Development of an Optimal Sensor Placement Procedure Based on Fault Evolution Sequences

Jung Yang Chen and Chuei-Tin Chang*

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

The diagnostic resolution issues are usually addressed with a quantitative or semiquantitative approach in the reported studies concerning sensor placement procedures. Since the required information for implementing the traditional methods may not be always available, a SDG-based strategy is proposed in this paper to design the sensor networks on the basis of qualitatively predicted fault evolution sequences. To achieve a maximum level of resolution and, at the same time, to ensure observability, the corresponding design problems are formulated as integer programs in this work. Two alternative strategies are also developed for producing the optimal solutions. The feasibility of the proposed method is demonstrated with three examples.

1. Introduction

Any fault diagnosis method must be implemented on the basis of sufficient online measurement data. In realistic applications, it is obvious that only a limited number of sensors can be installed due to budget constraints. Since improper selections may seriously hamper diagnostic performance, the development of sensor placement strategy has become an important research issue in recent years. Ideally, a sensor network should be configured to facilitate detection of all possible fault origins and also to maximize overall diagnostic resolution under a given cost limit. In fact, numerous related contributions have already been reported in the literature, e.g., Madron and Veverka,[1] Maquin et al.,[2] Ali and Narasimhan,[3–5] Ricker and Lee,[6,7] Lambert,[8] Sen et al.,[9] Bagajewicz and Sánchez,[10,11] Bhushan and Rengaswamy,[12–14] Bagajewicz and Carbrera,[15] Bagajewicz and Fuxman,[16] and Kotecha et al.[17]

Clearly the sensor network should be devised to best serve a chosen fault diagnosis strategy. The available diagnosis methods could generally be divided into three distinct categories,[18,19] i.e., the model-based approaches, the knowledge-based approaches, and the data-analysis-based approaches. The focus of the present study is concerned with the digraph-based techniques. In essence, the signed directed graphs (SDGs) are *qualitative* models that can be used to characterize the causal relations among faults, failures and their effects.[18–21] The advantage of this modeling approach is mainly due to the fact that these relations can almost always be established on the basis of simple engineering principles. On the other hand, the more rigorous mathematical models and the more case-specific knowledge bases are required to be built according to the measurement data and operational experiences obtained in the course of every possible accident. These needs are often not satisfiable.

To address the observability issue, Raghuraj and Rengaswamy[22] proposed a digraph(DG)-based design procedure to obtain a set of measured variables that could be influenced by the given fault origins. In principle, their approach was to select the sensors by identifying the directed paths emanating from every root node in the *unsigned* DG model and then collecting all the nodes on these paths. They also treated each pair of root causes as a pseudo fault origin. Additional set of sensors were then obtained to improve diagnostic resolution. A greedy search algorithm and an extended backtracking algorithm were developed in their study to solve the set covering problem in an optimal fashion. In a later work, Bhushan and Rengaswamy[12] used the additional information embedded in the *signed* digraph (SDG) for sensor placement so as to further enhance resolution. Although better performance can be achieved in simple systems, their SDG-based diagnosis approach is still unreliable for more realistic applications, e.g., the Tennessee−Eastman process.[6,7,23,24] Bhushan and Rengaswamy[13,14] formulated a slightly different sensor network design problem. In particular, they characterized the observability problem with a bipartite matrix and the corresponding mixed-integer linear programming (MILP) model. The objective function of this model was either capital cost or reliability. In addition, the edge gains in their SDGs were allowed to be nonintegers for the purpose of producing more informative semiquantitative solutions. Bagajewicz and Sánchez[25] and Bagajewicz and Fuxman[16] adopted a similar formulation to optimize reliability and/or gross-error robustness. Finally, Kotecha et al.[17] used a constrained programming (CP) approach to place the sensors in order to minimize unobservability in the presence of uncertainties.

It should be noted that, since the SDG models are basically static in nature, the existing fault identification techniques were developed mostly on the basis of steady-state symptoms, e.g., Rengaswamy et al.[20] In order to circumvent this drawback, Maurya et al.[26,27] has proposed a systematic framework to build appropriate digraph models and to determine the initial and final system responses to any given fault origin. Although their approach is quite effective, it fails to address issues concerning the *precedence order* (in time) between the fault propagation effects implied by every input−output connection in the digraph model. Thus, the SDG-based diagnosis methods and the aforementioned sensor placement procedures may be further improved by incorporating these additional insights. To this end, a series of studies have already been carried out to construct fuzzy inference systems according to a *given* set of online sensors.[28–30] In the present work, the concept of precedence order is extended to design sensor networks so as to maximize diagnostic resolution and also to ensure observability.

The rest of this paper is organized as follows. A brief review of the qualitative simulation procedure is first presented in the next section to facilitate illustration of the proposed sensor placement strategy. In this study, a bipartite matrix is used to record the diagnostic functions of all possible sensor pairs. Its

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

10.1021/ie701767v CCC: $40.75
Published on Web 08/27/2008

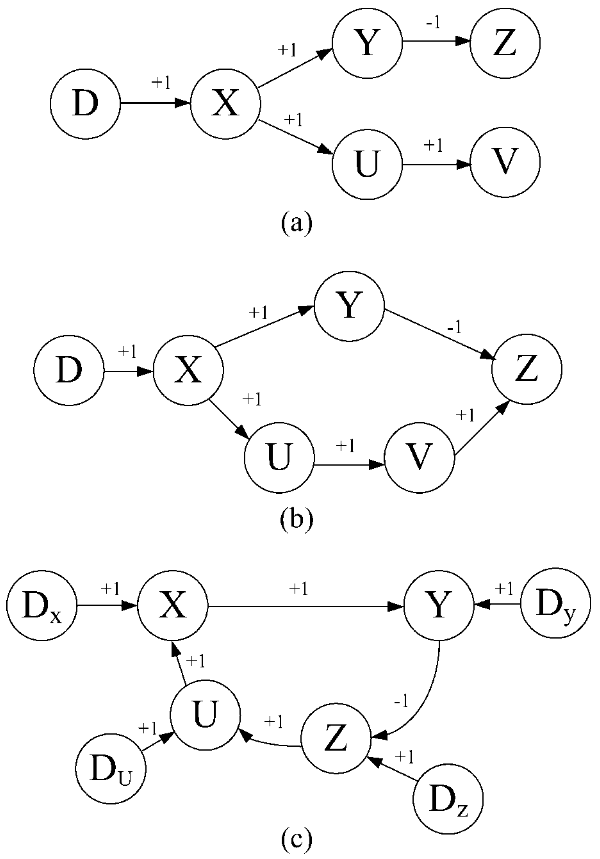

Figure 1. (a) A tree-shaped SDG; (b) a negative feed forward loop (NFFL); (c) a negative feedback loop (NFBL).

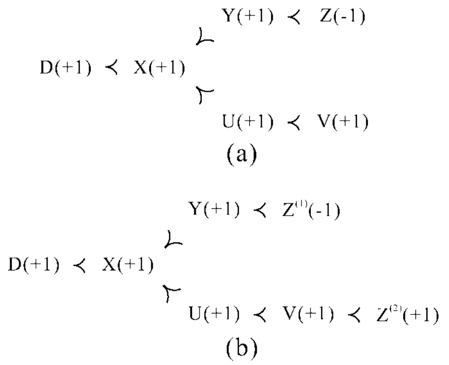

Figure 2. (a) FPP of the scenario resulting from $D(+1)$ in Figure 1(a); (b) FPP of the scenario resulting from $D(+1)$ in Figure 1b.

construction procedure is described in section 3. The sensor network design problems can then be formulated according to this matrix as integer programs. The mathematical programming models and their solution strategies are provided in section 4. Three examples are given in section 5 to demonstrate the feasibility of the proposed approach, and a short conclusion follows in the final section.

2. Qualitative Prediction of Fault Propagation Behavior

By definition, an accident is an unplanned rarely occurring event or a sequence of such events. Some of the catastrophic accidents may not be experienced even in a long-existing plant. Thus, in any realistic system, it is obviously infeasible to collect and analyze the historical data of *all* possible scenarios. As a result, there is a need to *predict* the fault propagation behaviors with qualitative simulation techniques. Although this simulation procedure can be found in the literature,[28–31] a brief review is still presented in the sequel to facilitate illustration of the proposed sensor placement strategy.

Table 1. Steady-State Values of Loop Variables in the Control NFBL Shown in Figure 1c

fault origin	X (controlled variable)	Y (sensor output)	Z (controller output)	U (mainpulated variable)
$D_X(+1)$	0	0	−1	−1
$D_Y(+1)$	−1	0	−1	−1
$D_Z(+1)$	0	0	0	0
$D_U(+1)$	0	0	−1	0

Table 2. Bipartite Matrix of the System in Figure 1c with Fault Origins $D_X(+1)$ and $D_Y(+1)$

function	{1,1}	{2,2}	{3,3}	{4,4}	{1,2}	{1,3}	{1,4}	{2,3}	{2,4}	{3,4}
(1,1)	1	1	1	1	1	1	1	1	1	1
(2,2)	1	1	1	1	1	1	1	1	1	1
(1,2)	1	0	0	0	1	1	1	0	0	0

Table 3. A Fictitious Bipartite Matrix

function	{1,1}	{2,2}	{3,3}	{4,4}	{1,2}	{1,3}	{1,4}	{2,3}	{2,4}	{3,4}
(1,1)	0	1	0	1	1	0	1	1	1	1
(2,2)	1	0	0	0	1	1	1	1	1	0
(3,3)	0	0	1	0	0	1	0	1	1	1
(1,2)	0	0	1	0	0	1	0	1	0	1
(1,3)	1	1	1	0	1	1	1	1	1	1
(2,3)	0	0	0	0	0	0	1	0	0	0

Table 4. Current Matrix Obtained after One Iteration in the Greedy Search Procedure

functions	{1,1}	{4,4}	{1,2}	{1,3}	{1,4}	{2,4}	{3,4}
(2,3)	0	0	0	0	1	0	0

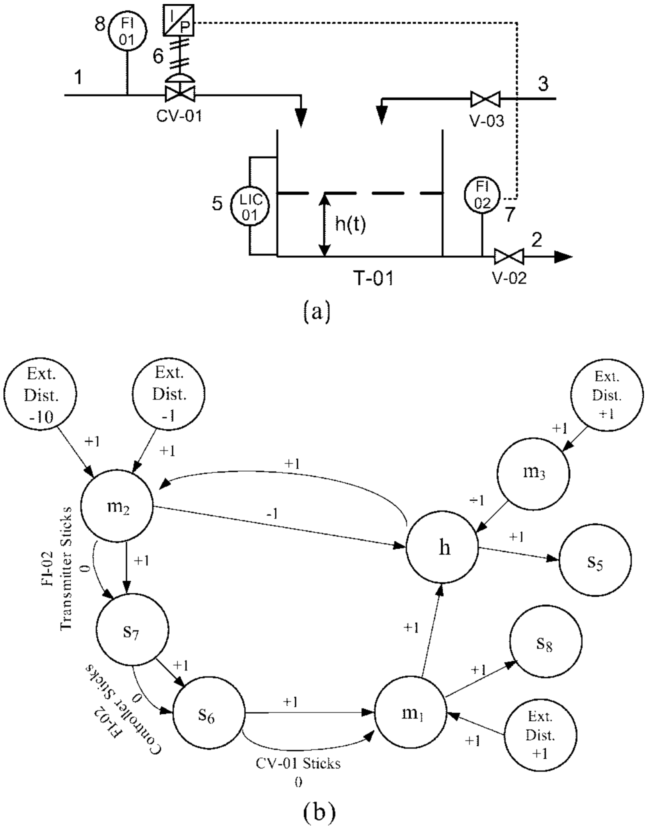

Figure 3. (a) Flow diagram of a single-tank storage system with feed-forward level-control loop; (b) the SDG model of single-tank storage system.

