It is assumed that fresh detergent is allowed to be stored in all five source tanks and any sink tank can be utilized for collecting the waste material generated after cleaning. If the shortest path length is the operation objective, it was found that the material-transfer tasks are required to be carried out in two stages via six different routes, i.e.,

Stage 1:

$$(1)\ FR_2 \rightarrow FR_7 \rightarrow FR_8 \rightarrow FR_{17} \rightarrow FR_{22} \rightarrow FR_{24} \rightarrow FR_{25} \rightarrow FR_{27(2)}$$

$$(2)\ FR_4 \rightarrow FR_{11} \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{28(1)}$$

$$(3)\ FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{14} \rightarrow FR_{21} \rightarrow FR_{20} \rightarrow FR_{26(2)}$$

Stage 2:

$$(4)\ FR_1 \rightarrow FR_6 \rightarrow FR_{13} \rightarrow FR_{16} \rightarrow FR_{26(1)}$$

$$(5)\ FR_3 \rightarrow FR_9 \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{27(1)}$$

$$(6)\ FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{19} \rightarrow FR_{28(2)}$$

The resulting operation steps can be found in Table 7. It can be observed from the above data that the total number of fragments in all paths is 37 and the total number of operation actions is 49. Notice also that it is actually not necessary to close V_{20},V_{22}, V_{31}, and V_{33} in the second stage of the above procedure. On the other hand, if the step number is used as the objective function, the minimum number of operation actions can be reduced to 44 (see Table 8). However, notice that a different set of material-transfer routes are selected in this situation, i.e.,

Stage 1:

$$(1)\ FR_2 \rightarrow FR_7 \rightarrow FR_8 \rightarrow FR_{17} \rightarrow FR_{22} \rightarrow FR_{24} \rightarrow FR_{16} \rightarrow FR_{26(1)}$$

$$(2)\ FR_3 \rightarrow FR_9 \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{28(1)}$$

$$(3)\ FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{19} \rightarrow FR_{28(2)}$$

Stage 2:

$$(4)\ FR_1 \rightarrow FR_6 \rightarrow FR_{13} \rightarrow FR_{14} \rightarrow FR_{21} \rightarrow FR_{20} \rightarrow FR_{26(2)}$$

$$(5)\ FR_3 \rightarrow FR_9 \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{27(1)}$$

$$(6)\ FR_4 \rightarrow FR_{11} \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{19} \rightarrow FR_{25} \rightarrow FR_{27(2)}$$

5.2. Beer Filtration Plant. Let us consider the beer filtration plant presented in Figure 13a.[18] This system consists of two filters (MMS1 and MMS2), two buffer tanks (T_1 and T_2), a supply and collection system for the cleanser or purger, and the interconnecting pipeline network. Notice that the filtration process is operated with 16 double-disk piston valves (DV_1−DV_{16}). Each valve can be switched to two alternative positions, i.e., ON and OFF, according to Figure 14a−c to manipulate the connections between its inlets and outlets. Consequently, the process flow diagram in Figure 13a can be divided into the eight color-coded fragments presented in Figure 13b. Notice that the path model of this system can be constructed accordingly in a straightforward fashion. This Petri net is not presented here for the sake of brevity.

There are four different operations in this beer filtration process, i.e., filling, filtering, bottling, and cleaning. The purpose of the filling operation is to transport fresh beer from a source tank to the buffer tank T_1. In the filtering operation, the beer is transferred from tank T_1 to T_2 via filter MMS1 or MMS2. Clearly, the filtered beer in T_2 should then be moved to the bottling station in another material-transfer operation. The last operation in the plant is concerned with cleaning the fragments in which beer has been processed previously. It is assumed in this example that each fragment must be cleaned with purger after it has been used for a designated number of times.

5.2.1. Petri-Net Representations of Buffer Tanks and Filters. On the basis of the process model given in Figure 5, specific Petri nets have been constructed to represent the state transition processes of tanks and filters in the beer filtration plant. The Petri net in Figure 15a can be considered as a general model of these two units. Notice that there are eight places in this net. Four of them are used to represent *equipment states* and the rest are *work states*. The former four states are described as “clean”, “full with beer”, “foul”, and “full with purger”, while the latter states can be interpreted as transferring “beer into”, “beer out of”, “purger into”, and “purger out of” the given unit. If the purger does not stay in the process unit for a significant period of time during cleaning operation, then the three places on the right can be combined to form the Petri net given in Figure 15b. This is the process model for the buffer tanks T_1 and T_2. Notice that the combined place can be regarded as the work state “cleaning” in this case. If it can be assumed that beer goes through the filters almost immediately, the Petri-net model in Figure 15b can be further simplified to the net in Figure 15c. The process models in Figure 15b and c can be used to replace the places representing the corresponding fragments, i.e., F_3 (T_1), F_4 (T_2), F_1 (MMS1), and F_2 (MMS2), in the original

