PETRI-NET BASED INTEGER PROGRAMS FOR SYNTHESIZING OPTIMAL MATERIAL-TRANSFER PROCEDURES IN PIPELINE NETWORKS

Jeng-Wen Lai, Hung-Hsiang Chou, and Chuei-Tin Chang*

ABSTRACT

Transferring materials through the pipeline network is a basic operation in almost every batch chemical plant. Traditionally, the tasks for conjecturing the needed operation steps are carried out manually on an *ad hoc* basis. This approach is often time-consuming for industrial processes and, furthermore, the resulting recipe may be error-prone. The aim of this paper is thus to develop a systematic strategy to generate the optimal operating procedures with Petri-net based integer programs. Specifically, the shortest material-transfer routes are selected on the basis of Petri-net representation of the path structure in pipeline network. The equipment models are then incorporated into this path model to create a complete system model. An integer program can therefore be constructed accordingly to identify the detailed operation steps. Finally, a realistic example is presented at the end of this paper to demonstrate the effectiveness and correctness of the proposed approach.

Key Words: petri net, integer program, pipeline network, batch process, material transfer.

I. INTRODUCTION

Transferring process materials from one unit to another via connecting pipelines is one of the basic operations that must be performed in many batch chemical plants. The goals of such operations may be multifaceted, e.g., mixing, separation, reaction, energy transfer and even cleaning (Chou and Chang, 2005; Wang *et al*., 2005). 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 process, the demands of these tasks for time and effort may be overwhelming and the resulting recipe is often error-prone. Thus, in order to reduce work load and also to enhance operation performance, it is highly desirable to develop a systematic strategy to synthesize the needed recipes correctly and efficiently.

*Corresponding author. (Tel: 886-6-275-7575 ext. 62663; Fax: 886-6-234-4496; Email: ctchang@mal.ncku.edu.tw)

The authors are with the Department of Chemical Engineering, National Cheng Kung University, Tainan, Taiwan 701, R.O.C.

In a pioneering work, Rivas and Rudd (1974) proposed a method for the synthesis of failure-safe procedures to help operators make proper decisions during emergency situations. A valve operation sequence can be quickly determined to reach the given operation objective. O'Shima (1978) 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*. (1988) 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 (1996) used digraph models to describe the network of interconnected processing equipment. The material transfer routes and the required operating procedures can be obtained using a graph search method.

Although interesting results have been generated by the above studies, the proposed methods are still not mature enough for practical applications. In fact, 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 (1995). It was shown that a Sequential Function Chart (SFC) is suitable for representing the hierarchical procedural model specified in this standard (Årzén, 1996). Since SFC is essentially derived from the basic Petri net concepts, it is the intention of the present work to use the latter as the modeling tool to accurately describe the material-transfer operations in pipeline networks. The mathematical representation of the ordinary Petri net is provided by Peterson (1981). A detailed review of the Petri-net elements and the transition enabling and firing rules can be found elsewhere (David and Alla, 1994; Wang, and Chang, 2004).

The focus of the present study is on the synthesis of optimal operating procedures to achieve any given material-transfer task on the basis of Petri-net models. Specifically, an accurate component model is first constructed for each pipeline fragment in the system. The fragment models are then connected according to the network configuration to build a Petri net in which all material-transfer paths can be embedded. This path model is described in section II. It is then used to construct an integer program for route identification. The program formulations are presented in section III. The Petri-net representations of the valves, pumps and compressors are then given in section IV. The complete system model can be assembled by attaching these equipment models to the path model. In the next section, various logic constraints of the operation steps are developed according to the system model. Two candidate objective functions are also proposed. The optimal operating procedures can be synthesized automatically by solving the corresponding mathematical programming models. Finally, the results of applying the proposed synthesis procedure to an industrial-size system are presented in the last section.

II. REPRESENTATION OF MATERIAL-TRANSFER PATHS

The first critical issue in modeling any network should be the division of the given system into distinct components. The concept of the piping fragments (Foulkes *et al.*, 1988) is adopted in this work for this purpose. In particular, a fragment is defined as a collection of pipeline branches and/or processing units separated from other fragments (or the environment) by valves, pumps and other means of flow blockages in the pipeline network. Let us consider Fig. 1 as an example. Eight fragments can be

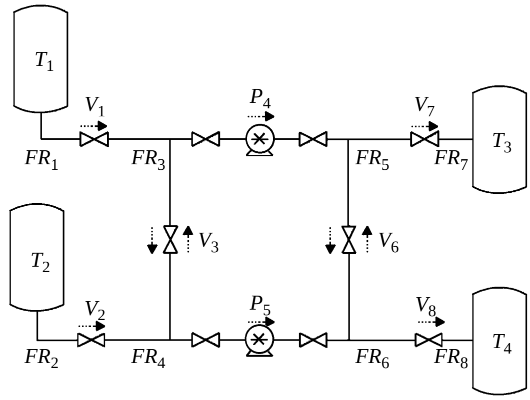

Fig. 1 A Typical Example of a Pipeline Network

identified according to this definition, i.e. FR_1 - FR_8. In this case, 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 external 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 most basic structure of fragment, i.e., a pipe branch isolated by an inlet valve and an outlet valve [see Fig. 2(a)]. Notice that, in this case, the flow in either valve is allowed only in one direction. The corresponding Petri-net model is presented in Fig. 2(b). 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. The place PV_1^O is used to reflect the connection status of FR with its upstream fragment, and PV_2^O is used for the same purpose concerning the downstream fragment. The transitions CN_1 and CN_2 can be considered as the events that establish these connections. On the other hand, if both valves permit bi-directional material transfer, the fragment model depicted in Fig. 2(b) should be changed to the one shown in Fig. 2(c). Notice that each transition in the former Petri net is now replaced by two transitions to denote the material transfer actions to and from the fragment FR via the corresponding valve.

