Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

SDG-Based Fault Isolation for Large-Scale Complex Systems Solved by Rough Set Theory

Fan Yang. Deyun Xiao

Tsinghua National Laboratory for Information Science and Technology
Department of Automation, Tsinghua University, Beijing 100084, China
(Tel: +86-10-62773654; e-mail: yang-f98@mails.thu.edu.cn, xiaody@tsinghua.edu.cn)

Abstract: Signed directed graph (SDG) is an important qualitative model that is used to describe large-scale complex systems and the cause-effect relationships among variables. It has been successfully applied in fault diagnosis, hazard assessment and other areas. In the fault isolation problem, the task is to find the fault origin that causes the abnormal phenomenon. However, as the basis of analysis, the inference method based on SDG, is simply a traversal search or a rule-based expert system. Because of the redundant or disordered information, the efficiency of these algorithms is quite low. Rough set theory provides an idea of handling vague information and can be used to data reduction, thus it can be introduced to the fault isolation problem (a kind of decision problems) to optimize the decision rules. The decision algorithm is proposed in this paper, in which the generation and reduction methods of the rules are related to the structure of the SDG model. We combine the algebraic and logical expression ways to achieve the purpose. Moreover, due to the convenience of expressing granularity, the decision algorithm is still applicable when the types of the faults we concerned are changed or reformed. Finally, an example of a 65t/h boiler system is carried out to illustrate and validate the proposed method, and some future trends of this method are also discussed.

1. INTRODUCTION

Signed directed graph (SDG) is a modelling method for complex systems to exhibit the process behaviours. A SDG uses nodes to represent process variables and uses branches between nodes to represent the cause-effect relations between variables (Iri *et al.*, 1979). The sign on the node represents the direction of the variable deviation. The sign on the branch represents the direction of influence and takes the value of "+", "-" or "0". The sign "+" implies that a positive (negative) deviation leads to a positive (negative) deviation. When the sign on the edge is "-", an increase (decrease) leads to decrease (increase). Up to now, SDG has been broadly used in many areas, especially for large-scale systems such as enterprises. The most typical application covers fault diagnosis and hazard assessment based on the fault inference along consistent paths (Yang *et al.*, 2005).

The SDG-based fault isolation is actually a traversal search (usually deep-first) along the consistent paths in the graph to find the fault origin (Iri *et al.*, 1979). This approach uses the deep-level knowledge of the system and shows the propagation path of each fault. Kramer *et al.* (1987) proposed to use rules and expert systems to make inference. Yang *et al.* (2007a) introduced structural residual to simplify the rules. These approaches, however, have many disadvantages – each fault is located on a single node and the type is only the value deviation caused by device malfunction or misoperation; when multiple faults or complex faults occur, it is hard to identify them, so we have to add extra nodes to denote complex faults (Yang *et al.*, 2006b); the algorithm is fixed and not easy to be adjusted if the actual demands of users or the choices of fault types are changed. Yang *et al.* (2006c) proposed a hierarchical description of SDG, which reveals some granularity, but the decomposition and concentration methods have not been well established. The rough set theory provides a new idea of this problem.

The rough set theory was presented by Pawlak (1982, 1991) and becomes an effective new mathematical approach to uncertain and vague data analysis. Rough sets describe a kind of roughness in knowledge representation by the idea of indiscernibility between elements (more formally, indiscernibility relations). Rough set theory can be used in many areas, among which data reduction is an important application in data mining. In an information system or decision system, mass decision rules can be reduced and simplified by data reduction. Moreover, the operations among rules can be used to change the granularity or roughness. Therefore, by regarding the fault isolation problem as a decision problem, we can use rough set theory to improve the inference efficiency.

This paper is organized as follows: In Section 2, the basic concepts of rough set theory and decision algorithm are introduced. Combined with the background of the fault isolation problem and the method of SDG, the fault isolation algorithm is proposed in Section 3. Section 4 gives an example of a boiler system to illustrate the proposed method. In Section 5, the conclusions and prospective applications are discussed.

2. DECISION SYSTEM AND DECISION ALGORITHM

2.1 Basic Concepts

In order to describe a decision problem, we should first give some definitions, which are the fundamental concepts of rough set theory (Pawlak, 1982, 1991; Liu, 2001).