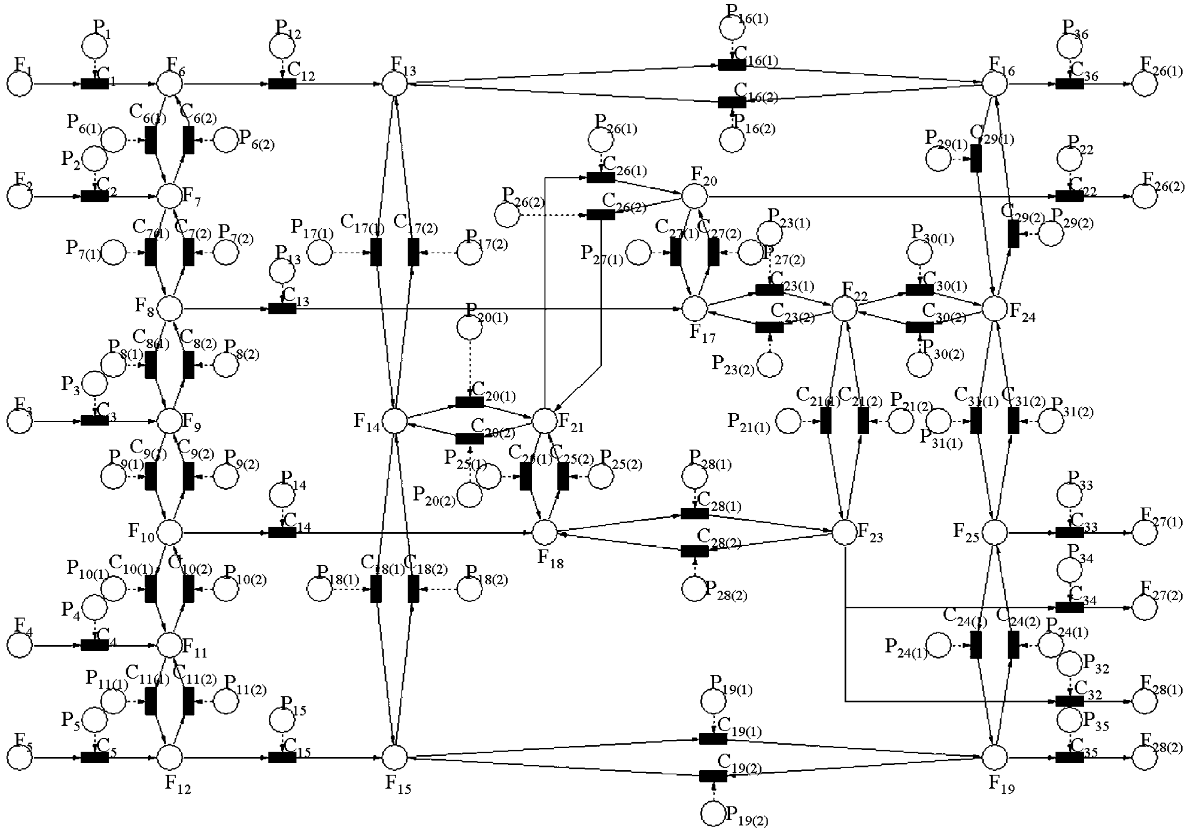

Figure 12. Path model of the complex network in Figure 11.

path model. This modified version is presented in Figure 16. In the four embedded process models, the places with label S are used to reflect the equipment states and those with label W represent the work states. The state of each of the remaining four fragments is modeled with a single place. In particular, F_5 and F_6 are the source and sink fragments of beer, while F_7 and F_8 are associated with the corresponding fragments of purger. From Figure 16, it can be observed that the sources F_5 and F_7 are connected to the places representing the work states of filters and tanks, and they are then linked to the sinks F_6 and F_8. Thus, it is clear that the material-transfer paths should contain places representing the work states. It should also be noted that the complete system model can be obtained by attaching the valve models to the Petri net in Figure 16.

Finally, it should be pointed out that the process models presented in Figure 15b and c imply that a buffer tank or filter can only be used once before cleaning for reuse. If a unit can be used more then one time without cleaning, it is necessary to incorporate additional places in these models to represent the extra unit states and work states. For example, the Petri nets in Figure 17a and b can be used to respectively model a tank and filter which allow two consecutive operations. These nets and those in Figure 15 can all be considered as special cases of the general model presented in Figure 5.