In principle, all mass-transfer paths can be found in a Petri net assembled by connecting the fragment models according to the network configuration. However, it should be noted that some of these paths may take the form of infinite loops. Let us use the pipeline network in Fig. 1 to illustrate this possibility.

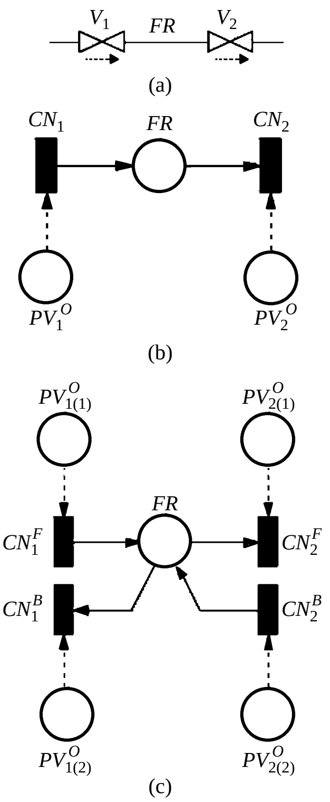

Fig. 2 (a) Basic Structure of a Piping Fragment with Two Single-Direction Valves, (b) Petri-Net Model of a Basic Fragment with Two Single-Direction Valves, (c) Basic Structure of a Basic Fragment with Two Bi-Direction Valves

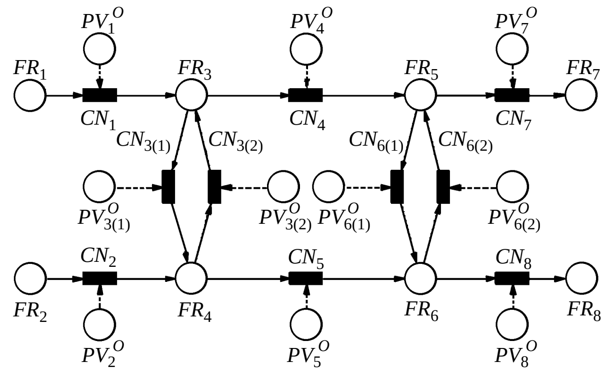

Fig. 3 Path Model of the Example Network in Fig. 1

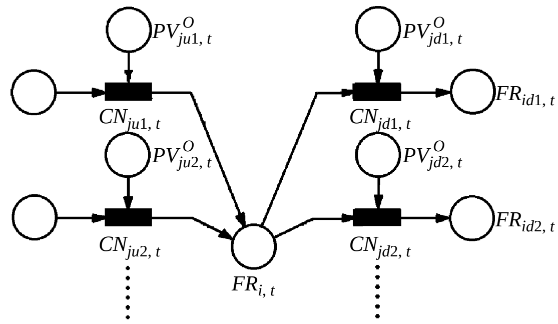

Fig. 4 The Upstream and Downstream Connections of a Fragment.

In the corresponding path model presented in Fig. 3, it is obvious that a token may travel endlessly in one of the following two loops: (1) $FR_3 \rightarrow FR_4 \rightarrow FR_3 \rightarrow \cdots$ and (2) $FR_5 \rightarrow FR_6 \rightarrow FR_5 \rightarrow \cdots$. Clearly these loops are caused by the bi-directional valves. There are of course other structural features which could result in additional loops. For example, let us reverse the directions of pump P_5 and valves V_2 and V_8 in the pipeline network in Fig. 1. The roles of tanks T_2 and T_4 in the resulting system are also reversed from source to sink and vice versa. As a result, another loop can be identified, i.e., $FR_3 \rightarrow FR_5 \rightarrow FR_6 \rightarrow FR_4 \rightarrow FR_3 \cdots$. In practical applications, these looping paths cannot be adopted as candidate routes for material transfer. Additional constraints are thus needed to limit the token movements in the Petri-net model.

III. IDENTIFICATION OF OPTIMAL ROUTES

Since there can be more than one route emanating from a particular source fragment to the sink fragments of a pipeline network, it is desirable to identify the shortest route among all possible candidates. This task can be achieved by constructing an integer program on the basis of a path model. In this program, two different types of binary variables, i.e., $x_{i,\,t}$ and $y^O_{j,\,t}$, are adopted to represent the token numbers in places representing the fragment states (FR_i) and connection status (PV^O_j) respectively during the tth operation stage. For illustration convenience, let us consider the generalized fragment model presented in Fig. 4. The causal relations between the fragment state of FR_i and those of its downstream fragments can be translated into a set of logic constraints, i.e.

$$(1 - x_{i,\,t}) + (1 - y^O_{jd,\,t}) + x_{id,\,t} \geq 1$$

$$jd \in \boldsymbol{JD}_i \quad id \in \boldsymbol{ID}_{jd} \tag{1}$$

where, $x_{i,\,t} \in \{0, 1\}$; $y^O_{j,\,t} \in \{0, 1\}$; $\boldsymbol{JD}_i = \{jd_1, jd_2, \cdots\}$; $\boldsymbol{ID}_{jd_k} = \{id_k\}$.