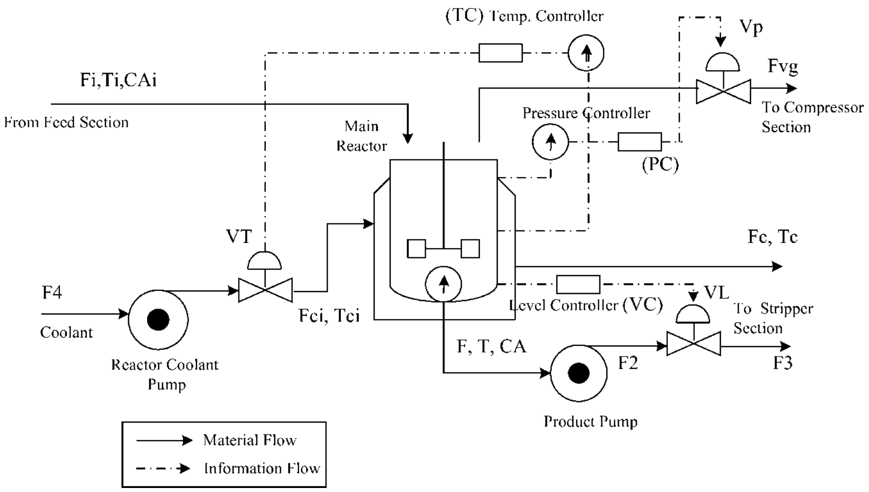

Figure 4. Flow diagram of a CSTR reactor with pressure, level, and temperature control loops.

2.1. Simulation Approach. The effects of a fault/failure in a process system can be determined according to its SDG model. The fault origins are usually associated with the primal nodes, i.e., nodes with no inputs. A set of three values, i.e., {−1,0,+1}, may be assigned to each node to *qualitatively* represent deviation levels from the normal value of corresponding variable. The value 0 means that it is at the normal steady state. The negative values are used to denote the lower-than-normal states and the positive ones signify the opposite. Notice that the causal relation between the deviations in two variables can be characterized with a directed arc and the corresponding gain. Again, each gain may assume one of the aforementioned qualitative values, i.e., 0 and ±1.

If a system can be modeled by tree-shaped SDG, the deviation values of all variables affected by a given fault origin can always be evaluated. In particular, the output value of any arc can be computed with the gain and its input value according to the following formula

$$v_{\text{out}} = g \times v_{\text{in}} \tag{1}$$

where g, v_{in}, and v_{out} denote respectively the gain, input, and output values. However, notice that the time at which each deviation occurs is really indeterminable. Without the reference of time in the SDG-based simulation results, it can nonetheless be safely assumed that *the change in an input variable should always occur earlier than those in its outputs*. This is the basic

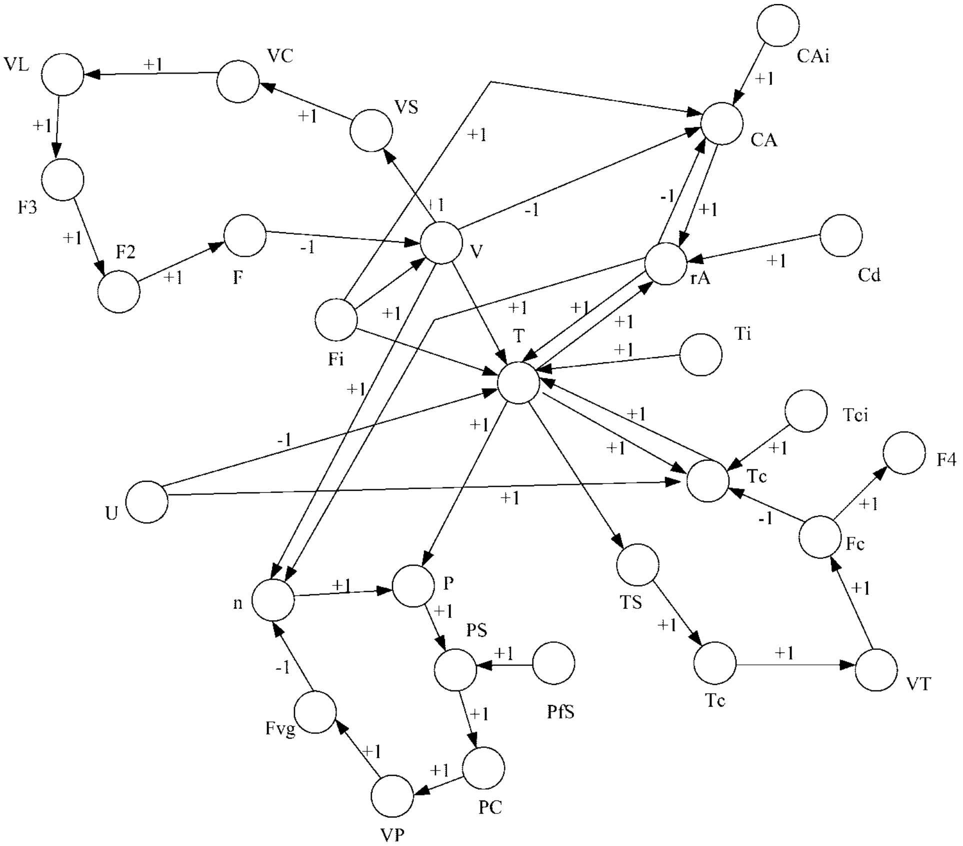

Figure 5. SDG model of the CSTR reactor system.

Table 5. Candidate Measurements in Example 2

no.	possible measurements
1	VL
2	F_3
3	F_2
4	F
5	V
6	VS
7	VC
8	CA
9	rA
10	T
11	T_c
12	F_4
13	Fc
14	VT
15	TC
16	TS
17	n
18	P
19	PS
20	PC
21	VP
22	F_{vg}

Table 6. Fault Origins in Example 2

no.	possible fault origins
1	$CA_i(+1)$
2	$CA_i(-1)$
3	$T_i(+1)$
4	$T_i(-1)$
5	$F_i(+1)$
6	$F_i(-1)$
7	$T_{ci}(+1)$
8	$T_{ci}(-1)$
9	$C_d(-1)$
10	$U(-1)$

Table 7. Candidate Measurements in Example 3

no.	possible measurements	no.	possible measurements	no.	possible measurements
1	P_r	22	$y_{A,8}$	43	$x_{G,11}$
2	P_s	23	$y_{B,8}$	44	$x_{H,11}$
3	P_m	24	$y_{C,8}$	45	$x_{D,r}$
4	F_6	25	$y_{D,8}$	46	$x_{E,r}$
5	$y_{A,6}$	26	$y_{E,8}$	47	$x_{F,r}$
6	$y_{B,6}$	27	$y_{F,8}$	48	$x_{G,r}$
7	$y_{C,6}$	28	$y_{G,8}$	49	$x_{H,r}$
8	$y_{D,6}$	29	$y_{H,8}$	50	$V_{L,r}{}^{s}$
9	$y_{E,6}$	30	$y_{A,9}$	51	$C_{L,r}$
10	$y_{F,6}$	31	$y_{B,9}$	52	CV_{Lr}
11	$y_{G,6}$	32	$y_{C,9}$	53	$V_{L,s}{}^{s}$
12	$y_{H,6}$	33	$y_{D,9}$	54	$C_{L,s}$
13	F_7	34	$y_{E,9}$	55	CV_{Ls}
14	$y_{A,7}$	35	$y_{F,9}$	56	$V_{L,p}{}^{s}$
15	$y_{B,7}$	36	$y_{G,9}$	57	$C_{L,p}$
16	$y_{C,7}$	37	$y_{H,9}$	58	CV_{Lp}
17	$y_{D,7}$	38	$x_{D,10}$	59	F_{10}
18	$y_{E,7}$	39	$x_{E,10}$	60	F_{11}
19	$y_{F,7}$	40	$x_{F,10}$	61	T_s
20	$y_{G,7}$	41	$x_{G,10}$		
21	$y_{H,7}$	42	$x_{H,10}$		

Table 8. Fault Origins in Example 3

no.	possible fault origins	description
1	$F_2(+1)$	a positive disturbance in the flow rate of stream 2
2	$F_4(+1)$	a positive disturbance in the flow rate of stream 4
3	$F_8(+1)$	a positive disturbance in the flow rate of stream 8
4	$F_9(+1)$	a positive disturbance in the flow rate of stream 9
5	$C_d(-1)$	catalyst deactivation
6	$F_2(-1)$	a negative disturbance in the flow rate of stream 2
7	$F_4(-1)$	a negative disturbance in the flow rate of stream 4
8	$F_8(-1)$	a negative disturbance in the flow rate of stream 8
9	$F_9(-1)$	a negative disturbance in the flow rate of stream 9

Table 9. Observable Event Pairs in Example 3

	fault	(P_r,F_6)	(F_6,P_r)	(P_r,P_r)	(F_6,F_6)
1	$F_2(+1)$		$(+,+)$	$(+,\pm)$	$(+,\pm)$
2	$F_4(+1)$		$(+,+)$	$(+,\pm)$	$(+,\pm)$
3	$F_8(+1)$	$(-,+)$		$(+,\pm)$	$(-,\pm)$
4	$F_9(+1)$	$(-,+)$		$(+,\pm)$	$(-,\pm)$
5	$C_d(-1)$	$(+,-)$	$(-,+)$	$(-,\pm)$	$(+,\pm)$
6	$F_2(-1)$		$(-,-)$	$(-,\pm)$	$(-,\pm)$
7	$F_4(-1)$		$(-,-)$	$(-,\pm)$	$(-,\pm)$
8	$F_8(-1)$	$(+,-)$		$(-,\pm)$	$(+,\pm)$
9	$F_9(-1)$	$(+,-)$		$(-,\pm)$	$(+,\pm)$

assumption adopted in the present study to interpret the simulation results.