5.2.2. Stage-Based Operation. As mentioned before, the equipment states of the third-level components can be controlled by altering the process configuration with the second-level components. This hierarchical structure can be observed in the modified path model presented in Figure 16. Notice that the places representing fragment states (i.e., F_5−F_8) are connected to the places representing work states in the process models. The connection states (P_1−P_{16}) can be manipulated with the double-disk piston valves. Notice also that the modified path model in Figure 16 can be converted to the original version by lumping the places and transitions in every process model into a single place. In other words, the path and operation constraints presented in section 3 can still be used here, but additional constraints are needed to characterize the token movements within the process models. These constraints can be formulated according to Figure 5, i.e.,

$$
\begin{aligned}
(1-\lambda_{i,s,t}) + (1-\tilde{x}_{i,s,t}) + \lambda_{i,s',t+1} \geq 1 \\
(1-\lambda_{i,s,t}) + \tilde{x}_{i,s,t} + (1-\lambda_{i,s',t+1}) \geq 1
\end{aligned}
\qquad (53)
$$

where

$$
s' = \begin{cases} s+1 & \text{if } s = 1, 2, ..., n-1 \\ 1 & \text{if } s = n \end{cases}
$$

In the above constraints, $\lambda_{i,s,t}$ is the token number in place $S_{i,s}$ representing the sth equipment state of unit i at stage t and $\tilde{x}_{i,s,t}$ is the token number in place $W_{i,s}$ representing the sth work state of unit i at stage t. Notice that the constraints in eq 53 are in a sense very similar to those given in eq 7. The first constraint above is equivalent to the logic statement that, given the corresponding work state, the equipment state must be switched from one to another in sequence. This statement is essentially the same as that implied by the first and third constraints in eq 7. On the other hand, if the aforementioned work state is absent, then the corresponding state-transition event should not occur. The second constraint in eq 53 is imposed in the proposed process model to enforce this logic relation, and its counterparts are the second and fourth constraints in eq 7.

Additional constraints have also been added to enhance the solution efficiency. First of all, it is obvious that the equipment state of every unit at any given stage is unique. This condition