As mentioned previously, additional constraints must be imposed to restrict the token flow so that loops can be eliminated in the search process. In practical systems, it is reasonable to assume that the source and sink fragments are not within loops. If this is the case, there should be only one downstream connection and/or one upstream connection for any fragment FR_i in a material-transfer route, i.e.

$$(1 - x_{i,\,t}) + \sum_{jd \in \boldsymbol{JD}_i} y^O_{jd,\,t} = 1 \tag{2}$$

$$(1 - x_{i,t}) + \sum_{jd \in \boldsymbol{JU}_i} y^O_{ju,t} = 1 \tag{3}$$

where, $\boldsymbol{JU}_i = \{ju_1, ju_2, \cdots\}$. Notice that constraints (1), (2) and (3) must be imposed upon all fragments except that (1) and (2) cannot be used to describe the flow connections of sinks and (3) is not applicable in the case of sources.

Besides the above-mentioned constraints, we still need to incorporate additional ones for the purpose of achieving specific operation objectives. For example, the simplest material-transfer goal is to establish a transportation route between a source and a sink. In this case, the sums of two corresponding binary variables $x_{i,t}$ in all stages should be set to 1, i.e.

$$\sum_{t=1}^{T} x_{i,t} = 1 \tag{4}$$

where, T is a sufficiently large positive integer and i is the label of a designated source or sink fragment. In addition, the sums associated with the other sources and sinks should all be set to 0. Notice further that, if there is a need to identify transportation routes from multiple sources to multiple sinks, then the same approach can also be taken to impose constraints on the designated fragments. On the other hand, if the operation objective is to clean the entire pipeline network by moving detergent through every fragment, then the following constraint should be imposed upon all fragments:

$$\sum_{t=1}^{T} x_{i,t} \ge 1 \tag{5}$$

Finally, a reasonable choice of the objective function for identifying the optimal material-transfer route can be expressed as:

$$\underset{Y^o}{\mathrm{Min}}[\sum_t \sum_i x_{i,t}] \tag{6}$$

where

$$Y^O = \begin{bmatrix} y^O_{1,1} & y^O_{1,2} & \cdots & y^O_{1,t} & \cdots & y^O_{1,T} \\ y^O_{2,1} & y^O_{2,2} & \cdots & y^O_{2,t} & \cdots & y^O_{2,T} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ y^O_{j,1} & y^O_{j,2} & \cdots & y^O_{j,t} & \cdots & y^O_{j,T} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \end{bmatrix}$$

Example 1. Let us consider the path model in Fig. 3. Constraints (1) - (3) can be written respectively for every fragment. Specifically,

- FR_1:

$$(1 - x_{1,t}) + (1 - y^O_{1,t}) + x_{3,t} \ge 1$$

$$(1 - x_{1,t}) + y^O_{1,t} = 1$$

- FR_2:

$$(1 - x_{2,t}) + (1 - y^O_{2,t}) + x_{4,t} \ge 1$$

$$(1 - x_{2,t}) + y^O_{2,t} = 1$$

- FR_3:

$$(1 - x_{3,t}) + (1 - y^O_{4,t}) + x_{5,t} \ge 1$$

$$(1 - x_{3,t}) + (1 - y^O_{3,t}) + x_{4,t} \ge 1$$

$$(1 - x_{3,t}) + y^O_{3(1),t} + y^O_{4,t} = 1$$

$$(1 - x_{3,t}) + y^O_{3(2),t} + y^O_{1,t} = 1$$

- FR_4:

$$(1 - x_{4,t}) + (1 - y^O_{5,t}) + x_{6,t} \ge 1$$

$$(1 - x_{4,t}) + (1 - y^O_{3,t}) + x_{3,t} \ge 1$$

$$(1 - x_{4,t}) + y^O_{3(2),t} + y^O_{5,t} = 1$$

$$(1 - x_{4,t}) + y^O_{3(1),t} + y^O_{2,t} = 1$$

- FR_5:

$$(1 - x_{5,t}) + (1 - y^O_{7,t}) + x_{7,t} \ge 1$$

$$(1 - x_{5,t}) + (1 - y^O_{6,t}) + x_{6,t} \ge 1$$

$$(1 - x_{5,t}) + y^O_{6(1),t} + y^O_{7,t} = 1$$

$$(1 - x_{5,t}) + y^O_{6(2),t} + y^O_{4,t} = 1$$

- FR_6:

$$(1 - x_{6,t}) + (1 - y^O_{8,t}) + x_{8,t} \ge 1$$

$$(1 - x_{6,t}) + (1 - y^O_{6,t}) + x_{5,t} \ge 1$$

$$(1 - x_{6,t}) + y^O_{6(2),t} + y^O_{8,t} = 1$$

$$(1 - x_{6,t}) + y^O_{6(1),t} + y^O_{5,t} = 1$$

- FR_7:

$$(1 - x_{7,t}) + y^O_{7,t} = 1$$

- FR_8:

$$(1 - x_{8,t}) + y^O_{8,t} = 1$$

If we simply want to accomplish a single material-transfer task from FR_1 to FR_8, then it is obvious that only one operation stage (T = 1) is needed. Also, additional constraints should be imposed upon the source and sink fragments in order to achieve this goal. In particular,

$$x_{1,\,1} = x_{8,\,1} = 1$$

$$x_{2,\,1} = x_{7,\,1} = 0 \quad (7)$$

The solution of the corresponding integer program was obtained with the CPLEX module in GAMS. The optimal route in this case was found to be $FR_1 \rightarrow FR_2 \rightarrow FR_6 \rightarrow FR_7 \rightarrow FR_8$.

