ATLANTIS PRESS

Advances in Engineering Research, volume 125

International Conference on Material Science, Energy and Environmental Engineering (MSEEE 2017)

The Study of Fault Diagnosis Method Using A* Search and OBDD

Bo Pang[1, 2], Zhigang Huang[1], Wenquan Feng[1], Wenfeng Zhang[1], Baoling Fu [3]

[1]School of Electronic and Information Engineering, Beihang University, Beijing, 100191, China;

[2]Beijing Institute of Spacecraft System Engineering, Beijing, 100094, China;

[3]Leadcore Technology Co.Ltd, Datang Telecom Technology & Industry Group, Beijing, 100191, China

Keywords: A*Search; Ordered Binary Decision Diagram (OBDD); Diagnosis; Model; Compilation

Abstract. Independent fault diagnosis is an effective method to improve the safety and reliability of satellite orbit operation. Model-based diagnosis approach has been widely studied and applied because of high accuracy, strong adaptability and other advantages. However, because of the computing and store resource are very limited in the satellite system, online reasoning of model based fault diagnosis method is greatly challenged by the low time efficiency. This paper applies a compilation Based diagnosis method used A* search and Ordered Binary Decision Diagrams (OBDD), which compile the system model to OBDD style, through store the relation between the Observations and Diagnosis results replace the on-line reasoning, use the A* algorithm to search result. This method can significantly improve the efficiency of online diagnostics and has a wide range of application value.

1. Introduction

Because of the long time working in the abominable space environment, components in satellite system are prone to failure. Once a component fails, the fault may spread quickly, and if it is not dealt with promptly, it may cause irreparable damage. At present, satellite fault diagnosis methods are mainly based on manual analysis, which is inefficient, and great influenced by the subjective factors of the staff. It is easy to make mistakes when applied to large-scale systems. In order to ensure stable and reliable operation of the satellite, satellite system needs more intelligent fault diagnosis methods.

Model Based Diagnosis (MBD) in the system model, comparing consistency between sensor data and system model to judge whether a fault occurs and further fault isolation is one of the research hotspot in the field of fault diagnosis. According to the characteristics of the equipment and the actual diagnostic requirements, the system model can be divided into three categories: qualitative, quantitative and mixed. Because of the complexity of the satellite systems, it is very difficult to establish exact quantitative model. Therefore, the fault diagnosis method based on qualitative model can rapidly improve the fault diagnosis ability of satellite system.

First, the consistency based diagnosis [1] uses propositional logic to describe the system model and converts it to Boolean function form; then assigns the observation variables according to the sensors data; finally finds the Boolean function for unknown variables really. Since the unknown variables have 2^n combinations, the time required for diagnosis is exponentially related to the unknown variables. In order to improve the efficiency of online diagnosis, the famous general diagnostic engine (GDE) use Assumption based Truth Maintenance System (ATMS) cache intermediate results for rapid diagnosis reasoning[2]; Williams proposed Conflict Directed A* algorithm(CDA*) use the A* search algorithm to reduce reasoning; Feldman proposed the Safari random search algorithm to find the optimal solution for diagnosis. Although these methods greatly improve the efficiency of online diagnosis, reasoning still consumes a lot of time.

The compilation [3] based diagnosis can pre-save the diagnosis solutions according to the observation value, and can greatly reduce the on-line diagnosis time. In this paper, a perfect consensus diagnosis framework is proposed, and the efficiency of partial order solution extraction is improved by using A* search algorithm, and is validated in a satellite thermal control system.

2. Background knowledge

2.1 The basic concept of consistency diagnosis

Discrete model based fault diagnosis uses discrete variables to describe system behavior, and the system model is obtained. When the system is running, according to the model inference system output, according to the model output and observation output is consistent, to determine whether there is a fault.

The discrete variables in system are divided into three categories: 1) observation variables, we can express through the sensor obtained the values of the variables; 2) internal variables, which cannot be directly measured by the sensor; 3) model variables that indicate the health state of system.

To facilitate reasoning, we generally encode system models into Boolean variables and connect them with logical symbols such as, or, and not. A solution is an assignment for the diagnosis of the model variables, the observation and the assignment, said the system model can satisfy the Boolean function (i.e. there exists at least one group of internal variable assignment, the function is true), for a description of the system can be defined as:

System description (SD)：The system description is defined by set pair (SV, DT), among them:

SV (System Variable) is a system variable, can be divided into OBS, MODE (observation variables) (model variables, each element of the domain represents a possible element model), INNER (internal variables) of three sets.

DT is a domain theory (Domain Theory), defined as a collection of SV logic.