Definition 1 (Decision system). An information system is a formal structure viewed as a four-tuple of the form

$$S = <\mathbf{X}, \mathbf{Q}, \mathbf{V}, f>, \tag{1}$$

where, **X** is a finite universe of discourse including all elements (objects) we are interested in some problem description; **Q** is a finite set of attributes used in the description of elements of **X**; **V** describes values of all attributes; f is called a decision function

$$f: \mathbf{X} \times \mathbf{Q} \rightarrow \mathbf{V}, \tag{2}$$

indicating the attribute of the element. If the attribute set **V** is divided into two disjoint sets called condition attributes (**C**) and decision attributes (**D**), then such information systems are referred to as decision tables

$$S = <\mathbf{X}, \mathbf{Q}, \mathbf{C} \cup \mathbf{D}, f>. \tag{3}$$

Definition 2 (Indiscernibility). With the same information system S in mind, denote by **A** a subset of attributes, $\mathbf{A} \in \mathbf{Q}$. We say that two objects (x and y) are indiscernible by the set of attributes **A** in S iff f(x,a) = f(y,a) for every a in **A**.

Indiscernibility forms an equivalence relation in **X**,

$$\mathrm{IND}(\mathbf{A}) = \{(x,y) \in \mathbf{X} \times \mathbf{X} \mid f(x,a) = f(y,a) \text{ for all } a \in \mathbf{A}\} \tag{4}$$

Definition 3 (Partition and class). Given the information system S and its condition attributes **C** and decision attributes **D**, **X**|IND(**C**) and **X**|IND(**D**) are the partitions of the universe **X** on the attribute sets **C** and **D** respectively. The elements in the sets **X**|IND(**C**) and **X**|IND(**D**) are called condition classes and decision classes respectively.

Definition 4 (Consistency). Given the information system S and its condition attributes **C**, if each condition class $E \in \mathbf{X}|\mathrm{IND}(\mathbf{C})$ has the same decision value, then we call E is consistent, otherwise we call E is inconsistent. For a decision table S, if all the condition classes are consistent, then S is consistent; otherwise S is inconsistent.

Definition 5 (Core and reduct). For an information system S with a subset $\mathbf{A} \in \mathbf{Q}$, attribute $a \in \mathbf{A}$ is dispensable if IND(**A**)= IND(**A**\{a}). Otherwise we call a to be indispensable. The set of all indispensable attributes of **A** is called a core of **A**, denoted by CORE(**A**). A minimal set of attributes that discerns all objects in S that are discernable by **A** and cannot be further reduced is called a reduct of **A**, denoted by RED(**A**). The intersection of all reducts of **A** is a core of **A**

$$\mathrm{CORE}(\mathbf{A}) = \cap_{\text{all reducts}} \mathrm{RED}(\mathbf{A}). \tag{5}$$

CORE(**A**) is composed of such attributes that cannot be removed from **A** without causing any loss in the quality of classification.

As to the expressions of the decision algorithm, besides the table form, we can also express it in logic form.

Definition 6 (Rough logic). Rough logic language (RLL) is composed of attribute set **Q**, value set $\mathbf{V} = \cap_{q \in \mathbf{Q}} \mathbf{V}_q$ ($\mathbf{V}_q$ is the value set of the attribute q), logic connectives, and well formed formulas (wffs):

(1) (q,v) is an atomic formula, where $q \in \mathbf{Q}$, $v \in \mathbf{V}_q$, and atomic formulas are all wffs;

(2) If φ and ψ are formulas, then $\sim\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, (φ), and $\varphi \rightarrow \psi$ are all wffs;

(3) The result of operating the formulas defined in (1) and (2) by logic connectives for limited times, is a wff.

Definition 7 (Granule). Function $f^{1}(\varphi)$ denotes the object set in which the element satisfies the wff φ. A granule is defined as Gr = (φ, $f^{1}(\varphi)$).

By the operation of wffs, granules are also changed and describe things from different levels. For example, if two wffs are operated by '$\vee$', the corresponding granules combines into a bigger one. So the granularity is lower because we cannot distinguish the initial two granules.