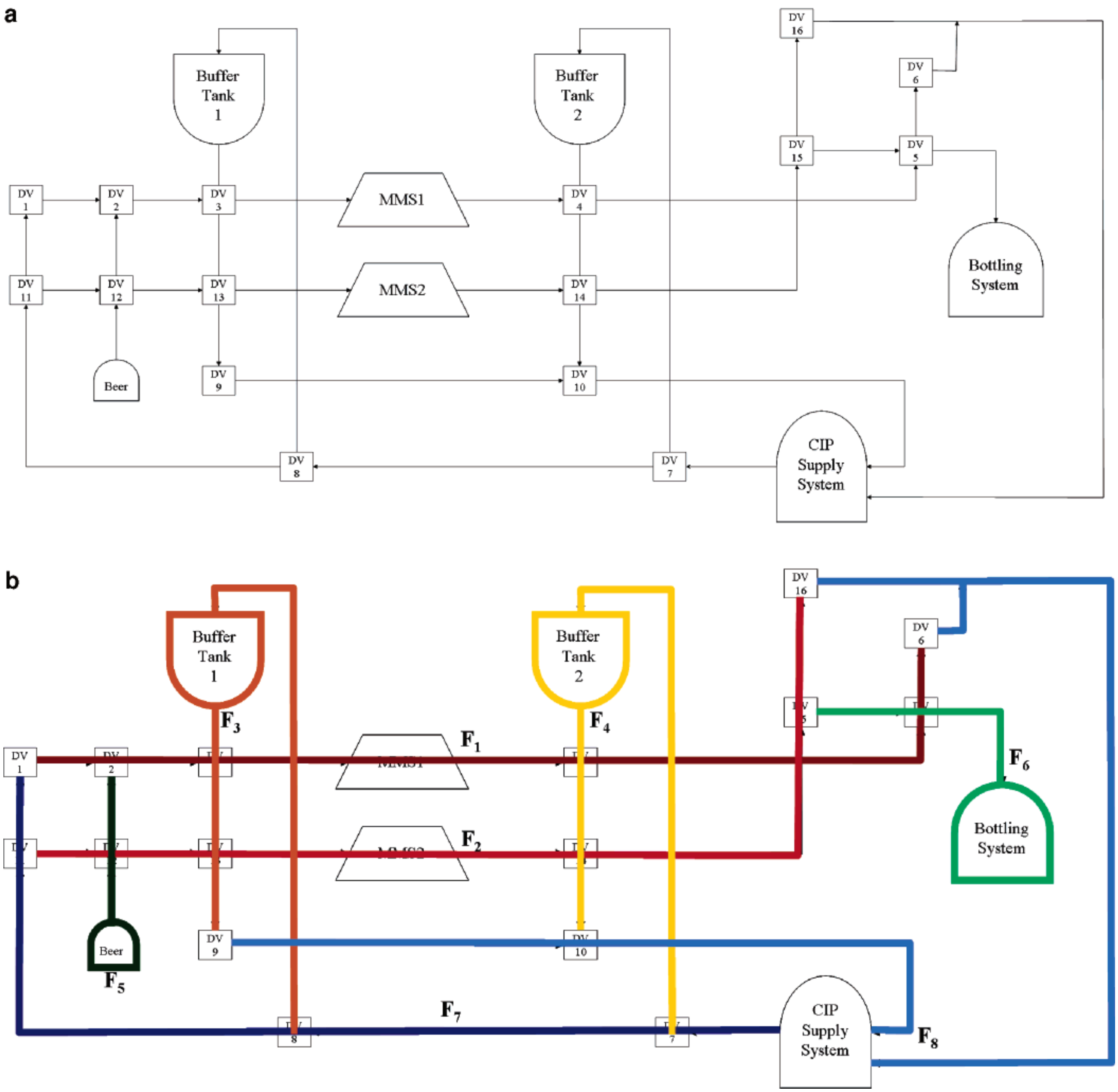

Figure 13. (a) Beer filtration plant (application 2). (b) Piping fragments in beer filtration plant.

can simply be described with the following constraint:

$$\sum_{s=1}^{n} \lambda_{i,s,t} = 1 \tag{54}$$

Notice that the rationale for imposing this constraint is the same as that for eq 8. Next, it is convenient to ensure that, if a particular unit state is not present, then the corresponding work state should be prohibited, i.e.,

$$\lambda_{i,s,t} + (1 - x_{i,s,t}) \geq 1 \tag{55}$$

Notice that the constraints in eq 9 can be interpreted from a similar standpoint.

It is conceivable that there are only two possibilities for a unit to reach a particular state at a given stage, i.e., (1) the unit was in a preceding state at the previous stage and switched to the present state afterward and (2) the unit was in the present state at the previous stage and remained unchanged. The corresponding constraint can be written as

$$(1 - \lambda_{i,s',t}) + \lambda_{i,s,t-1} + \lambda_{i,s',t-1} \geq 1 \tag{56}$$

Since this inequality is always valid in the case of two-state components, it is therefore not included in the operation constraints.

As mentioned previously, the places representing work states can be viewed as members of material-transfer paths. Thus, the goal constraints in this example are formulated on the basis of both fragment states and work states of various units. It should also be pointed out that, for generating stage-based operating procedures in the present application, there is no need to introduce the subscript r to distinguish the source fragments of material-transfer paths. This is due to the fact that constraint 54 prevents multiple operations to be performed in the same

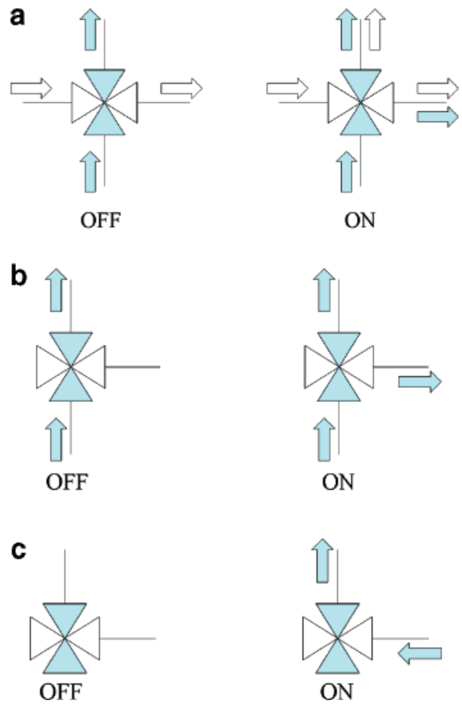

Figure 14. Double-disk piston valve. (a) Two inputs and two outputs. (b) One input and two outputs. (c) One input and one output.

units. Consequently, the goal constraints of the aforementioned four different operations can be summarized in the following:

• In the filling operation, it is required to transfer fresh beer from the source tank to buffer tank T_1. This requirement can be formulated as

$$x_{5,t} = \sum_{n_1=1}^{N_1} \tilde{x}_{3,b_i(n_1),t} \tag{57}$$

where N_1 is the number of times allowed for repeating the filling operation in T_1 without cleaning; $b_i(n_1)$ denotes the work state of transferring beer into the buffer tank for the n_1th time.