On the other hand, the optimal cleaning routes of the pipeline network in Fig. 3 were obtained by setting $T = 5$. It was found that the detergent should be transferred concurrently in one stage along two distinct non-overlapping routes, i.e., $FR_1 \rightarrow FR_3 \rightarrow FR_5 \rightarrow FR_7$ and $FR_2 \rightarrow FR_4 \rightarrow FR_6 \rightarrow FR_8$.

IV. REPRESENTATION OF EQUIPMENT OPERATIONS

In order to generate the specific operation steps to accomplish the material-transfer tasks according to a given criterion, the Petri-net representation of a pipeline network must contain not only the fragment models but also models of the installed equipment, i.e., valves, pumps and compressors. The Petri-net model of a single-directional valve is presented in Fig. 5. Here, the places PV^O and PV^C denote two opposite valve positions respectively, i.e., open and close. Notice that the former place should be connected (with a static test arc) to the transition representing the event that establishes the connection between the upstream and downstream fragments of the corresponding valve. The transitions TV^O and TV^C represent the valve-switching actions from to PV^C and PV^O vice versa. The input place PC^O of the transitions TV^O can 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.

On the other hand, since the operating procedures of pumps, compressors and their isolation valves can be regarded as well-established industrial practices, e.g., see Karassik and McGuire (1998), their detailed steps are not described in the equipment models for the sake of simplicity. Thus, the Petri net presented in Fig. 5 is also used to model the power-generating systems in this study. 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 operation actions to turn on and of the pump/compressor system.

Finally, notice that the connection status between two adjacent fragments is described with two distinct places in the path model (see Fig. 3) if the corresponding valve is bi-directional. In this study, two distinct single-directional valve models are

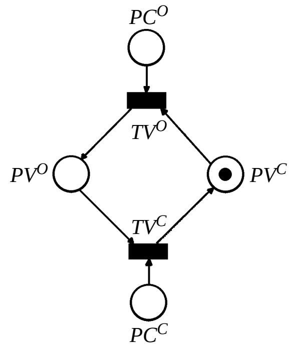

Fig. 5 A Standard Valve Model

attached to these two places respectively for the purpose of characterizing its operation steps. Since the places representing the valve positions and also control commands are all duplicated by such a modeling approach, it is necessary to impose extra constraints in the integer program in order to resolve the conflicting operation decisions which may be generated in the optimal solution.

V. IDENTIFICATION OF OPTIMAL OPERATION STEPS

For the purpose of building the integer program, every single-directional equipment model has to be translated into a set of logic constraints. Let us use the binary variables $y^O_{j,\,t}$ and $y^C_{j,\,t}$ to respectively denote the token numbers in the two places representing the states of equipment j during operation stage t, and use $z^O_{j,\,t}$ and $z^C_{j,\,t}$ to denote the token numbers in places representing the corresponding control commands. The token movement in an equipment model can therefore be described as:

$$(1 - y^{OI}_{j,\,t}) + (1 - z^C_{j,\,t}) + y^C_{j,\,t} \geq 1$$

$$(1 - y^{OI}_{j,\,t}) + z^C_{j,\,t} + y^O_{j,\,t} \geq 1$$

$$(1 - y^{CI}_{j,\,t}) + (1 - z^O_{j,\,t}) + y^O_{j,\,t} \geq 1$$

$$(1 - y^{CI}_{j,\,t}) + z^O_{j,\,t} + y^C_{j,\,t} \geq 1 \qquad (8)$$

In the above constraints, $y^{OI}_{j,\,t}$ and $y^{CI}_{j,\,t}$ represent respectively the initial values of $y^O_{j,\,t}$ and $y^C_{j,\,t}$ during stage t. 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, i.e. $y^{OI}_{j,\,t} = y^O_{j,\,t-1}$ and $y^{CI}_{j,\,t} = y^C_{j,\,t-1}$. 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

Table 1 Operation steps for cleaning the pipeline network in Fig. 1

Procedure	Operation steps
1	Open valves V_1, V_2, V_7, and V_8. Switch on pumps P_4 and P_5.
2	Switch off pumps P_4 and P_5. Close valves V_1 and V_2.

corresponding binary variables $y^{OI}_{j,t}$s should be set to 0 and $y^{CI}_{j,t}$s should be 1 in all operation stages.