2.2 Decision Algorithm

By the above definitions, we can describe a decision algorithm by a decision table or rough logics. A decision table can be regarded as a set of formulas. We should deal with all the possible decision rules and obtain a concise and self-contained decision algorithm. The way of inference is as follows (Liu, 2001):

(1) List all the possible rules as *Table A* (as Table 1), with each row denoting a rule $\varphi \rightarrow \psi$, where φ denotes the values of the condition attributes are assumed and ψ denotes the decision to be obtained. For convenience, we can give each attribute value a notion.

Table 1. The framework of a decision table

Attributes Q / Objects X	Condition attributes **C**	Decision attributes **D**

(2) Try to delete each condition attribute in turn and test the consistency of the formula and obtain the reducts and the core. Delete all the elements except the cores and get *Table B*. There are several methods to test the consistency. For example,

– Each condition class $E \in X|IND(\mathbf{C})$ has the same decision value.

– For each object x, the condition class covering x is contained in the decision class covering x.

– For every two decision rules $\varphi \to \psi$ and $\varphi' \to \psi'$, we have $\varphi = \varphi' \to \psi = \psi'$.

(3) Calculate the reducts of each rule by use of Table B, and get *Table C*.

(4) Delete redundant rules and thus get *Table D*.

(5) Educe the rules and the decision algorithm according to Table D.

The decision algorithm derived here assures the minimization of the resulted condition attribute set. Theoretically it is an NP-hard problem (Pal *et al.*, 2001).

3. FAULT ISOLATION ALGORITHM BASED ON THE SDG MODEL

Fault isolation problem is an instance of decision problem. The purpose is to determine the system state, that is, normal or abnormal? Where does the fault occur? What kind of fault?

In expert systems, the decision is realized by rules. It is similar here that in order to express the decision problem in the framework of rough set theory, we should transform the problem expression into decision table or rough logics at first.

The basic method is to take the variable set as the condition attribute set, and to take all the possible samples (combinations of all the variable values) as the objects, and to take the system states as the decision values. In SDG, the values of the variables are "+", "-", or "0", so the condition attribute set is composed of these three signs. The decision attribute is the system state including all the kinds of faults and a normal state.

According to the structure of the SDG, we can get some observations:

Observation 1. If two variable sets have no intersection and no branches linking them, then the rules concerning these two sets are independent. For example, in Fig. 1, variable a and {b, c, d} are separate, so in Table 2, b, c, and d can be reduced in the 1st row, and a can be reduced in the 2nd and 3rd rows.

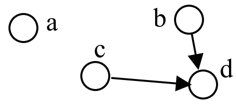

Fig. 1. An example of SDG.

Table 2. An example of the decision table

	a	b	c	d	State
1	+	/	/	/	F1
2	/	+	/	/	F2
3	/	/	+	/	F3
4	0	0	0	0	Normal

Observation 2. If two or more nodes in SDG have the same downstream node, then the core is null, because the conjunct node and its upstream nodes can all be reduced. In Table 2, the 2nd and 3rd rows are reducts and there are other reducts omitted here.

Considering the different granularity, the rules can be combined or disjoined. In the SDG in Fig. 1, if we only pay attention to the fault of two clusters, a and {b, c, d}, then the 2nd and 3rd rows can be combined together.

The advantages of this method are: Firstly, it can figure out the complex fault problem that is hard to handle by the search on SDG. Secondly, faults can be classified into different types and combined into different granules according to our actual needs, and are not limited to the value deviation of the variables. Thirdly, the samples are possible decision rules that can be created by the actual measurement or by inference on SDG. So this method takes SDG method into account to the sample generation, but for real-time inference, it also uses the expert system to improve the efficiency.