• In the filtering operation, the beer in tank T_1 must be sent to tank T_2 by way of a filter. The corresponding constraint is

$$\sum_{n_1=1}^{N_1} \tilde{x}_{3,b_o(n_1),t} = \sum_{n_2=1}^{N_2} \tilde{x}_{4,b_i(n_2),t} \tag{58}$$

where N_2 is the number of repeat operations allowed in T_2; $b_i(n_2)$ represents the work state of transferring beer into the buffer tank T_2 for the n_2th time; $b_o(n_1)$ is corresponding to the work state of withdrawing beer from tank T_1 for the n_1th time. It can be observed from Figure 16 that the resulting path must contain the filter in fragment F_1 or F_2.

• The third operation is concerned with the transportation of filtered beer from tank T_2 to the bottling station in fragment F_6. The goal constraint in this case can be developed with essentially the same approach as before:

$$\sum_{n_2=1}^{N_2} \tilde{x}_{4,b_o(n_2),t} = x_{6,t} \tag{59}$$

• The final operation is cleaning. This operation can be characterized as the task of moving the cleanser from supply fragment (F_7) to collection fragment (F_8) via a path containing the fouled units. Thus,

$$x_{7,t} = x_{8,t} \tag{60}$$

The operation objective in this application is to minimize the number of operation steps to produce a fixed amount of bottled beer in a given stage horizon H. Thus, the objective function of the corresponding BIP model can be expressed by eq 29 and an extra constraint must be included in this model to stipulate the given product quantity B, i.e.,

$$\sum_{t=1}^{H} x_{6,t} = B \tag{61}$$

The appropriate product quantity may be determined by maximizing B while satisfying all constraints [except (61)] in the aforementioned integer program.

In the present application, it is assumed that two repeat operations are allowed in filters and three are allowed in tanks. The total stage number (H) adopted in the BIP model is 15. It was found that the maximum number of bottling operations (B)

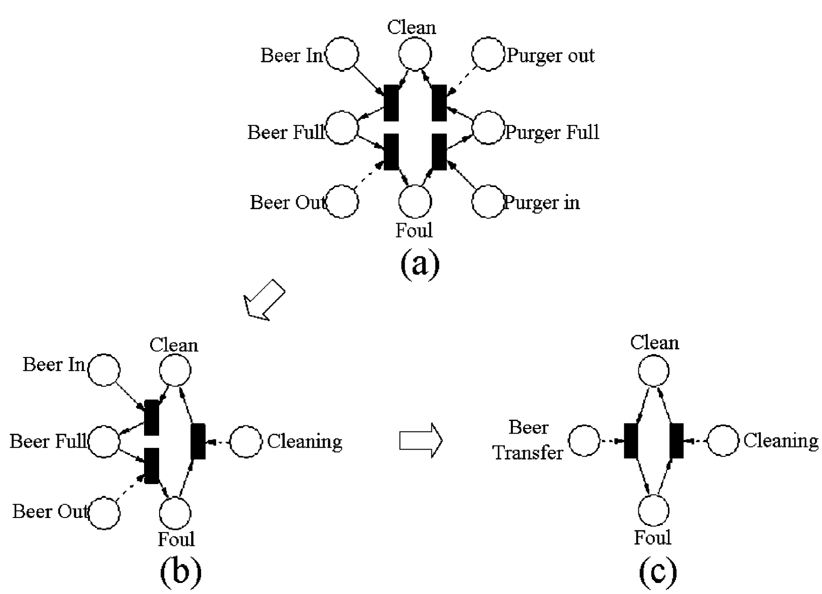

Figure 15. Petri-net models of process units. (a) Generalized model. (b) Tank model. (c) Filter model.