2.2. Fault Propagation Paths. Due to the unique information structure generated with the above approach, a special representation has been designed to characterize the predicted fault propagation behaviors. This representation is referred to as the *fault propagation path* (FPP). Let us first consider the simple tree-shaped SDG given in Figure 1a. The fault propagation path associated with a positive disturbance $D(+1)$ is presented in Figure 2a. Notice that the path structure is identical to the corresponding SDG in Figure 1a. Each node here represents a previously nonexistent fault effect. Every effect is specified with a qualitative value $+1$ or -1, which can be computed according to eq 1. The precedence order of two different effects is specified by connecting them with the "precedes" symbol, i.e., the effect on its left should occur earlier than that on the right. The sequence of conditions on the same branch in a FPP should be interpreted as the order of occurrence (in time) of different effects resulting from the given fault origin. However, the order of two distinct events located on two *separate* branches should be considered as indeterminable.

Other than the tree-shaped structure, various feed forward loops (FFLs) and feedback loops (FBLs) can often be identified in realistic digraph models.[30] These loops may cause problems in performing simulation. Additional modifications have thus been introduced into the FPP representations. These changes are summarized below:

Feed forward loops: For illustration purpose, let us consider the fictitious SDG in Figure 1b as an example. The feed forward loop in this case contains two paths, i.e., (1) $X \rightarrow Y \rightarrow Z$, and (2) $X \rightarrow U \rightarrow V \rightarrow Z$. Notice that the products of the edge gains along these two paths can be found to be -1 and $+1$, respectively. Consequently, this FFL is also referred to as a *negative* feed forward loop (NFFL). It is assumed that the effects of fault origin $D(+1)$ propagate along separate paths independently. In this case, the FPP corresponding to a positive disturbance in D also takes the form of a tree, i.e., Figure 2b. The symbols $Z^{(1)}(-1)$ and $Z^{(2)}(+1)$ are used here to denote the changes in variable Z caused by disturbances propagating along paths (1) and (2), respectively. Since they represent two separate effects on the same variable, a computation procedure is needed to evaluate their net effects at various instances. This reconciliation method is presented elsewhere.[30]

Feedback loops: Let us consider the feedback loop in Figure 1c as an example. Since the product of all edge gains on this loop is negative, it is referred to as a *negative* feedback loop (NFBL) in this paper. It is in general very difficult to accurately simulate the dynamic behavior of a NFBL on the basis of SDG model alone. To illustrate this point, let us consider the effects of disturbance $D_X(+1)$ on the example system. Obviously, the incipient responses can be determined according to eq 1, i.e.,

$$D_X(+1) \prec X(+1) \prec Y(+1) \prec Z(-1) \prec U(-1) \prec \cdots$$

However, since the net effect of two simultaneous inputs, i.e., $D_X(+1)$ and $U(-1)$, on variable X is uncertain afterward, the event sequence following this initial FPP is really indeterminable without further quantitative and/or qualitative knowledge of the physical system in question. Thus, the complete fault propagation process in a NFBL is often described on a case-by-case basis.[30] However, if the NFBL in Figure 1c is a *control* loop, then the final steady-state values of the loop variables can be assigned according to the general guidelines suggested by Ju et al.[32] Table 1 is a listing of the final states of control NFBLs in various scenarios. Thus, the FPP corresponding to $D_X(+1)$ in this situation can be expressed as

$$D_X(-1) \prec X(+1) \prec Y(-1) \prec Z(-1) \prec U(-1) \prec \cdots \prec [X(0), Y(0), Z(-1), U(-1)]_f \tag{2}$$

It can be observed that the final states of all loop variables in this FPP are lumped into a single ending node in a square bracket and their precedence order is left unspecified. This is due to the difficulties in verifying the occurrence order of these eventual symptoms in real time.

2.3. Event Strings. The use of FPP representation essentially implies that the number of *occurred* events is assumed in this study to be increased one at a time in a fault propagation scenario. To be specific, let us again consider the digraph in Figure 1a. The precedence order of newly occurred events in this simple system can be determined by inspection. If fact, there are six possible sequences, i.e.,

$$X(+1) \prec U(-1) \prec V(+1) \prec Y(+1) \prec Z(-1) \tag{3}$$

$$X(+1) \prec U(-1) \prec Y(+1) \prec V(+1) \prec Z(-1) \tag{4}$$

$$X(+1) \prec U(-1) \prec Y(+1) \prec Z(-1) \prec V(+1) \tag{5}$$

$$X(+1) \prec Y(+1) \prec U(+1) \prec V(+1) \prec Z(-1) \tag{6}$$

$$X(+1) \prec Y(+1) \prec U(+1) \prec Z(-1) \prec V(+1) \tag{7}$$

$$X(+1) \prec Y(+1) \prec Z(-1) \prec U(+1) \prec V(+1) \tag{8}$$

These sequences will be referred to as the *event strings* in this paper and, for convenience, the precedence symbols will all be dropped later. It should be noted that these strings can be encoded into IF-THEN rules in a two-layer fuzzy inference system to enhance the diagnostic performance. The detailed implementation procedure can be found in Chang et al.,[31] Chang and Chang,[28] and Chen and Chang.[29,30]

3. Diagnostic Functions of Sensor Pairs

3.1. Ordered Event Pairs. In order to identify a specific fault origin, it is clearly necessary to confirm at least one of the effects listed in the corresponding event strings with online sensors. For example, the candidate measurements for detecting any fault origin in Figure 1c should be those for monitoring the loop variables, i.e., X, Y, Z, and U. However, if the sensors are placed independently according to this criterion *alone*, the resulting system may not be capable of capturing all the key features of event strings. To fix idea, let us consider a *positive* feedback loop (PFBL) represented by the structure given in Figure 1c, i.e., the gain between Y and Z is changed from -1 to $+1$ in this case. If only two of the loop variables are allowed to be measured online, then selection of the sensor set $\{X,Y\}$ (or $\{Z,U\}$) will fail to distinguish the origins $D_X(+1)$ and $D_Z(+1)$. On the other hand, any of the remaining combinations, i.e., $\{X,Z\}$, $\{X,U\}$, $\{Y,Z\}$, and $\{Y,U\}$, can be adopted to produce the correct diagnosis. This is due to the fact that the precedence orders of the measured effects in these scenarios are different for the two fault origins under consideration.

Notice that an event string can be uniquely described with all possible ordered pairs of its elements. As an example, let us consider eq 3. A list of the ordered event pairs in this string can easily enumerated, i.e.,

$$X(+1)U(+1),\ X(+1)V(+1),\ X(+1)Y(+1),\ X(+1)Z(-1),\ U(+1)V(+1),\ U(+1)Y(+1),\ U(+1)Z(-1),\ V(+1)Y(+1),\ V(+1)Z(-1),\ Y(+1)Z(-1) \tag{9}$$

Since the event strings (3)−(8) are derived from the same FPP, their ordered event pairs should all be considered as possible symptoms of the same fault origin. Thus, other than the ordered pairs in eq 9, additional ones can be identified from strings (4)−(8), i.e.,

$$Y(+1)V(+1),\ Z(-1)V(+1),\ Y(+1)U(+1),\ Z(-1)U(+1) \tag{10}$$

As mentioned before, the incipient and eventual effects of a particular fault origin on the process variables in a system with NFBLs and/or NFFLs may not be the same. They are usually the ones associated with either the nodes on a feedback loop (and their outputs) or with the ending node of a feed forward loop (and its outputs). Thus, it is logical to consider the ordered event pairs associated with the same variable as fault symptoms also. For example, let us assume that the net effect of $Z^{(1)}(-1)$ and $Z^{(2)}(+1)$ in Figure 1b can be determined to be $Z(-1)$. Consequently, these pairs can be written as

$$X(+1)X(+1),\ Y(+1)Y(+1),\ Z(-1)Z(-1),\ Z(+1)Z(-1),\ U(+1)U(+1),\ V(+1)V(+1) \tag{11}$$

As another example, let us assume that the NFBL structure in the Figure 1c can be adopted to model a process control system. In response to disturbance $D_X(+1)$, the controlled variable X should initially deviate from its normal state in the positive direction and then return to the set point within a finite time period. Thus, the single-variable dual-valued event pair $X(+1)X(0)$ can be regarded as a possible fault signature. Finally, in any system that can be modeled with a treelike digraph, it can be determined with the proposed qualitative simulation procedure that the initial and final deviations of a fault-affected variable are the same. In this work, this potential symptom is written as a single-variable single-valued event pair, e.g., these pairs for the scenario presented in Figure 1a are given below:

$$X(+1)X(+1),\ Y(+1)Y(+1),\ Z(-1)Z(-1),\ U(+1)U(+1),\ V(+1)V(+1) \tag{12}$$

Since the string number grows exponentially with the node number in the FPP, enumeration of the ordered pairs may become intractable in large systems. A graph−theoretic computer algorithm has thus been developed to perform this operation efficiently. The computation procedure is summarized in Appendix A.