4. EXAMPLES

A 65t/h steam boiler system, which is widely applied in large-scale petrochemical enterprise, is taken as our example system. We implement it by simulation software PS (Wu, 2002) with its flow chart shown as Fig. 2. Some key variables in the process are controlled by single loops, and valves can be manipulated manually or automatically. There are 16 controlled variables in the model: inlet flow rate of the boiler, FR-01; outlet flow rate of the overheated steam, FR-02; flow rate of the cooling water, FI-03; flow rate of the soften water, FR-04; flow rate of the smoke, FI-06; flow rate of the fuel oil, FR-07; flow rate of the deoxidizing water to be catalyzed, FI-08; pressure of the smoke at the exit, PI-05; oxygen percentage of the smoke, AI-01; pressure of the main steam, PIC-01; pressure of the high pressure gas, PIC-02; pressure of the liquid hydrocarbon, PIC-03; pressure of the deaerator, PIC-04; water level of the top steam drum, LIC-01; water level of the deaerator, LIC-02; and temperature of the overheated steam, TIC-01. Besides, key variables in the process include the temperature of the hearth, TI-07; the flow rate of the inlet air, F_A; and the flow rate of the high-, medium- and low-pressure gas denoted by F_H, F_M and F_L individually. The SDG of this system is established by Yang *et al.* (2006a), shown as Fig. 3, which illustrates the cause-effect relationships among the key variables.

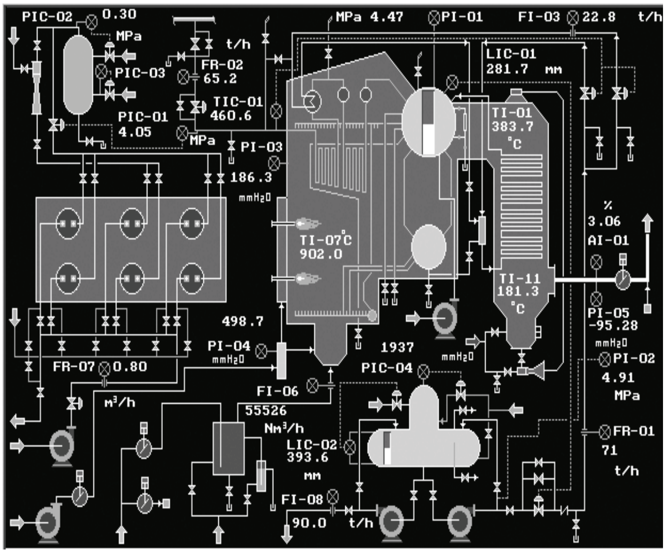

Fig 2 Flow chart of the boiler system.

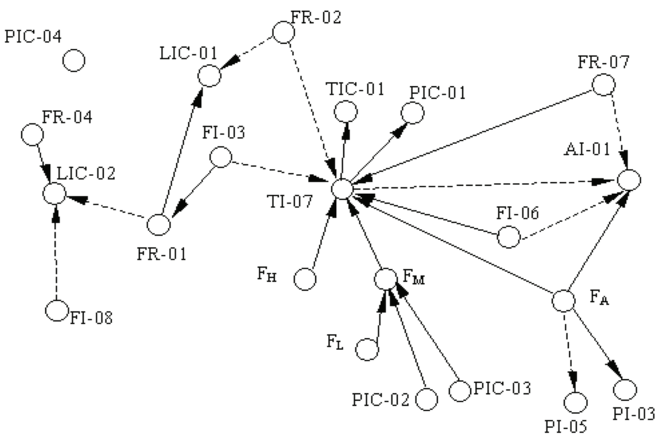

Fig 3 SDG of the boiler system.

The most typical faults are operational malfuctions, because each controlled variable is involved in a control loop and is controlled by a valve, and the open of the loop or the wrong settings may lead to the corresponding fault, that is, the deviation of the controlled variable. Besides, there are other kinds of faults caused by complicated or compositive reasons. These faults are listed in Table 3 together with their consequences.

Table 3. Typical faults of the boiler system

Notation	Name	Consequence
F2	Full of water in steam drum	Inlet reduces heavily
F3	Lack of water in steam drum	Water level decreases gradually
F4	Fire extinguished	All the gas muzzles are extinguished; pressure and temperature of the stream decrease
F5	Power off	Several phenomena
F6	Failure in the cooler	Temperature of overheated steam reduces; cooling water reduces abnormally, etc.