In addition to the constraints given above, there are other auxiliary constraints in the proposed model which are designed to enhance search efficiency. First of all, the two equipment states of any valve or pump should be mutually exclusive, i.e.

$$y^O_{j,t} + y^C_{j,t} = 1 \quad (9)$$

Second, if a valve is open (or closed) initially in an operation stage, then it is meaningless to execute the operation step to open (or close) the same valve. Thus, the following constraints are adopted to prevent such possibilities

$$y^{OI}_{j,t} + z^O_{j,t} \le 1 \quad (10)$$

$$y^{CI}_{j,t} + z^C_{j,t} \le 1 \quad (11)$$

Third, for the purpose of ensuring practical applicability, it is assumed in this work that every valve (except the ones installed at the sources) in the pipeline network can only be operated at most once. In other words, the following constraints are also incorporated in the mathematical programming model:

$$z^O_{j,t} + z^C_{j,t} \le 1 \quad (12)$$

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

$$(1 - y^O_{j,t}) + z^O_{j,t} \ge 1$$

$$(1 - y^O_{j,t}) + z^C_{j,t} \ge 1 \quad (13)$$

Notice that the equipment model of a bi-directional valve is built with two single-directional models. Extra constraints are needed to reconcile the conflicting control commands embedded in this model. In particular, the fictitious single-directional valves cannot be both open, i.e.

$$y^O_{j(1),t} + y^O_{j(2),t} \le 1 \quad (14)$$

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

$$z^O_{j(1),t} + z^O_{j(2),t} \le 1$$

$$z^C_{j(1),t} + z^C_{j(2),t} \le 1 \quad (15)$$

Furthermore, all possible states of a bi-directional valve can be classified accordingly as:

1. $y^O_{j(1),t} = 0$, $y^C_{j(1),t} = 0$, $y^O_{j(2),t} = 0$, $y^C_{j(2),t} = 0$;

2. $y^O_{j(1),t} = 1$, $y^C_{j(1),t} = 0$, $y^O_{j(2),t} = 0$, $y^C_{j(2),t} = 0$;

3. $y^O_{j(1),t} = 0$, $y^C_{j(1),t} = 0$, $y^O_{j(2),t} = 1$, $y^C_{j(2),t} = 0$.

The transition from one state to another can be realized by issuing control commands to the fictitious valves. There are 6 possible combinations. To resolve the conflicting control commands required in these transition processes, let us use the binary variables $u^O_{j,t}$ and $u^C_{j,t}$ to represent respectively the actual control commands for opening and closing the bi-directional valve j during stage t. The correspondence between the fictitious and actual commands of a bi-directional valve is summarized in Table 2.

Table 2 The correspondence between the fictitious and actual commands of a bi-directional valve

$z^O_{j(1),t}$	$z^C_{j(1),t}$	$z^O_{j(2),t}$	$z^C_{j(2),t}$	$u^O_{j,t}$	$u^C_{j,t}$
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

The fictitious commands listed in the first two rows of Table 2 are adopted to change from state 1 to state 2 and vice versa. On the other hand, the fictitious commands 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 commands, the corresponding actual command should be taken accordingly. These logic relations can be expressed in the form of inequality constraints, i.e.

$$(1 - z^O_{j(1),t}) + z^C_{j(2),t} + u^O_{j,t} \geq 1$$

$$(1 - z^C_{j(1),t}) + z^O_{j(2),t} + u^C_{j,t} \geq 1$$

$$(1 - z^O_{j(2),t}) + z^C_{j(1),t} + u^O_{j,t} \geq 1$$

$$(1 - z^C_{j(2),t}) + z^O_{j(1),t} + u^C_{j,t} \geq 1 \qquad (16)$$

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 Eqs. (12) and (15).

The fictitious commands in the 5th and 6th rows of Table 2 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 is carried out in practice, the bi-directional valve nust be first opened and then closed or vice versa. This implies that the actual valve position is unchanged during stage *t* and thus no real actions should be taken. Following are the inequality constraints representing the inference rules given in row 5:

$$(1 - z^O_{j(1),t}) + (1 - z^C_{j(2),t}) + (1 - u^O_{j,t}) \geq 1$$
$$(1 - z^O_{j(1),t}) + (1 - z^C_{j(2),t}) + (1 - u^C_{j,t}) \geq 1 \qquad (17)$$

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

$$(1 - z^C_{j(1),t}) + (1 - z^O_{j(2),t}) + (1 - u^C_{j,t}) \geq 1$$
$$(1 - z^C_{j(1),t}) + (1 - z^O_{j(2),t}) + (1 - u^O_{j,t}) \geq 1 \qquad (18)$$

Notice that the last row in Table 2 is associated with the possibilities that the valve remain in its original state during stage *t*. In this situation, there should not be any actual action either. The corresponding constraints are:

$$z^O_{j(1),t} + z^O_{j(2),t} + z^C_{j(1),t} + z^C_{j(2),t} + (1 - u^O_{j,t}) \geq 1$$
$$z^O_{j(1),t} + z^O_{j(2),t} + z^C_{j(1),t} + z^C_{j(2),t} + (1 - u^C_{j,t}) \geq 1$$
$$(19)$$

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

$$u^O_{j,t} = z^O_{j,t}$$
$$u^C_{j,t} = z^C_{j,t} \qquad (20)$$

The objective function of the resulting integer program can be the same as that defined in Eq. (5) with different decision variables, i.e.

$$\min_{Z^O, Z^C} \left[\sum_t \sum_i x_{i,t}\right] \qquad (21)$$

where

$$Z^O = \begin{bmatrix} z^O_{1,1} & z^O_{1,2} & \cdots & z^O_{1,t} & \cdots & z^O_{1,T} \\ z^O_{2,1} & z^O_{2,2} & \cdots & z^O_{2,t} & \cdots & z^O_{2,T} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ z^O_{j,1} & z^O_{j,2} & \cdots & z^O_{j,t} & \cdots & z^O_{j,T} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \end{bmatrix}$$

$$Z^C = \begin{bmatrix} z^C_{1,1} & z^C_{1,2} & \cdots & z^C_{1,t} & \cdots & z^C_{1,T} \\ z^C_{2,1} & z^C_{2,2} & \cdots & z^C_{2,t} & \cdots & z^C_{2,T} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ z^C_{j,1} & z^C_{j,2} & \cdots & z^C_{j,t} & \cdots & z^C_{j,T} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \end{bmatrix}$$