3.2. Symptom Sets, Sensor Sets, and Function Sets. In this work, the online sensors are chosen on the basis of the ordered pairs identified from FPP. Let us first construct a *symptom set* $\mathscr{A}_i$ in which all possible symptoms implied by the event strings of fault origin i are collected. Specifically

$$\mathscr{A}_{\rm i} = \mathscr{A}_{\rm i}^{(1)} \cup \mathscr{A}_{\rm i}^{(2)} \tag{13}$$

where $\mathscr{A}_i^{(1)}$ and $\mathscr{A}_i^{(2)}$ denote respectively the one- and two-variable symptom sets of fault origin i. For example, the symptoms in set $\mathscr{A}_i^{(1)}$ for the scenarios described by strings

(3)−(8) should be those given in (12), while those in $\mathcal{A}_i^{(2)}$ are listed in (9) and (10). Similarly, this symptom set for the scenario described in Figure 1b is given below:

$$\left\{\begin{matrix} X(+1)X(+1), & Y(+1)Y(+1), & Z(\pm1)Z(-1), & U(+1)U(+1), \\ V(+1)V(+1), & X(+1)V(+1), & X(+1)Y(+1), & X(+1)Z(\pm1), \\ X(+1)U(+1), & Y(+1)Z(\pm1), & Y(+1)U(+1), & Y(+1)V(+1), \\ Z(-1)U(+1), & Z(-1)V(+1), & Z(+1)Y(+1), & U(+1)Z(\pm1), \\ U(+1)V(+1), & U(+1)Y(+1), & V(+1)Y(+1), & V(+1)Z(\pm1) \end{matrix}\right\} \tag{14}$$

and, for disturbance $D_X(+1)$ in Figure 1(c), set $\mathcal{A}_i$ is

$$\left\{\begin{matrix} X(+1)X(0), & Y(+1)Y(0), & Z(-1)Z(-1), & U(-1)U(-1), \\ X(+1)Y(+1), & X(+1)Z(-1), & X(+1)U(-1), & Y(+1)Z(-1), \\ & Y(+1)U(-1), & Z(-1)U(-1) & \end{matrix}\right\} \tag{15}$$

Confirmation of the event(s) in $\mathcal{A}_i$ is clearly needed to ensure *observability*. On the other hand, the resolution issues are addressed in this study with additional symptom sets. In particular

$$\mathcal{B}_{ij} = \mathcal{A}_i \cup \mathcal{A}_j - \mathcal{A}_i \cap \mathcal{A}_j \tag{16}$$

where $i \neq j$. If the set $\mathcal{B}_{ij}$ is nonempty, it is clear that appropriate sensors can be selected to observe the effects of either fault i or fault j (but not both).

Let us consider disturbances $D_X(+1)$ (fault origin 1) and $D_Y(+1)$ (fault origin 2) in Figure 1c as an example to illustrate the above ideas. Notice that the symptom set of the first origin, i.e., $\mathcal{A}_1$, is given in (15) and

$$\mathcal{A}_2 = \left\{\begin{matrix} X(+1)X(-1), & Y(+1)Y(0), & Z(-1)Z(-1), & U(-1)U(-1), \\ Y(+1)Z(-1), & Y(+1)U(-1), & Y(+1)X(-1), & Z(-1)U(-1), \\ & Z(-1)X(-1), & U(-1)X(-1) & \end{matrix}\right\} \tag{17}$$

Consequently, $\mathcal{B}_{12}$ can be obtained according to equation (16), i.e.

$$\mathcal{B}_{12} = \left\{\begin{matrix} X(+1)X(0), & X(-1)X(-1), & X(+1)Y(+1), & X(+1)Z(-1), \\ X(+1)U(-1), & Y(+1)X(-1), & Z(-1)X(-1), & U(-1)X(-1) \end{matrix}\right\} \tag{18}$$

Let us next define a set operation $\mathcal{O}(\bullet)$ to be that of *dropping the qualitative value of every event in a symptom set and then replacing the resulting symbols with numerical labels*. The symptom sets $\mathcal{A}_i$ and $\mathcal{B}_{ij}$ can be converted to the corresponding *sensor sets* $\mathcal{A}_i^S$ and $\mathcal{B}_{ij}^S$ with this operation. Specifically, $\mathcal{A}_i^S = \mathcal{O}(\mathcal{A}_i)$ and $\mathcal{B}_{ij}^S = \mathcal{O}(\mathcal{B}_{ij}) - \mathcal{O}(\mathcal{A}_i \cap \mathcal{A}_j)$. Notice that additional sensor pairs are excluded from the set $\mathcal{B}_{ij}^S$, i.e., those in $\mathcal{O}(\mathcal{A}_i \cap \mathcal{A}_j)$. This is due to the fact that the sensor pairs required to monitor the ordered event pairs in $\mathcal{A}_i \cap \mathcal{A}_j$ may also appear in $\mathcal{O}(\mathcal{B}_{ij})$. Thus, in order to clearly distinguish the two fault origins in question, these repeated sensors pairs must be removed. For the example mentioned above, one can obtain

$$\mathcal{A}_1^S = \left\{\begin{matrix} \{1,1\} & \{2,2\} & \{3,3\} & \{4,4\} & \{1,2\} \\ \{1,3\} & \{1,4\} & \{2,3\} & \{2,4\} & \{3,4\} \end{matrix}\right\} \tag{19}$$

$$\mathcal{A}_2^S = \left\{\begin{matrix} \{1,1\} & \{2,2\} & \{3,3\} & \{4,4\} & \{2,3\} \\ \{2,4\} & \{2,1\} & \{3,4\} & \{3,1\} & \{4,1\} \end{matrix}\right\} \tag{20}$$

$$\mathcal{B}_{12}^S = \{\{1,1\} \quad \{1,2\}\{1,3\} \quad \{1,4\}\} \tag{21}$$

where X, Y, Z, and U are labeled with 1, 2, 3, and 4, respectively. Notice that all sensor pairs are unordered.

Finally, a *function set* $\mathcal{C}_{mn}$ can be defined for every possible sensor pair $\{m,n\}$. Specifically, the elements in this set represent the diagnostic functions that can be performed by measuring variables m and n. For example, the function sets of the sensor pairs {1,2}, {1,1} and {2,2} can be extracted from eqs 19–21:

$$\mathcal{C}_{12} = \mathcal{C}_{21} = \{(1,1), (2,2), (1,2)\} \tag{22}$$

$$\mathcal{C}_{11} = \{(1,1), (2,2), (1,2)\} \tag{23}$$

$$\mathcal{C}_{22} = \{(1,1), (2,2)\} \tag{24}$$

where the elements with repeating indices are used to denote the observable fault origins and those with distinct indices represent the distinguishable origin pairs.

3.3. Bipartite Matrix. For computational convenience, a bipartite matrix[13,14,33] is constructed in this work to characterize the mapping between sensor pairs and diagnostic functions according to the sensor sets $\mathcal{A}_i^S$s and $\mathcal{B}_{ij}^S$s (or the function sets $\mathcal{C}_{mn}$s). The columns of this matrix (say $\mathbf{D}$) are associated with candidate sensor pairs and the rows are corresponding to the diagnostic functions. In particular

$$\mathbf{D} = (d_{lk})_{(F+FF)\times(S+SS)} \tag{25}$$

and

$$d_{lk} = \begin{cases} 1 & \text{if sensor pair } k \text{ is capable of performing} \\ & \qquad\qquad \text{diagnostic function } l \\ 0 & \text{otherwise} \end{cases}$$

where S represents the total number of measurable variables;

$$SS = \binom{S}{2}$$

is the number of all possible sensor pairs; F denotes the total number of fault origins;

$$FF = \binom{F}{2}$$

is the number of all possible origin pairs. For example, the bipartite matrix in Table 2 can be constructed on the basis of the sensor sets in eqs 19–21 or the corresponding function sets. It should also be noted that the columns in this bipartite matrix are arranged in a specific order according to the following rules:

(1) The first S columns are associated with the elements in the function sets $\mathcal{C}_{k_1k_1}$s, where $k_1 = 1,2,..., S$. Each column number is the same as the sensor label k_1 of the corresponding set.

(2) The remaining SS columns are associated with the function sets $\mathcal{C}_{k'_1k''_1}$s, where $k'_1 < k''_1$ and k'_1, $k''_1 = 1,2,..., S$. Every column number k_2 can be related to the sensor labels of the corresponding function set, i.e., k'_1 and k''_1, according to the following equation

$$k_2 = k''_1 + k'_1 S - \frac{k'_1(k'_1 + 1)}{2} \tag{26}$$

where $S + 1 \leq k_2 \leq SS$.

4. Optimal Sensor Placement Problem

4.1. Problem Formulation. The sensor selection problem can be formulated as two alternative *linear* integer programs. Let us use a binary vector to reflect all possible sensor pair selections. Specifically

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix} \tag{27}$$

where $\mathbf{x}^{(1)} = [x_1, x_2,..., x_S]^T$ and $\mathbf{x}^{(2)} = [x_{S+1}, x_{S+2},..., x_{SS}]^T$. Notice that the former vector represents the selections corresponding to the first S columns of the bipartite matrix, while the latter denotes the decisions concerning the remaining sensor pairs. A second binary vector $\mathbf{y}$ is also introduced in these programs to denote the distinguishable origin pairs, i.e.,