First, we consider three operational malfunctions C1, C8, and C7, which correspond to the deviation of PIC-04, PIC-02, and PIC-03 individually. According to SDG, we can search along the consistent paths to find out the consequence of each operational malfunction, the affected variables of which are listed in Table 4.

Table 4. The affected variables of three operational malfunctions

	PIC-04	TIC-01	PIC-02	PIC-01	PIC-03	AI-01
C1	+	0	0	0	0	0
C8	0	+	0	+	+	-
C7	0	+	+	+	0	-

According to the above decision algorithm, this table can be regarded as a decision table (Table 5) by adding a row denoting normal state, whose condition attributes are all the variables and whose decision attribute is the state of the system. Obviously this table is consistent. And because of the redundancy, the column of PIC-01 and AI-01 can be deleted. In the first row, which determines the fault C1, the columns except PIC-04 are reducible and thus PIC-04 is the core. In the next two rows, all the attributes are reducible, so there are no cores. But we can reduce PIC-04 and choose one or two attributes to combine a reduction. In its SDG, PIC-04 is a separate node, so it becomes the only attribute after reduction. PIC-02 and PIC-03 have the same downriver nodes, so at least one of these two attributes must be chosen to identify the two faults. Here we have validated the two observations in Section 3.

Table 5. Decision table when considering three operational malfunctions

PIC-04	TIC-01	PIC-02	PIC-01	State
+	0	0	0	C1
0	+	0	+	C8
0	+	+	+	C7
0	0	0	0	Normal

Finally we obtain the decision table by collecting all the reductions, shown in Table 6.

Table 6. Reduced decision table when considering three operational malfunctions

PIC-04	TIC-01	PIC-02	PIC-03	State
+	/	/	/	C1
/	/	/	+	C8

/	+	0	/	C8
/	/	+	/	C7
/	+	/	0	C7
0	/	0	0	Normal

The decision rules are written with rough logic formulas as follows:

(PIC-04,+)→C1

(PIC-03,+)→C8

(TIC-01,+) ∧ (PIC-02,0)→C8

(PIC-02,+)→C7

(TIC-01,+) ∧ (PIC-03,0)→C7

(PIC-04,0) ∧ (PIC-02,0) ∧ (PIC-03,0)→Normal

Or we can combine them as

(PIC-04,+)→C1

(PIC-03,+) ∨ ((TIC-01,+) ∧ (PIC-02,0))→C8

(PIC-02,+) ∨ ((TIC-01,+) ∧ (PIC-03,0))→C7

(PIC-04,0) ∧ (PIC-02,0) ∧ (PIC-03,0)→Normal

Obviously, the so-called normal state is not a real normal state, because there are many other faults that have not been considered here. Moreover, the combinational faults, such as the simultaneous fault of C8 and C7, are also not considered here. We note that these two faults compose a fault on the superior level or lower granularity, which means the flow rate of the medium-pressure gas F_M is deviated and ignore its original cause.

Up to now, we find that the search method and the method proposed in this paper can both meet the demands of the fault isolation problem in large-scale complex systems. SDG method demonstrates the propagation process of the faults, so it is more applicable in the system analysis. However the method in this paper reduces the variables to be considered, so it is more applicable in the real-time diagnosis.

When a fault occurs, the affected variables are probably not located in a concentrated area, but in several parts that are not connective in the graph. For example, when power off (F5), many variables will be abnormal and they consist of several origins. Table 7 is a sample set of the typical faults.

Table 7. Sample set of the typical faults

	C1	C8	C7	F2	F3	F4	F5	F6
PIC-04	+	0	0	0	0	0	0	0
FR-04	0	0	0	0	0	-	-	0
LIC-02	0	0	0	0	0	0	+	0
FR-01	0	0	0	+	-	-	-	0
FI-08	0	0	0	0	0	0	-	0
LIC-01	0	0	0	+	-	0	-	0
FI-03	0	0	0	0	+	-	-	-
FR-02	0	0	0	0	-	-	-	0
TIC-01	0	+	+	0	0	0	-	-
PIC-02	0	0	+	0	0	0	0	0
PIC-01	0	+	+	0	0	-	-	0
PIC-03	0	+	0	0	0	0	0	0
FR-07	0	0	0	0	0	0	-	0
FI-06	0	0	0	0	0	0	0	0
PI-05	0	0	0	0	0	0	+	0
AI-01	0	-	-	0	0	+	0	0
PI-03	0	0	0	0	0	-	0	0

By the above algorithm, we can execute the data reduction to get the decision table (Table 8) and decision rules.