Alternatively, if the operation objective is to simplify the operation procedure, then it may be desirable to minimize the total number of actual operation steps. In such cases, the objective function can be expressed as:

$$\min_{Z^O, Z^C} \left[\sum_t \sum_j u^O_{j,t} + \sum_t \sum_j u^C_{j,t}\right] \qquad (22)$$

VI. CASE STUDIES

The pipeline network described in Foulkes *et al.* (1998) is adopted in the present work as a realistic example to demonstrate the capability of the proposed method. The network contains eight storage tanks, 36 valves and four pumps (see Fig. 6). A total of five source fragments, i.e., FR_1 - FR_5, twenty internal fragments, i.e., FR_6 - FR_{25}, and six sink fragments, i.e., $FR_{26(1)}$ - $FR_{28(1)}$ and $FR_{26(2)}$ - $FR_{28(2)}$, can be defined in this system. It is assumed that there are no upper limits imposed upon the amounts of raw materials stored in the source tanks, i.e., T_1 - T_5, or the capacities of the sink tanks, i.e., M_1 - M_3. In addition, the spent materials gathered from separate transfer routes are allowed to be stored in the same sink tank. 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 Fig. 7). Notice that, to enhance the legibility of this Petri net, the labels of the places and transitions are abbreviated and their definitions are given in the figure legend. Notice also that the equipment models of the valves and pumps can be attached in a straightforward fashion to this path model. For the sake of conciseness, the resulting Petri net is not shown in this paper. Finally, it is assumed in