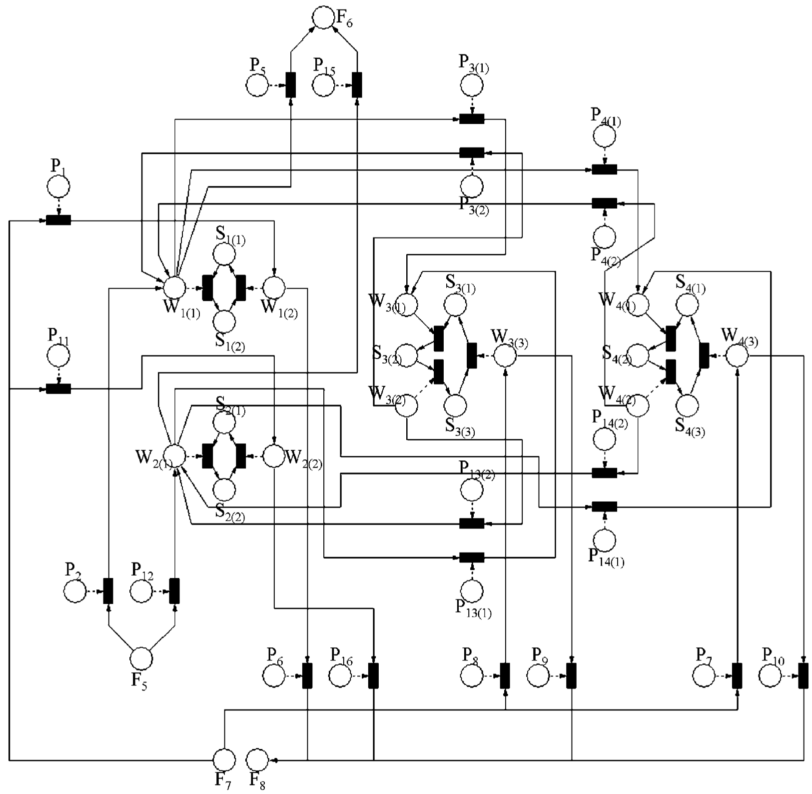

Figure 16. Modified path model of the beer filtration plant.

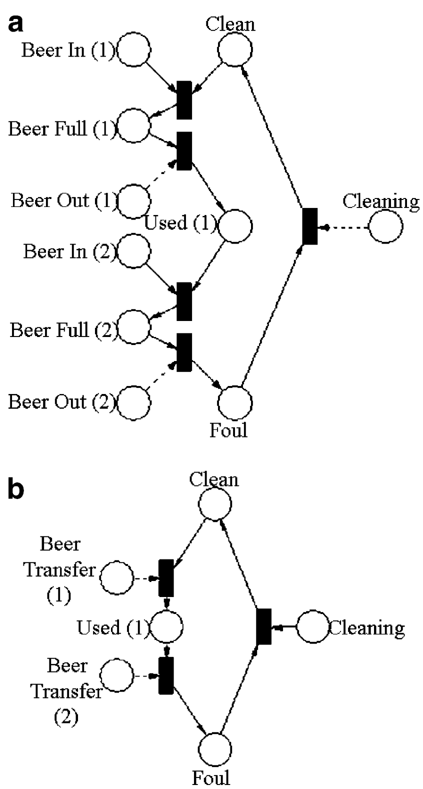

Figure 17. (a) Petri-net model of a tank which allows two consecutive operations. (b) Petri-net model of a filter which allows two consecutive operations.

is 5 and the minimum number of operation steps is 74. A total of 15 stages can be identified, and the operations performed at each stage can be found in Table 9. The resulting stage-based operating procedure is presented in Table 10.

5.2.3. Time-Based Operation. In order to determine the processing times of different operations, an additional index r is introduced to distinguish the variables representing the fragment states and also the work states of filters and tanks. The time-tracking mechanisms outlined in eqs 30−43 are still applicable for characterizing the dynamic behaviors of fragment states in the present case. On the other hand, the work states of process units should be described with an equation similar to eq 30:

Table 9. Stage-Based Operation Sequence of the Beer Filtration Plant

stage	task
1	replenish T_1 with beer via MMS2
2	transfer beer from T_1 to T_2 via MMS1 (beer filtration operation)
3	replenish T_1 with beer via MMS1 transfer filtered beer from T_2 to bottling system via MMS2
4	clean MMS1
5	transfer filtered beer from T_2 to bottling system via MMS1
6	replenish T_1 with beer via MMS1 clean MMS2
7	clean MMS1
8	transfer beer from T_1 to T_2 via MMS2 (beer filtration operation)
9	transfer filtered beer from T_2 to bottling system via MMS1 clean T_1
10	replenish T_1 with beer via MMS2 clean T_2
11	transfer beer from T_1 to T_2 via MMS2 (beer filtration operation)
12	replenish T_1 with beer via MMS1 clean MMS2
13	transfer filtered beer from T_2 to bottling system via MMS2 clean MMS1
14	transfer beer from T_1 to T_2 via MMS1 (beer filtration operation)
15	transfer filtered beer from T_2 to bottling system via MMS2

$$\tilde{\sigma}_{i,s,r,t+1} = \tilde{\sigma}_{i,s,r,t} + \tilde{\nu}_{i,s,r,t+1} - \tilde{\omega}_{i,s,r,t+1} \tag{62}$$

The time-tracking constraints of $\tilde{\sigma}_{i,s,r,t}$, $\tilde{\nu}_{i,s,r,t}$, and $\tilde{\omega}_{i,s,r,t+1}$ can be derived in a straightforward fashion on the basis of eqs 31−43. They are omitted in this paper for the sake of brevity.