Table 8. Decision table without reduction

PIC-04	LIC-01	FI-03	FR-04	PIC-02	PIC-03	State
+	0	0	0	0	0	C1
0	0	0	0	0	+	C8
0	0	0	0	+	0	C7
0	+	0	0	0	0	F2
0	-	+	0	0	0	F3
0	0	-	-	0	0	F4
0	-	-	-	0	0	F5
0	0	-	0	0	0	F6
0	0	0	0	0	0	Normal

5. CONCLUSIONS

This paper combines the theories of SDG and rough sets to improve the inference efficiency of decision problem and uses them to the fault isolation problem. It shows the flexibility of building the rules and classifying the faults. The decision algorithm proposed here implemented is to obtain the minimum set to achieve the purpose.

Also, this method can be used in other problems, such as sensor location problem in system design. In order to monitor the state or performance of the system, many sensors are placed on the devices to measure the variables. Theoretically, more sensors located in more places are better for fault detection, but because of economic and technical limitations, we cannot use too many sensors. The basic principles are set to be able to detect all the faults (i.e. detectability) and to distinguish between different faults (i.e. identifiability) (Yang *et al.*, 2007b). These two principles are consistent with the decision problem. Thus we can choose the minimum attribute set as the sensor location set. And we can expand, reduce, or reform the table according to the demands of the fault isolation task.

ACKNOWLEDGMENT

The work was sponsored by the National High Technology Research and Development Programme of China (No. 2003AA412310) and National Natural Science Foundation of China (No. 60736026). We gratefully acknowledge the financial aid for this research.

REFERENCES

Bargiela, A. and W. Pedrycz (2003). *Granular Computing: an introduction*. Kluwer Academic Publishers, Boston/Dordrecht/London.

Iri, M., K. Aoki, E. O'shima, and H. Matsuyama (1979). An algorithm for diagnosis of system failures in the chemical process, *Computers and Chemical Engineering*, **3**, 489-493.

Kramer, M.A. and B.L. Palowitch (1987). A rule-based approach to fault diagnosis using the signed directed graph, *AIChE Journal*, **33**, 1607-1078.

Liu, Qing (2001). *Rough sets and rough inference*. Science Press, Beijing.

Pal, S.K. and A. Skowron (1999). *Rough fuzzy hybridization. A new trend in decision-making.* Springer-Verlag, Singapore.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences*, **11**, 341-356.

Pawlak, Z. (1991). *Rough sets – theoretical aspects of reasoning about data*. Kluwer Academic Publishers. Dordrecht.

Wu, Chongguang (2002). Modeling and simulation of 65t/h steam boiler for operation analysis, *Boiler Technology*, **33**(9), 1-6.

Yang, Fan and Deyun Xiao (2005). Review of SDG modeling and its application. *Control Theory and Applications*, **22**, 767-774.

Yang, Fan and Deyun Xiao (2006a). Model and fault inference with the framework of probabilistic SDG, *Proceedings of 9th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006)*, Singapore, 1023-1028.

Yang, Fan and Deyun Xiao (2006b). Probabilistic SDG model and approach to inference for fault analysis. *Control and Decision*, **21**, 487-491.

Yang, Fan and Deyun Xiao (2006c). Hierarchical description of SDG model and its fault inference, *Proceedings of 2006 Seminar on Production Safety and Control in Petrochemical Industry*, Beijing, 1-5.

Yang, Fan and Deyun Xiao (2007a). The application of structural residuals in the SDG-based fault isolation strategies. *Control Engineering of China*, **14**(3), 320-324.

Yang, Fan and Deyun Xiao (2007b). Model and fault inference with the framework of probabilistic SDG, *Proceedings of 2007 IEEE International Conference on System, Control and Cybernetics (SMC 2007)*, Montreal, Canada, 2843-2847.