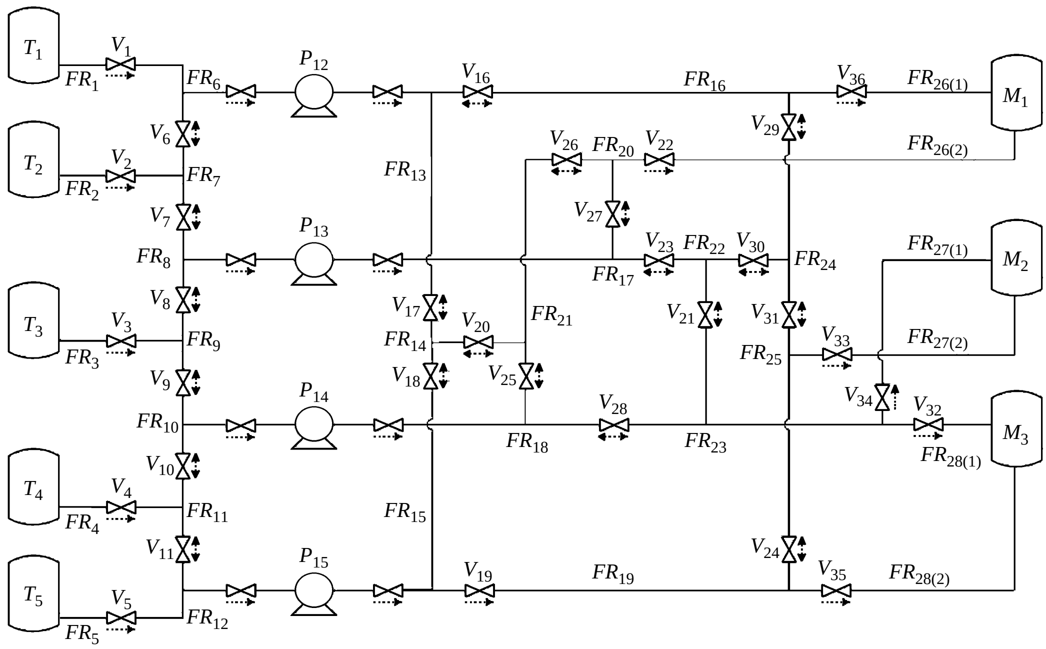

Fig. 6 A Complex Pipeline Network

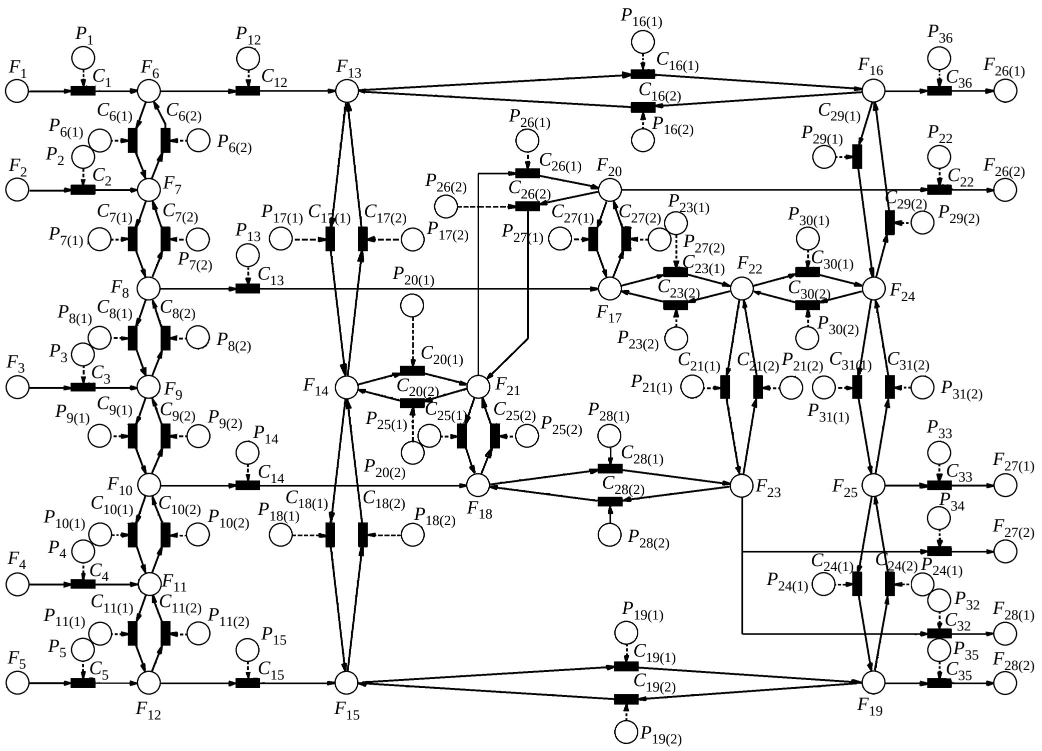

Fig. 7 Path Model of the Complex Network in Fig. 9 ($F \equiv FR$; $P \equiv PV^{O}$; $C \equiv CN$)

the cases presented below that the all valves are closed and all pumps switched off initially, i.e.

$$y^{OI}_{j,1} = 0$$

$$y^{CI}_{j,1} = 1$$

Let us first consider the task of transferring material from source T_1 to sink M_2. The material-transfer routes identified with the two objective functions given in Eqs. (21) and (22) are essentially the same, i.e.

Table 3 Operation steps to transfer material from T_1 to M_2 in the pipeline network in Fig. 6

Procedure	Operation steps
1	Open valves V_1, V_{16}, V_{29}, V_{31} and V_{33}. Switch on pump P_{12}.
2	Switch off pump P_{12}. Close valve V_1.

Table 4 Operation steps to transfer material A and material B from T_1 and T_3 to M_3 in the pipeline network in Fig. 6

Procedure	Operation steps
1	Open valves V_1, V_3, V_9, V_{17} V_{18}, V_{19}, V_{28}, V_{32}, and V_{35}. Switch on pumps P_{12} and P_{14}.
2	Switch off pumps P_{12} and P_{14}. Switch on pumps V_1 and V_3.

$$FR_1 \rightarrow FR_6 \rightarrow FR_{13} \rightarrow FR_{16} \rightarrow FR_{24} \rightarrow FR_{25} \rightarrow FR_{27(1)}$$

The corresponding optimal operating procedure can be found in Table 3.

Next, let us consider the operating procedure for mixing material ***A*** in T_1 and material ***B*** in T_3. These two raw materials must both be transferred into tank M_3 for the required mixing operation. Again, it was found that both objective functions yield the same solution. Specifically, ***A*** and ***B*** can be delivered into M_3 concurrently in a single operation stage via the following two routes:

- ***A***: $FR_1 \rightarrow FR_6 \rightarrow FR_{13} \rightarrow FR_{14} \rightarrow FR_{15} \rightarrow FR_{19} \rightarrow FR_{28(2)}$
- ***B***: $FR_3 \rightarrow FR_9 \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{28(1)}$

The best operating procedures in this case are listed in Table 4.

In the last case study, optimal cleaning procedures were synthesized by use of the proposed integer program on the basis of the objective functions specified in Eqs. (21) and (22). 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. The selected material-transfer routes are listed in Tables 5 and 6. The former solution consists of the fewest fragments and the latter requires the smallest number of operation steps. Notice that, in both cases, the implied cleaning tasks must be carried out sequentially in two stages. During each stage, multiple material-transfer operations can be executed concurrently via the selected routes. Notice also that the corresponding operation steps can also be obtained with the same integer program. The optimal operating procedures of the two cases considered here can be found in Tables 7 and 8 respectively.

Table 5 Cleaning routes of the pipeline network in Fig. 6 containing the minimum total number of fragments

Implementation stage	Route(s)
1	• $FR_1 \rightarrow FR_6 \rightarrow FR_{13} \rightarrow FR_{16} \rightarrow FR_{26(1)}$ • $FR_4 \rightarrow FR_{11} \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{27(2)}$ • $FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{14} \rightarrow FR_{21} \rightarrow FR_{20} \rightarrow FR_{26(2)}$
2	• $FR_2 \rightarrow FR_7 \rightarrow FR_8 \rightarrow FR_{17} \rightarrow FR_{22} \rightarrow FR_{24} \rightarrow FR_{25} \rightarrow FR_{27(1)}$ • $FR_3 \rightarrow FR_9 \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{28(1)}$ • $FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{19} \rightarrow FR_{28(2)}$

Table 6 Cleaning routes of the Pipeline Network in Fig. 6 requiring the minimum total number of operation steps