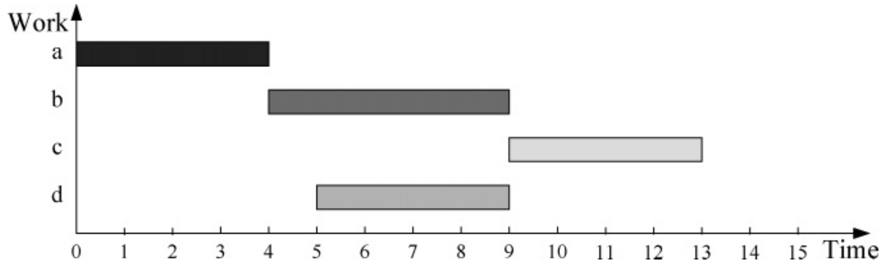

Figure 18. Optimal schedule of time-based procedure for operating the beer filtration plant (H = 15).

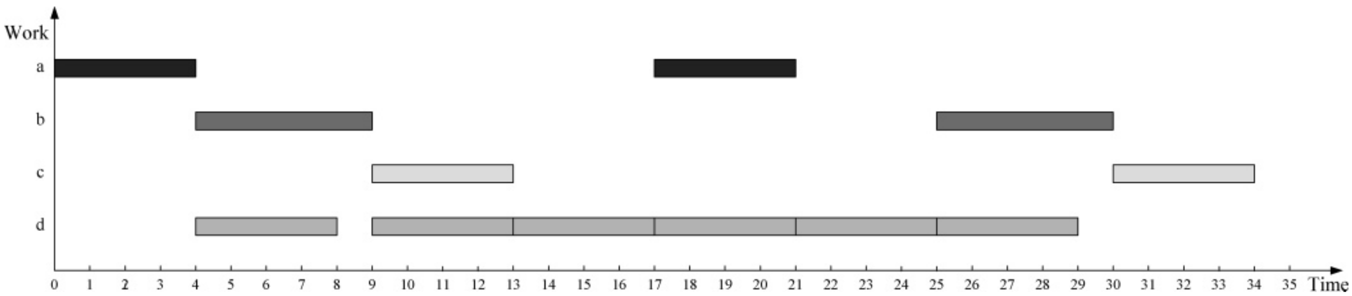

Figure 19. Optimal schedule of time-based procedure for operating the beer filtration plant (H = 35).

Table 10. Stage-Based Operating Procedure of the Beer Filtration Plant

stage	operation actions
1	open valves V_{12} and V_{13}
2	close valve V_{12}
	open valves V_3, V_4, and V_{14}
3	close valves V_3 and V_4
	open valves V_2 and V_5
4	close valves V_2 and V_5
	open valves V_1 and V_6
5	close valves V_1, V_6, V_{13}, and V_{14}
	open valves V_4 and V_5
6	close valves V_4 and V_5
	open valves V_2, V_3, V_{11}, and V_{16}
7	close valves V_2, V_3, V_{11}, and V_{16}
	open valves V_1, V_6, V_{13}, and V_{14}
8	close valves V_1 and V_6
9	close valve V_{13}
	open valves V_4, V_5, V_8, and V_9
10	close valves V_4, V_8, V_9, and V_{14}
	open valves V_7, V_{10}, V_{12}, and V_{13}
11	close valves V_7, V_{10}, and V_{12}
	open valve V_{14}
12	close valves V_5, V_{13}, and V_{14}
	open valves V_2, V_3, V_{11}, and V_{16}
13	close valves V_2, V_3, V_{11}, and V_{16}
	open valves V_1, V_6, V_{14}, and V_{15}
14	close valves V_1, V_6, and V_{15}
	open valves V_3 and V_4
15	open valve V_{15}

Due to the aforementioned changes in expressing the work states, it is also necessary to modify the constraints used in the stage-based case for describing the process models. Specifically, eq 53 should be changed to

$$(1 - \lambda_{i,s,t}) + (1 - \sum_r \tilde{\omega}_{i,s,r,t}) + \lambda_{i,s',t+1} \geq 1$$
$$(1 - \lambda_{i,s,t}) + \sum_r \tilde{\omega}_{i,s,r,t} + (1 - \lambda_{i,s',t+1}) \geq 1$$
$$\sum_r \tilde{\omega}_{i,s,r,t} \leq 1 \tag{63}$$

Equation 55 must also be rewritten as

$$\lambda_{i,s,t} + (1 - \tilde{\sigma}_{i,s,r,t}) \geq 1 \tag{64}$$

Table 11. Time-Based Operating Procedure of the Beer Filtration Plant

time	operation actions
0	open P_2, P_3, and P_{14}
4	close P_2 and P_3; open P_{13}
5	open P_1 and P_6
9	close P_1, P_6, P_{13}, and P_{14}; open P_4 and P_5
13	close P_5

Equations 54 and 56 should remain the same.