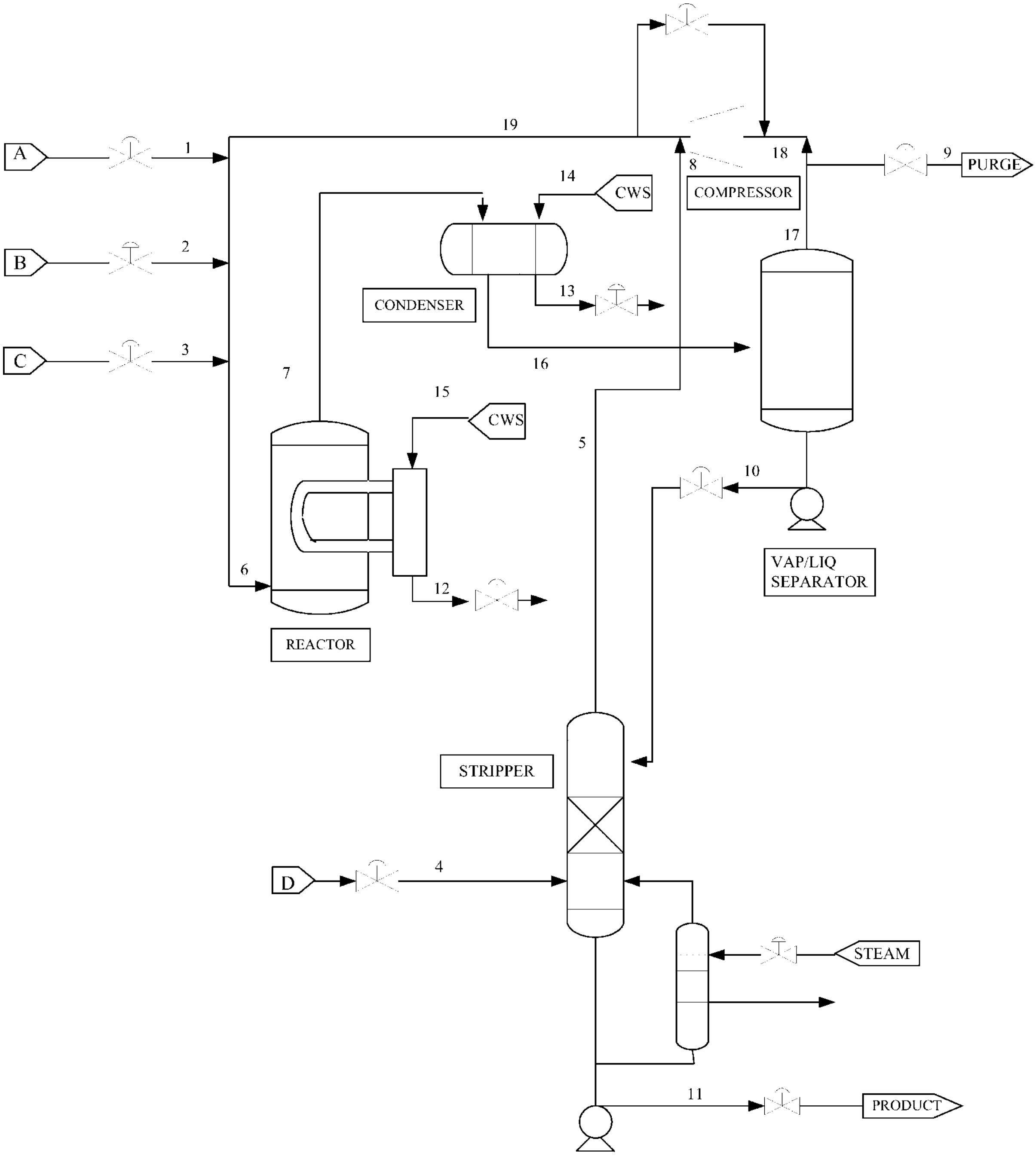

Figure 6. Flow diagram of Tennessee−Eastman process.

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \end{bmatrix} \tag{28}$$

where $\mathbf{y}^{(1)} = [y_1, y_2, ..., y_F]^T$ and $\mathbf{y}^{(2)} = [y_{F+1}, y_{F+2}, ..., y_{FF}]^T$.

The first integer program (IP_1) can thus be written as

$$\min_{\mathbf{x},\mathbf{y}} \sum_{K_1=1}^{S} x_{k_1} \tag{29}$$

subject to

$$\sum_{k=1}^{S+SS} d_{lk} x_k \geq y_l \tag{30}$$

where $l = 1, 2, ..., F + FF$. Since the observability of every fault origin is considered to be an essential requirement of the fault diagnosis system in this study, the following equality constraint is included in the integer program:

$$\sum_{l_1=1}^{F} y_{l_1} = F \tag{31}$$

Notice this is the same as $y_1 = y_2 = \cdots = y_F = 1$. On the other hand, various inequality constraints can be imposed on the remaining binary variables in $\mathbf{y}^{(2)}$ to address the resolution issues on a case-by-case basis. For example, a lower resolution limit can be enforced with

$$\sum_{l_2=F+1}^{FF} y_{l_2} \geq L_y \tag{32}$$

where L_y is a user-specified number. In addition, the following logic constraints must be imposed to ensure consistency among sensor selections:

$$(1 - x_{k'_1}) + (1 - x_{k''_1}) + x_{k_2} \geq 1 \tag{33}$$

$$(1 - x_{k_2}) + x_{k'_1} \geq 1 \tag{34}$$

$$(1 - x_{k_2}) + x_{k''_1} \geq 1 \tag{35}$$

where $1 \leq k'_1, k''_1 \leq S$, $S + 1 \leq k_2 \leq SS$ and the constraint in eq 26 should also be included in the mathematical programming model.

Notice that a second integer program (IP_2) can be constructed by adopting an alternative objective function, i.e.

$$\max_{\mathbf{x},\mathbf{y}} \sum_{l=F+1}^{FF} y_l \tag{36}$$

and by replacing the constraint in (32) by

$$\sum_{k=1}^{S} x_k \leq U_x \tag{37}$$

where U_x is another user-specified number.

4.2. Solution Procedures. In essence, the objective of IP_1 is to select a minimal subset from the set of all possible sensor pairs in order to perform the required diagnostic functions. This set-covering problem is solved in this study with a modified version of the so-called "greedy" algorithm.[34] The detailed solution procedure is given below:

- Step 0: Produce a working copy of the bipartite matrix, which is referred to as the *current matrix* in this procedure.
- Step 1: Select a column from the current matrix with the maximum number of 1's, i.e., the corresponding sensor pair covers the largest number of diagnostic functions. If there are multiple choices, then one of the single-sensor pairs should be considered first.
- Step 2: Construct a *function record* for the selected sensor pair based upon the original bipartite matrix.
- Step 3: If one or more diagnostic functions in the present record can also be found in a previously constructed function record, then remove these functions from the latter.
- Step 4: From the current matrix, remove the selected column and also the rows with nonzero entries in this column. Additional columns and rows should then be eliminated to satisfy the logic constraints (33)−(35). Specifically,

(i) if the present selection is associated with one of the first S columns in the original bipartite matrix, identify the redundant sensor pairs among those represented by the remaining columns according to eq 26 and delete the corresponding columns and rows in the current matrix;

(ii) if the present selection is associated with one of the last ($SS-S$) columns in the original bipartite matrix, identify the redundant sensor pairs among those represented by the remaining columns according to eq 26 and delete the corresponding columns and rows in the current matrix.

- Step 5: If the current matrix is not empty, go to step 1. Otherwise, the next two steps should be executed.
- Step 6: Check the resulting function records of all selected sensor pairs. If any record is empty, then the corresponding selection should be excluded.
- Step 7: If all diagnostic functions in the final record of a selected two-sensor pair can be covered with only one of the two sensors in this pair, then the corresponding single-sensor pair should be adopted to replace this selection.

The above procedure can be either carried out by hand for small systems or with a computer program for larger ones. For illustration purpose, let us consider the fictitious bipartite matrix in the Table 3. The detailed solution steps for this problem are outlined below:

- Step 0: Generate a working copy of Table 3 and use it as the current matrix.
- Step 1: According to the current matrix, the sensor pair {2,3} should be selected first since it covers the maximum number of diagnostic functions.
- Step 2: The corresponding function record [(1,1), (2,2), (3,3), (1,2), (1,3)] is extracted from the original bipartite matrix and stored temporarily.
- Step 3: Since there are no previous records, this step is ignored.
- Step 4: The column corresponding to {2,3} and the rows with nonzero entries in this column are deleted from the current matrix. In addition, the columns associated with {2,2} and {3,3} should be eliminated and the rows with nonzero entries in these columns should be removed too. The resulting current matrix is given in Table 4.
- Step 5: Since the current matrix is not empty, step 1 should be carried out next.
- Step 1: From the current matrix in Table 4, it is clear that the next selection should be the sensor pair {1,4}.
- Step 2: The corresponding function record extracted from Table 3 is [(1,1), (2,2), (1,3), (2,3)].
- Step 3: The previously constructed record [(1,1), (2,2), (3,3), (1,2), (1,3)] should be reduced to [(3,3), (1,2)], since (1,1), (2,2), and (1,3) are the common functions of the two existing function records.
- Step 4: By dropping the corresponding column and row, the current matrix becomes empty.
- Step 5: Step 6 should be followed next since the current matrix is empty.
- Step 6: Since none of the selected records are empty, skip this step.
- Step 7: From the original bipartite matrix, it can be observed that the sensor pair {3,3} can also be used for the functions (3,3) and (1,2). Thus, the pair {2,3} should be replaced with {3,3}. On the other hand, it can be found that the pair {1,4} is irreplaceable. After checking all selected two-sensor pairs, the procedure can be terminated. The final selections should be sensors 1, 3, and 4.

It should be noted that eqs 26–35 can certainly be solved with the standard algorithms provided by commercial software, e.g., GAMS. The optimal solution of IP_1 for $L_y = 3$ was found to be

$$\mathbf{x} = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]^T$$

and

$$\mathbf{y} = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]^T$$

Specifically, the measured variables selected in this case are X, Z, and U, which are exactly the same as those obtained with the greedy algorithm. Finally, let us consider the integer program IP_2 and assume that U_x in eq 37 is 3. It was found that the same optimization results can be generated. However, if the value of U_x is reduced to 2, the optimal solution becomes

$$\mathbf{x} = [1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

and

$$\mathbf{y} = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0]^T$$

In other words, only the sensors for X and Z are selected and fault origins 2 and 3 are indistinguishable.

Generally speaking, the aforementioned two solution approaches can both be used to determine the optimal sensor locations with approximately the same computation time. However, the greedy search procedure can be applied by hand and, thus, should be more appropriate for small-system applications since engineering judgment can be judiciously exercised when multiple solutions are present.

5. Case Studies

Three examples are presented in the sequel to demonstrate the feasibility of the proposed solution methods. It is assumed in these examples that the probabilities of two or more coexisting failures are negligibly low. Thus, it is only necessary to consider the single-fault scenarios.

5.1. A Single-Tank Storage System with Feed-Forward Level-Control Loop. Let us first consider the level-control system presented in Figure 3a and the corresponding SDG model in Figure 3b. The definitions of model variables are listed in Appendix B. In this system, a feed-forward control strategy is used to maintain the liquid level at a desired height by adjusting the input flow according to online measurement of the flow rate at outlet. It should be noted that, although such a control scheme may not be realistic, the present example is adopted mainly to demonstrate the usefulness of proposed sensor placement strategy and also the potential benefits of incorporating the precedence orders embedded in event strings into the inference system. Specifically, it is assume in this example that the measurement signals s_5, s_6, s_7, and s_8 can be selected for fault diagnosis purpose. For simplicity, let us consider two fault propagation scenarios only. A partial malfunction in the control valve CV-01 is introduced in the first case to cause an increase in the inlet flow rate in line 1, i.e., $m_1(+1)$, while the initially closed hand valve V-03 on line 3 is accidentally opened in the second case to produce an additional input flow, i.e., $m_3(+1)$. The resulting event strings can therefore be predicted with the proposed qualitative simulation techniques, i.e.