Implementation stage	Route(s)
1	• $FR_4 \rightarrow FR_{11} \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{28(1)}$ • $FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{19} \rightarrow FR_{28(2)}$
2	• $FR_1 \rightarrow FR_6 \rightarrow FR_{13} \rightarrow FR_{16} \rightarrow FR_{26(1)}$ • $FR_2 \rightarrow FR_7 \rightarrow FR_8 \rightarrow FR_{17} \rightarrow FR_{22} \rightarrow FR_{24} \rightarrow FR_{25} \rightarrow FR_{27(1)}$ • $FR_3 \rightarrow FR_9 \rightarrow FR_{10} \rightarrow FR_{18} \rightarrow FR_{23} \rightarrow FR_{27(2)}$ • $FR_5 \rightarrow FR_{12} \rightarrow FR_{15} \rightarrow FR_{14} \rightarrow FR_{21} \rightarrow FR_{20} \rightarrow FR_{26(2)}$

Table 7 Operation steps to realize the cleaning tasks specified in Table 5

Procedure	Operation steps
1	Open valves V_1, V_4, V_5, V_{10}, V_{16}, V_{18}, V_{20}, V_{22}, V_{26}, V_{28}, V_{34}, and V_{36}. Switch on pumps P_{12}, P_{14} and P_{15}.
2	Switch off pumps P_{12}, P_{14} and P_{15}. Close valves V_1, V_4 and V_5
3	Close valves V_{10}, V_{16}, V_{18}, V_{20}, V_{22}, V_{26}, V_{28}, V_{34} and V_{36}. Open valves V_2, V_3, V_5, V_7, V_9, V_{19}, V_{23}, V_{30}, V_{31}, V_{32}, V_{33} and V_{35}. Switch on pumps P_{13}, P_{14} and P_{15}.
4	Switch off pumps P_{13}, P_{14} and P_{15}. Close valves V_2, V_3, and V_5.

Table 8 Operation steps to realize the cleaning tasks specified in Table 6

Procedure	Operation steps
1	Open valves V_4, V_5, V_{10}, V_{19}, V_{28}, V_{32}, and V_{35}. Switch on pumps P_{14} and P_{15}.
2	Switch off pumps, P_{14} and P_{15}. Close valves, V_4 and V_5
3	Close valves V_4, V_5, V_{10}, V_{19}, V_{32}, and V_{35}. Open valves V_1, V_2, V_3, V_5, V_7, V_9, V_{16}, V_{18}, V_{20}, V_{22}, V_{23}, V_{26}, V_{30}, V_{31}, V_{33}, V_{34} and V_{36}. Switch on pumps P_{12}, P_{13}, P_{14} and P_{15}.
4	Switch off pumps P_{12}, P_{13}, P_{14} and P_{15}. Close valves V_1, V_2, V_3, and V_5.

VII. CONCLUSION

A systematic strategy is presented in this paper for generating the optimal operating procedures to perform various designated material-transfer tasks within a given pipeline network. Specifically, several standard Petri-net models are developed to represent the fragments, valves and pumps in this system. The objective function and logic constraints of an integer program can be formulated on the basis of the system model constructed with these components. A recipe containing the detailed operation steps can then be generated by solving this mathematical programming model. The effectiveness of the proposed approach is clearly demonstrated with the realistic example given at the end of this paper.

REFERENCES

Årzén, K.-E., and Johnsson, C., 1996, "Object-Oriented SFC and ISA-S88.01 Recipes," *ISA Transactions*, Vol. 35, pp. 237-244.

Chou, H. H., and Chang, C. T., 2005, "A Petri-Net Based Strategy to Synthesize the Operating Procedures for Cleaning Pipeline Networks," *Industrial and Engineering Chemistry Research*, Vol. 44, pp. 114-123.

David, R., and Alla, H., 1994, "Petri Nets for Modeling of Dynamic Systems - a Survey," *Automatica*, Vol. 30, pp. 175-202.

Foulkes, N. R., Walton, M. J., Andow, P. K., and Galluzzo, M., 1988, "Computer-Aided Synthesis of Complex Pump and Valve Operations," *Computers and Chemical Engineering*, Vol. 12, pp. 1035-1044.

ISA, 1995, *S88.01 Batch Control, Part 1: Models and Terminology*, ISA, Research Triangle Park, NC, USA.

Karassik, I. J., and McGuire, J. T., 1998, *Centrifugal Pumps*, 2nd ed., Chapman & Hall: New York, USA, pp. 885-887.

O'Shima, E., 1978, "Safety Supervision of Valve Operation," *Journal of Chemical Engineering of* Japan, Vol. 11, pp. 390-395.

Peterson, J. L., 1981, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Rivas, J. R., and Rudd, D. F., 1974, "Synthesis of Failure-Safe Operations," *AIChE Journal*, Vol. 20, pp. 320-325.

Uthgenannt, J. A., 1996, "Path and Equipment Allocation for Multiple, Concurrent Pro-Cesses on Networked Process Plant Units," *Computers and Chemical Engineering*, Vol. 20, pp. 1081-1087.

Wang, Y. F., and Chang, C. T., 2004,"A Petri-Net Based Deductive Reasoning Strategy for Fault Identification in Batch Processes," *Industrial and Engineering Chemistry Research*, Vol. 43, pp. 2704-2720.

Wang, Y. F., Chou, H. H., and Chang, C. T., 2005, "Generation of Batch Operating Procedures for Multiple Material-Transfer Tasks with Petri Nets, " *Computers and Chemical Engineering*, Vol. 29, pp. 1822-1836.

Manuscript Received: Dec. 05, 2005
and Accepted: Dec. 06, 2005