In the case studies for time-based procedures, it is assumed that all units must be cleaned after every operation. Thus, the modified path model of this system can be described by the Petri net in Figure 16. The residence times of beer and cleanser in filters are chosen to be 2 time units. The residence times associated with all other fragments and units are all assumed to be 1.

A time horizon (H) of 15 time units was first used in the BIP model. It was found that only one bottling operation can be performed and the minimum number of operation steps is 15. The corresponding operation schedule and operating procedure are presented in Figure 18 and Table 11, respectively. The time horizon was then extended to 35 time units. As a result, two batches of bottled beer can be produced with 47 operation steps. The corresponding schedule is shown in Figure 19.

6. Conclusion

A systematic strategy is presented in this paper for generating the optimal operating procedures to perform various batch operations. Specifically, the standard Petri-net models are developed to represent the fragments, valves, power-generating devices, and process units in this system. A binary integer program can be formulated on the basis of the system model constructed with these components. The optimal recipes containing the detailed operation steps can then be generated by solving this mathematical programming model according to various objective functions. Two distinct classes of operation modes can be identified: (1) stage-based operating procedures and (2) time-based operating procedures. The feasibility of the Petri-net based approach is clearly verified with the moderately complex examples given at the end of this paper. Additional case studies will be performed in the future to unequivocally demonstrate the effectiveness of the proposed method in more practical applications.

Literature Cited

(1) Chou, H. H.; Chang, C. T. Petri-Net Based Strategy to Synthesize the Operating Procedures for Cleaning Pipeline Networks. *Ind. Eng. Chem. Res.* **2005**, *44*, 114.

(2) Wang, Y. F.; Chou, H, H.; Chang, C. T. Generation of Batch Operating Procedures for Multiple Material-Transfer Tasks with Petri Net. *Comput. Chem. Eng.* **2005**, *29*, 1822.

(3) Rivas, J. R.; Rudd, D. F. Synthesis of Failure-Safe Operation. *AIChE J.* **1974**, *20*, 320.

(4) O'Shima, E. Safety Supervision of Valve Operation. *J. Chem. Eng. Jpn.* **1978**, *11*, 390.

(5) Foulkes, N. R.; Walton, M. J.; Andow, P. K.; Galluzzo, M. Computer-Aided Synthesis of Complex Pump and Valve Operations. *Comput. Chem. Eng.* **1988**, *12*, 1035.

(6) Uthgenannt, J. A. Path and Equipment Allocation for Multiple, Concurrent Processes on Networked Process Plant Units. *Comput Chem. Eng.* **1996**, *20*, 1081.

(7) Li, H. S.; Lu, M. L.; Naka, Y. A Two-Tier Methodology for Synthesis of Operating Procedures. *Comput. Chem. Eng.* **1997**, *21*, s899.

(8) Kim, J.; Moon, I. Synthesis of Safe Operating Procedure for Multi-Purpose Batch Processes Using SMV. *Comput. Chem. Eng.* **2000**, *24*, 385.

(9) ISA, *S88.01 Batch Control, Part 1: Models and Terminology*; ISA: Research Triangle Park, NC, 1995.

(10) Årzén, K.-E.; Johnsson, C. Object-Oriented SFC and ISA-S88.01 Recipes. *ISA Trans.* **1996**, *35*, 237.

(11) David, R.; Alla, H. Petri Nets for Modeling of Dynamic System - A Survey. *Automatica* **1994**, *30*, 175.

(12) Ferrarini, L.; Piroddi, L. Modular Design and Implementation of a Logic Control System for a Batch Process. *Comput. Chem. Eng.* **2003**, *27*, 983.

(13) Peterson, J. L. *Petri Net Theory and the Modeling of Systems*; Pentice-Hall: Englewood Cliffs, NJ, 1981.

(14) Wang, Y. F.; Chang, C. T. A Hierarchical Approach to Construct Petri Nets for Modeling Fault Propagation Mechanisms in Sequential Operations. *Comput. Chem. Eng.* **2003**, *27*, 259.

(15) Wang, Y. F.; Chang, C. T. Petri-Net Based Deductive Reasoning Strategy for Fault Identification in Batch Processes. *Ind. Eng. Chem. Res.* **2004**, *43*, 2704.

(16) Karassik, I. J.; McGuire, J. T. *Centrifugal Pumps*, 2nd ed.; Chapman & Hall: New York, 1998; pp 885−887.

(17) Raman, R.; Grossmann, I. E. Relation between MILP Modeling and Logical Inference for Chemical Process Synthesis. *Comput. Chem. Eng.* **1991**, *15*, 73.

(18) Lai, Y. H. Process Control of Brewery Plant by EVALPSN. MS Thesis, NTUST, Taipei, 2005.

Received for review August 8, 2006
Revised manuscript received February 12, 2007
Accepted February 19, 2007

IE0610400