- Origin 1:

$$s_8(+1)s_5(+1)s_7(+1)s_6(+1)[s_5(+1),\ s_6(+1),\ s_7(+1),\ s_8(+1)]_f$$

- Origin 2:

$$s_5(+1)s_7(+1)s_6(+1)s_8(+1)[s_5(+1),\ s_6(+1),\ s_7(+1),\ s_8(+1)]_f$$

By constructing the corresponding bipartite matrix and solving the optimal sensor location problem by hand, it was found that the selected sensor set is $\{s_5, s_8\}$. Notice that the same selections can also be obtained by applying the available techniques, e.g., see Bhushan and Rengaswamy.[12] However, it should be pointed out that the aforementioned fault origins cannot be differentiated with these methods by considering only the final symptoms. It is therefore necessary to enhance the diagnostic resolution by introducing additional diagnostic rules that take advantage of the different precedence orders of the two observable events, i.e., $s_5(+1)$ and $s_8(+1)$, in the two scenarios studied here. The effectiveness of the proposed strategy has been verified in extensive numerical simulation studies. The diagnosis results are presented elsewhere.[29]

5.2. A CSTR Reactor with Temperature, Pressure and Level Control Loops. The second example is concerned with an exothermic CSTR reactor with its temperature, pressure, and level control loops (see Figure 4). The corresponding SDG model is shown in Figure 5 and its model variables are defined in Appendix C. For comparison purpose, the candidate measurements (see Table 5) and the fault origins (see Table 6) considered in this example are essentially the same as those adopted in Bhushan and Rengaswamy.[12] By assuming perfect control, i.e., each NFBL is capable of bringing the controlled variable back to the normal set-point value, the measurement variables CA, T_c, and VL can be chosen with the proposed greedy algorithm. It should be noted that, with these selections, only one pair of fault origins cannot be distinguished, i.e., $T_i(+1)$ and $U(-1)$. On the other hand, an alternative sensor set $\{F, T_c, VT\}$ can also be obtained by solving IP_1 to achieve the same degree of resolution.

Notice that the above results are superior to those obtained with the traditional approach. Without considering the sensor pairs, the selected measurements should be CA_i, T_i, T_{ci}, and F_i, but the corresponding unresolved origin pair is still the same, i.e., $\{T_i(+1), U(-1)\}$. In other words, an extra sensor is used in this situation to achieve the same level of resolution.

5.3. The TE Process. The final example discussed in this paper is the well-known TE process (see Figure 6). A list of definitions of the process variables and a detailed description of the corresponding SDG model are provided in Appendix D. The candidate measurements adopted for this system are presented in Table 7. Although a total of 15 fault origins were reported in the literature,[24] only a subset of them (see Table 8) are considered here mainly for the sake of illustration and computation convenience. With the perfect control assumption, the measurements selected with the proposed greedy algorithm are P_r and F_6. Among the 36, i.e. C(9,2) different origin pairs, 6 of them cannot be distinguished with these selections. The unresolved origin pairs are $(F_2(+1),F_4(+1))$, $(C_d(-1),\ F_9(-1))$, $(F_8(+1),\ F_9(+1))$, $(C_d(-1),\ F_8(-1))$, $(F_2(-1),\ F_4(-1))$, and $(F_8(-1),\ F_9(-1))$. Notice that the observable event pairs of every fault origin in this case are presented in Table 9 for the purpose of providing a clear interpretation of the rationale behind the proposed sensor placement scheme. Finally, it can be found that the solution of IP_1 yields two alternative sensor selections, i.e., P_m and $y_{C,6}$, with the same resolution level.

6. Conclusion

The evolution sequences of fault propagation effects can be qualitatively characterized and incorporated in the SDG-based diagnosis methods to enhance resolution of the conventional approach. To facilitate online implementation of this idea, a systematic method is proposed in the present study to design the sensor networks on the basis of the event strings caused by every fault origin. This design problem is formulated as integer programs to achieve a maximum level of resolution and, at the same time, to ensure diagnostic observability. Two alternative strategies are developed for producing the optimal solutions. The feasibility of the proposed sensor placement strategy is clearly demonstrated in case studies provided in this paper.

Appendix A. An Enumeration Procedure for Identifying All Ordered Event Pairs in a Given FPP

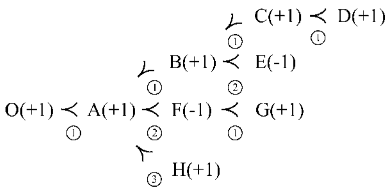

Figure A1. The fictitious SDG model.

The proposed enumeration procedure is developed on the basis of the standard depth-first search (DFS) algorithm.[34] A brief summary is given below:

- Step 1: Label the successors of each node in FPP with increasing positive integers (starting with 1).
- Step 2: Perform a depth-first search from the node representing fault origin. At each explored node, the next visit is selected according to the labels assigned in step 1. The final result is a DFS sequence, i.e., a list of nodes arranged sequentially in the order of visits during the search process.
- Step 3: Enumerate all possible ordered pairs in the DFS sequence and denote the collection of these pairs as set $\boldsymbol{S}_0$. Let $k = 0$.
- Step 4: Identify an unmarked node with more than one successor. If none can be found, generate the set of all ordered event pairs $\mathcal{S} = \cup_{i=0}^{k} \mathcal{S}_i$ and terminate the procedure.
- Step 5: Let $k = k + 1$. Mark the node identified in step 4 and identify their positions in the DFS sequence. These positions are labeled according to the order of visits in the DFS search (say, i_0, i_1, i_2,..., i_{N_k} and $i_0 < i_1 < i_2 <... < i_{N_k}$).
- Step 6: Collect all elements between position i_{n-1} and i_{n-1} in the DFS sequence into set $\mathcal{X}_{n-1}$ for n = 1, 2,..., N_k.
- Step 7: Find the Cartesian product of $\mathcal{X}_{N_{k-j}}$ (j = 0, 2,..., N_k − 1) and each of its preceding sets, i.e., $\mathcal{X}_{N_{k-j-1}}$, $\mathcal{X}_{N_{k-j-2}}$,..., and $\mathcal{X}_0$. Let the unions of these product sets be $\mathcal{S}_k$. Return to Step 4.

To illustrate the above procedure, let us consider the FPP in Figure A1 with fault origin $O(+1)$. The detailed implementation steps of the enumeration procedure are

- Step 1: Labels are given in Figure A1.
- Step 2: A DFS results in the sequence $\{O,A,B,C,D,E,F,G,H\}$.
- Step 3: $\mathcal{S}_0 = \{\{A,B\}, \{A,C\}, \{A,D\}, \{A,E\}, \{A,F\}, \{A,G\}, \{A,H\}, \{B,C\}, \{B,D\}, \{B,E\}, \{B,F\}, \{B,G\}, \{B,H\}, \{C,D\}, \{C,E\}, \{C,F\}, \{C,G\}, \{C,H\}, \{D,E\}, \{D,F\}, \{D,G\}, \{D,H\}, \{E,F\}, \{E,G\}, \{E,H\}, \{F,G\}, \{F,H\}, \{G,H\}\}$.
- Step 4: A is found.
- Step 5: $k = 1$. Successors of A are B, F, and H.
- Step 6: According to the DFS sequence in step 2, one can find $\mathcal{X}_0 = \{B,C,D,E\}$, $\mathcal{X}_1 = \{F,G\}$ and $\mathcal{X}_2 = \{H\}$.
- Step 7: Cartesian product of $\mathcal{X}_2$ and $\mathcal{X}_1$ is $\mathcal{P}_{2,1} = \{\{H,F\}, \{H,G\}\}$; Cartesian product of $\mathcal{X}_2$ and $\mathcal{X}_0$ is $\mathcal{P}_{2,0} = \{\{H,B\}, \{H,C\}, \{H,D\}, \{H,E\}\}$; Cartesian product of $\mathcal{X}_1$ and $\mathcal{X}_0$ is $\mathcal{P}_{1,0} = \{ \{F,B\}, \{F,C\}, \{F,D\}, \{F,E\}, \{G,B\}, \{G,C\}, \{G,D\}, \{G,E\}\}$. Therefore, $\mathcal{S}_1 = \mathcal{P}_{2,1} \cup \mathcal{P}_{2,0} \cup \mathcal{P}_{1,0}$.
- Step 4: B is found.
- Step 5: $k = 2$. Successors of B are C and E.
- Step 6: According to the DFS sequence in step 2, one can find $\mathcal{X}_0 = \{B,C,D\}$ and $\mathcal{X}_1 = \{E\}$.
- Step 7: Cartesian product of $\mathcal{X}_1$ and $\mathcal{X}_0$ is $\mathcal{P}_{1,0} = \{\{E,B\},\{E,C\},\{E,D\}\}$. Thus, $\mathcal{S}_2 = \mathcal{P}_{1,0}$.
- Step 4: $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2$. The procedure is terminated.

Appendix B. Definition of Process Variables in the Single-Tank Storage System with Feed-Forward Level-Control Loop

Table B1. Definitions of SDG Nodes in Example 1

parameter	definition
m_1	input flow rate
m_2	output flow rate
m_3	disturbance flow rate
h	liquid level
s_5	liquid level measurement
s_6	output flow rate measurement
s_7	controller output
s_8	input flow rate measurement

Appendix C. Definitions of Process Variables in the CSTR Reactor System with Temperature, Pressure, and Level Control Loops

Table C1. Definitions of SDG Nodes in Example 2

parameter	definition
VL	input value of valve in level controller
F_3	flow rate of valve in level controller
F_2	outlet flow rate of pump
F	outlet flow rate of reactor
V	reactor volume
VS	level sensor
VC	level controller output
CA	reactant concentration in reactor
rA	reaction rate
T	reactor temp
T_c	jacket temp
F_4	coolant flow rate sensor
Fc	coolant flow rate
VT	input value of valve in temp controller
TC	temp controller output
TS	temp. sensor
n	no. of moles of vapor
P	pressure in the vapor space
PS	pressure sensor
PC	pressure controller output
VP	input value of valve in pressure controller
F_{vg}	vent flow rate
U	overall heat transfer coefficient
T_{ci}	inlet coolant temp
C_{Ai}	inlet reactant concentrate
F_i	inlet feed flow rate
VL	input value of valve in level controller

Appendix D. SDG Model of TE Process

Table D1. Definitions of SDG Nodes in Example 3

1	T_{cr}	28	N_{Cm}	55	$y_{D,7}$	82	$x_{H,11}$	109	rho$_H$	136	$y_{F,5}$
2	C_d	29	N_{Dm}	56	$y_{E,7}$	83	R_1	110	$P_{A,s}$	137	$y_{G,5}$
3	F_1	30	N_{Em}	57	$y_{F,7}$	84	R_2	111	$P_{B,s}$	138	$y_{H,5}$
4	F_2	31	N_{Fm}	58	$y_{G,7}$	85	R_3	112	$P_{C,s}$	139	M_7
5	F_3	32	N_{Gm}	59	$y_{H,7}$	86	$P_{A,r}$	113	$P_{D,s}$	140	$F_{A,5}$
6	F_4	33	N_{Hm}	60	$y_{A,8}$	87	$P_{B,r}$	114	$P_{E,s}$	141	$F_{B,5}$
7	F_8	34	N_{Gp}	61	$y_{B,8}$	88	$P_{C,r}$	115	$P_{F,s}$	142	$F_{C,5}$
8	F_9	35	N_{Hp}	62	$y_{C,8}$	89	$P_{D,r}$	116	$P_{G,s}$	143	$F_{D,5}$
9	T_r	36	P_r	63	$y_{D,8}$	90	$P_{E,r}$	117	$P_{H,s}$	144	$F_{E,5}$
10	N_{Ar}	37	$V_{L,r}$	64	$y_{E,8}$	91	$P_{F,r}$	118	$V_{V,s}$	145	$F_{F,5}$
11	N_{Br}	38	P_s	65	$y_{F,8}$	92	$P_{G,r}$	119	$P_{D,s}^{\text{sat}}$	146	$F_{G,5}$
12	N_{Cr}	39	$V_{L,s}$	66	$y_{G,8}$	93	$P_{H,r}$	120	$P_{E,s}^{\text{sat}}$	147	$F_{H,5}$
13	N_{Dr}	40	$V_{L,p}$	67	$y_{H,8}$	94	$V_{V,r}$	121	$P_{F,s}^{\text{sat}}$	148	$V_{L,r}^{s}$
14	N_{Er}	41	P_m	68	$y_{A,9}$	95	$x_{D,r}$	122	$P_{G,s}^{\text{sat}}$	149	$C_{L,r}$
15	N_{Fr}	42	F_6	69	$y_{B,9}$	96	$x_{E,r}$	123	$P_{H,s}^{\text{sat}}$	150	$CV_{L,r}$
16	N_{Gr}	43	$y_{A,6}$	70	$y_{C,9}$	97	$x_{F,r}$	124	V_m	151	$V_{L,s}^{s}$
17	N_{Hr}	44	$y_{B,6}$	71	$y_{D,9}$	98	$x_{G,r}$	125	T_m	152	$C_{L,s}$
18	N_{As}	45	$y_{C,6}$	72	$y_{E,9}$	99	$x_{H,r}$	126	M_6	153	$\mathrm{CV}_{L,s}$
19	N_{Bs}	46	$y_{D,6}$	73	$y_{F,9}$	100	$P_{D,r}^{\text{sat}}$	127	$z_{A,4}$	154	$V_{L,p}^{s}$
20	N_{Cs}	47	$y_{E,6}$	74	$y_{G,9}$	101	$P_{E,r}^{\text{sat}}$	128	$z_{B,4}$	155	$C_{L,p}$
21	N_{Ds}	48	$y_{F,6}$	75	$y_{H,9}$	102	$P_{F,r}^{\text{sat}}$	129	$z_{C,4}$	156	$CV_{L,p}$
22	N_{Es}	49	$y_{G,6}$	76	$x_{D,10}$	103	$P_{G,r}^{\text{sat}}$	130	F_5	157	F_{10}
23	N_{Fs}	50	$y_{H,6}$	77	$x_{E,10}$	104	$P_{H,r}^{\text{sat}}$	131	$y_{A,5}$	158	F_{11}
24	N_{Gs}	51	F_7	78	$x_{F,10}$	105	rho$_D$	132	$y_{B,5}$	159	T_s
25	N_{Hs}	52	$y_{A,7}$	79	$x_{G,10}$	106	rho$_E$	133	$y_{C,5}$	160	
26	N_{Am}	53	$y_{B,7}$	80	$x_{H,10}$	107	rho$_F$	134	$y_{D,5}$	161	
27	N_{Bm}	54	$y_{C,7}$	81	$x_{G,11}$	108	rho$_G$	135	$y_{E,5}$	162	
28	N_{Cm}	55	$y_{D,7}$	82	$x_{H,11}$	109	rho$_H$	136	$y_{F,5}$	163	

Table D2. Positive Edges in SDG Model of TE Process[a]

(42,10)	(43,10)	(42,11)	(44,11)	(42,12)	(45,12)	(42,13)	(46,13)	(42,14)	(47,14)	(42,15)	(48,15)
(85,15)	(42,16)	(49,16)	(83,16)	(42,17)	(50,17)	(84,17)	(7,18)	(8,18)	(51,18)	(52,18)	(7,19)
(8,19)	(51,19)	(53,19)	(7,20)	(8,20)	(51,20)	(54,20)	(7,21)	(8,21)	(51,21)	(55,21)	(7,22)
(8,22)	(51,22)	(56,22)	(7,23)	(8,23)	(51,23)	(57,23)	(7,24)	(8,24)	(51,24)	(58,24)	(7,25)
(8,25)	(51,25)	(59,25)	(3,26)	(7,26)	(60,26)	(140,26)	(7,27)	(61,27)	(141,27)	(7,28)	(62,28)
(142,28)	(4,29)	(7,29)	(63,29)	(143,29)	(5,30)	(7,30)	(64,30)	(144,30)	(7,31)	(65,31)	(145,31)
(7,32)	(66,32)	(146,32)	(7,33)	(67,33)	(147,33)	(79,34)	(157,34)	(80,35)	(157,35)	(42,37)	(46,37)
(47,37)	(48,37)	(49,37)	(50,37)	(7,39)	(8,39)	(51,39)	(55,39)	(56,39)	(57,39)	(58,39)	(59,39)
(79,40)	(80,40)	(157,40)	(86,36)	(87,36)	(88,36)	(89,36)	(90,36)	(91,36)	(92,36)	(93,36)	(18,38)
(19,38)	(20,38)	(76,38)	(77,38)	(78,38)	(79,38)	(80,38)	(159,38)	(26,41)	(27,41)	(28,41)	(29,41)
(30,41)	(31,41)	(32,41)	(33,41)	(125,41)	(41,42)	(26,43)	(27,44)	(28,45)	(29,46)	(30,47)	(31,48)
(32,49)	(33,50)	(36,51)	(86,52)	(87,53)	(88,54)	(89,55)	(90,56)	(91,57)	(92,58)	(93,59)	(110,60)
(111,61)	(112,62)	(113,63)	(114,64)	(115,65)	(116,66)	(117,67)	(60,68)	(61,69)	(62,70)	(63,71)	(64,72)
(65,73)	(66,74)	(67,75)	(21,76)	(22,77)	(23,78)	(24,79)	(25,80)	(34,81)	(35,82)	(1,83)	(86,83)
(88,83)	(89,83)	(94,83)	(1,84)	(86,84)	(88,84)	(90,84)	(94,84)	(1,85)	(86,85)	(89,85)	(90,85)
(94,85)	(1,86)	(10,86)	(1,87)	(11,87)	(1,88)	(12,88)	(95,89)	(100,89)	(96,90)	(101,90)	(97,91)
(102,91)	(98,92)	(103,92)	(99,93)	(104,93)	(13,95)	(14,96)	(15,97)	(16,98)	(17,99)	(1,100)	(1,101)
(1,102)	(1,103)	(1,104)	(18,110)	(159,110)	(19,111)	(159,111)	(20,112)	(159,112)	(76,113)	(119,113)	
(77,114)	(120,114)	(78,115)	(121,115)	(79,116)	(122,116)	(80,117)	(123,117)	(159,119)	(159,120)		
(159,121)	(159,122)	(159,123)	(6,140)	(127,140)	(6,141)	(128,141)	(6,142)	(129,142)	(76,143)	(157,143)	
(77,144)	(157,144)	(78,145)	(157,145)	(79,146)	(157,146)	(80,147)	(157,147)	(37,148)	(148,149)	(39,151)	
(151,152)	(152,153)	(40,154)	(154,155)	(155,156)	(153,157)	(156,158)	(150,159)				

[a] Starting and ending node numbers in each pair of parentheses are defined in Table D1.

Table D3. Negative Edges in SDG Model of TE Process[a]

(51,10)	(52,10)	(83,10)	(84,10)	(85,10)	(51,11)	(53,11)	(51,12)	(54,12)	(83,12)	(84,12)	(51,13)		
(55,13)	(83,13)	(85,13)	(51,14)	(56,14)	(84,14)	(85,14)	(51,15)	(57,15)	(51,16)	(58,16)	(51,17)		
(59,17)	(7,18)	(8,18)	(60,18)	(7,19)	(8,19)	(61,19)	(7,20)	(8,20)	(62,20)	(7,21)	(8,21)		
(63,21)	(76,21)	(157,21)	(7,22)	(8,22)	(64,22)	(77,22)	(157,22)	(7,23)	(8,23)	(65,23)	(78,23)		
(157,23)	(7,24)	(8,24)	(66,24)	(79,24)	(157,24)	(7,25)	(8,25)	(67,25)	(80,25)	(157,25)	(3,26)		
(7,26)	(42,26)	(43,26)	(7,27)	(42,27)	(44,27)	(7,28)	(42,28)	(45,28)	(4,29)	(7,29)	(42,29)		
(46,29)	(5,30)	(7,30)	(42,30)	(47,30)	(7,31)	(42,31)	(48,31)	(7,32)	(42,32)	(49,32)	(7,33)		
(42,33)	(50,33)	(81,34)	(158,34)	(82,35)	(158,35)	(51,37)	(55,37)	(56,37)	(57,37)	(58,37)	(59,37)		
(85,37)	(7,39)	(8,39)	(63,39)	(64,39)	(65,39)	(66,39)	(67,39)	(76,39)	(77,39)	(78,39)	(79,39)		
(80,39)	(157,39)	(81,40)	(82,40)	(158,40)	(118,38)	(124,41)	(36,42)	(126,42)	(27,43)	(28,43)	(29,43)		
(30,43)	(31,43)	(32,43)	(33,43)	(26,44)	(28,44)	(29,44)	(30,44)	(31,44)	(32,44)	(33,44)	(26,45)		
(27,45)	(29,45)	(30,45)	(31,45)	(32,45)	(33,45)	(26,46)	(27,46)	(28,46)	(30,46)	(31,46)	(32,46)		
(33,46)	(26,47)	(27,47)	(28,47)	(29,47)	(31,47)	(32,47)	(33,47)	(26,48)	(27,48)	(28,48)	(29,48)		
(30,48)	(32,48)	(33,48)	(26,49)	(27,49)	(28,49)	(29,49)	(30,49)	(31,49)	(33,49)	(26,50)	(27,50)		
(28,50)	(29,50)	(30,50)	(31,50)	(32,50)	(38,51)	(139,51)	(87,52)	(88,52)	(89,52)	(90,52)	(91,52)		
(92,52)	(93,52)	(86,53)	(88,53)	(89,53)	(90,53)	(91,53)	(92,53)	(93,53)	(86,54)	(87,54)	(89,54)		
(90,54)	(91,54)	(92,54)	(93,54)	(86,55)	(87,55)	(88,55)	(90,55)	(91,55)	(92,55)	(93,55)	(86,56)		
(87,56)	(88,56)	(89,56)	(91,56)	(92,56)	(93,56)	(86,57)	(87,57)	(88,57)	(89,57)	(90,57)	(92,57)		
(93,57)	(86,58)	(87,58)	(88,58)	(89,58)	(90,58)	(91,58)	(93,58)	(86,59)	(87,59)	(88,59)	(89,59)		
(90,59)	(91,59)	(92,59)	(111,60)	(112,60)	(113,60)	(114,60)	(115,60)	(116,60)	(117,60)	(110,61)			
(112,61)	(113,61)	(114,61)	(115,61)	(116,61)	(117,61)	(110,62)	(111,62)	(113,62)	(114,62)	(115,62)			
(116,62)	(117,62)	(110,63)	(111,63)	(112,63)	(114,63)	(115,63)	(116,63)	(117,63)	(110,64)	(111,64)			
(112,64)	(113,64)	(115,64)	(116,64)	(117,64)	(110,65)	(111,65)	(112,65)	(113,65)	(114,65)	(116,65)			
(117,65)	(110,66)	(111,66)	(112,66)	(113,66)	(114,66)	(115,66)	(117,66)	(110,67)	(111,67)	(112,67)			
(113,67)	(114,67)	(115,67)	(116,67)	(22,76)	(23,76)	(24,76)	(25,76)	(21,77)	(23,77)	(24,77)	(25,77)		
(21,78)	(22,78)	(24,78)	(25,78)	(21,79)	(22,79)	(23,79)	(25,79)	(21,80)	(22,80)	(23,80)	(24,80)		
(35,81)	(34,82)	(2,83)	(9,83)	(2,84)	(9,84)	(2,85)	(9,85)	(9,86)	(94,86)	(9,87)	(94,87)	(9,88)	(94,88)
(37,94)	(14,95)	(15,95)	(16,95)	(17,95)	(13,96)	(15,96)	(16,96)	(17,96)	(13,97)	(14,97)	(16,97)	(17,97)	
(13,98)	(14,98)	(15,98)	(17,98)	(13,99)	(14,99)	(15,99)	(16,99)	(9,100)	(9,101)	(9,102)	(9,103)	(9,104)	
(118,110)	(118,111)	(118,112)	(39,118)	(6,140)	(6,141)	(6,142)	(149,150)						

[a] Starting and ending node numbers in each pair of parentheses are defined in Table D1.

Table D4. Control Loops in TE Process

loop no.	controlled variable	sensor output	controller output	control valve position	mainpulated variable
1	$V_{L,s}$	$V^s_{L,s}$	$C_{L,s}$	$CV_{L,s}$	F_{10}
2	$V_{L,p}$	$V^s_{L,p}$	$C_{L,p}$	$CV_{L,p}$	F_{11}
3	$V_{L,r}$	$V^s_{L,r}$	$C_{L,r}$	$CV_{L,r}$	F_7

Literature Cited

(1) Madron, F.; Veverka, V. Optimal selection of measuring points in complex plants by linear models. *AIChE J.* **1992**, *38* (2), 227.

(2) Maquin, D.; Luong, M.; Ragot, J. Observability analysis and sensor placement, SAFE PROCESS' 94 IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for technical Process, June 13−15, Espoo, Finland, 1994.

(3) Ali, Y.; Narasimhan, S. Sensor network design for maximizing reliability of linear processes. *AIChE J.* **1993**, *39* (5), 820.

(4) Ali, Y.; Narasimhan, S. Redundant sensor network for linear processes. *AIChE J.* **1995**, *41* (10), 820.

(5) Ali, Y.; Narasimhan, S. Sensor network design for maximizing reliability of bilinear processes. *AIChE J.* **1996**, *42* (9), 2563.

(6) Ricker, N. L.; Lee, J. H. Nonlinear model predictive control of the tennessee eastman challenge process. *Comput. Chem. Eng.* **1995**, *19* (9), 961.

(7) Ricker, N. L.; Lee, J. H. Nonlinear modeling and state estimation for the tennessee eastman challenge process. *Comput. Chem. Eng.* **1995**, *19* (9), 983.

(8) Lambert, H. E. Fault trees for location sensors in process systems. *Chem. Eng. Prog.* **1977**, 81.

(9) Sen, S.; Narasimhan, S.; Deb, K. Sensor network design of linear processes using genetic algorithms. *Comput. Chem. Eng.* **1998**, *22* (3), 385.

(10) Bagajewicz, M.; Sánchez, M. Design and upgrade of non-redundant and redundant linear sensor networks. *AIChE J.* **1999**, *45* (9), 1927.

(11) Bagajewicz, M.; Sánchez, M. Duality of sensor network design models for parameter esitmation. *AIChE J.* **1999**, *45* (3), 661.

(12) Bhushan, M.; Rengaswamy, R. Design of sensor network based on the signed directed graph of the process for efficient fault diagnosis. *Ind. Eng. Chem. Res.* **2000**, *39*, 999–1019.

(13) Bhushan, M.; Rengaswamy, R. Comprehensive design of a sensor network for chemical plants based on various diagnosability and reliability criteria. 1. Framework. *Ind. Eng. Chem. Res.* **2002**, *41*, 1826–1839.

(14) Bhushan, M.; Rengaswamy, R. Comprehensive design of a sensor network for chemical plants based on various diagnosability and reliability criteria. 2. Applications. *Ind. Eng. Chem. Res.* **2002**, *41*, 1840–1860.

(15) Bagajewicz, M.; Carbrera, E. New MILP formulation for instrumentation network design and upgrade. *AIChE J.* **2002**, *48* (10), 2271.

(16) Bagajewicz, M.; Fuxman, A. Instrumentation network design and upgrade for process monitoring and fault detection. *AIChE J.* **2004**, *50* (8), 1870.

(17) Kotecha, P. R.; Bhushan, M.; Gudi, R. D. Constraint programming based robust sensor network design. *Ind. Eng. Chem. Res.* **2007**, *46*, 5985–5999.

(18) Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A review of process fault detection and diagnosis. Part I: Quantitative model based methods. *Comput. Chem. Eng.* **2003**, *27*, 293–311.

(19) Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. N. A review of process fault detection and diagnosis. Part II: Qualitative model and search strategies. *Comput. Chem. Eng.* **2003**, *27*, 313–326.

(20) Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. A signed directed graph-based systematic framework for steady state malfunction diagnosis inside control loops. *Chem. Eng. Sci.* **2006**, *61*, 1790–1810.

(21) Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. Application of signed digraph-based analysis for fault diagnosis chemical process flowsheets. *Eng. Appl. Artif. Intell.* **2004**, *17*, 501.

(22) Raghuraj, B. M.; Bhushan, R.; Rengaswamy, R. Location of sensors in complex chemical plants based on fault diagnostic observability criteria. *AIChE J.* **1999**, *45* (2), 310.

(23) McAvoy, T. J.; Ye, N. Base control for the Tennessee Eastman problem. *Comput. Chem. Eng.* **1994**, *18* (5), 383–413.

(24) Downs, J. J.; Vogel, E. F. A plant-wide industrial process control problem. *Comput. Chem. Eng.* **1993**, *17* (3), 245–255.

(25) Bagajewicz, M.; Sánchez, M. Cost-optimal design of reliable sensor networks. *Comput. Chem. Eng.* **2000**, *23* (11/12), 1757.

(26) Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. A systematic framework for the development and analysis of signed digraphs for chemical processes. 1. algorithm and analysis. *Ind. Eng. Chem. Res.* **2003**, *42*, 4789.

(27) Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. A systematic framework for the development and analysis of signed digraphs for chemical processes. 2. control loops and flowsheet analysis. *Ind. Eng. Chem. Res.* **2003**, *42* (20), 4811.

(28) Chang, S. Y.; Chang, C. T. A fuzzy-logic based fault diagnosis strategy for process control loops. *Chem. Eng. Sci.* **2003**, *58*, 3395.

(29) Chen, J. Y.; Chang, C. T. Fuzzy diagnosis method for control systems with coupled feed forward and feedback loops. *Chem. Eng. Sci.* **2006**, *61*, 3105–3128.

(30) Chen, J. Y.; Chang, C. T. Systematic enumeration of fuzzy diagnosis rules for identifying multiple faults in chemical processes. *Ind. Eng. Chem. Res.* **2007**, *46*, 3635–3655.

(31) Chang, S. Y.; Lin, C. R.; Chang, C. T. A fuzzy diagnosis approach using dynamic fault trees. *Chem. Eng. Sci.* **2002**, *57*, 2971.

(32) Ju, S. N.; Chen, C. L.; Chang, C. T. Constructing fault trees for advanced process control systems application to cascade control loops. *IEEE Trans. Reliab.* **2004**, *53*, 43–60.

(33) Asratian, A. S.; Denley, T. M. J.; Häggkvist, R. Bipartite Graphs and Their Applications; Cambridge University Press: London, 1988.

(34) Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. *Introduction to Algorithms*, 2nd ed.; The MIT Press: Cambridge, MA, 2001.

Received for review December 26, 2007
Revised manuscript received May 8, 2008
Accepted May 30, 2008

IE701767V