The corresponding diagnostic problem can be defined as:

The diagnosis problem (DP). The diagnostic problem is defined by two tuple DP = (SD, OBS). Among them, SD refers to the system description, and OBS is an instance of set observation variable set OBS.

The Consistency-Based Diagnosis (CBD):Given a set of logical propositions set SD and observation value set OBS, assignment on the modal variable MODE is a consistent diagnostic solution, if and only if

$$SD \cup \boldsymbol{OBS} \cup \boldsymbol{MODE} \nvdash \perp \tag{1}$$

In order to facilitate reasoning, the SD should be converted to Boolean functions form, the OBS and MODE observation assignment model variables into the Boolean function, when this function can satisfy (Satisfiability), the consistency set up.

2.2 Ordered binary decision diagrams

Binary Decision Diagram (BDD) is a method of describing Boolean functions using tree structure. BDD expands all variables in the Boolean function according to the values of ***true*** and ***false***, and removes redundant components so that Boolean functions can be efficiently represented. OBDD [4, 5, 6, 7, 8] is based on BDD and requires that each layer must have the same variable. The OBDD representation of the Boolean function given by function (2) is shown in Figure 1. The solid line below the node represents that the node variable is ***true***, the dotted line represents ***false***, and the final leaf node **1/0** indicates that the function is ***true*** or ***false*** at that value. For example, path 2 means that when $\boldsymbol{m}$ is true and $\boldsymbol{x_1}$ to $\boldsymbol{x_3}$ are ***false***, the function (2) is ***true***.

As shown in Figure 2(c), through merging the same leaf nodes, the same internal nodes and deleting the internal redundant nodes, we can get the simplest Reduced OBDD(ROBDD) of the Boolean function (2). If in some path, there are some variables does not appear on a certain level, it indicates that the value of the variable does not affect the result of the function on the path. ROBDD greatly compresses the OBDD and can compactly represent the system model.

$$f(m, x_1, x_2, x_3) = !m + (x_1 + x_2) \bullet x_3 + !x_1 \bullet !x_2 \bullet !x_3 \tag{2}$$

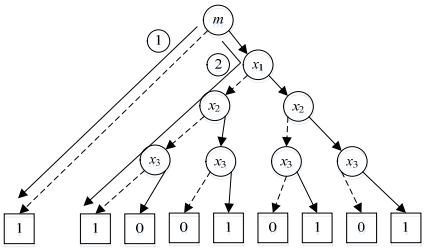

Fig.1 OBDD REPRESENTATION OF FUNCTION f

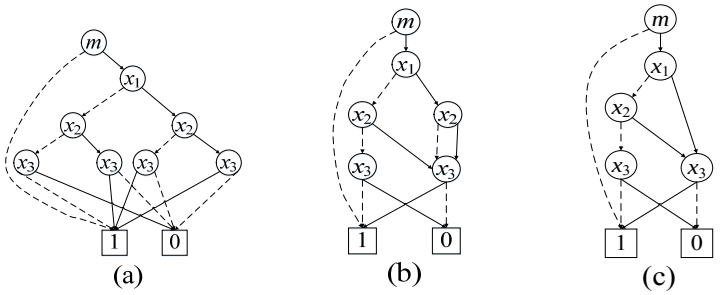

Fig.2 ROBDD REPRESENTATION OF FUNCTION f

The whole diagnosis process is implemented by basic operations of OBDD including *CONSTRUCT*, *APPLY* and *RESTRICT* which are introduced and analyzed below.

2.2.1 CONSTRUCT Operation

CONSTRUCT operation constructs a new OBDD representing Boolean function *f* based on Shannon expansion (3).

$$f(x_1, x_2, ..., x_n) = !x_i \bullet f(x_i = 0) + x_i \bullet f(x_i = 1) \tag{3}$$

Before calling *CONSTRUCT* operation, a variable sort *S* must be given where *S*(*i*) is the i_{th} variable. *CONSTRUCT* operation firstly creates a node *v* labelled by variable *S*(1). Then nodes for $f(S(1) = 0)$ and $f(S(1) = 1)$ are created and *S*(1)'s two arcs are directed to them respectively. When all variables are assigned values, the recursion is stopped and we get the result of *f* under a special assignment.

2.2.2 APPLY Operation

By applying algebraic operations to other functions, The *APPLY* operation generates new Boolean functions. For Boolean operator <*op*> (such as *AND*/∧ or *OR*/∨), given Boolean function *f* and *g*, *APPLY* operation returns new function *f* <*op*> *g*. *APPLY* operation can also be implemented by "commute" Shannon expansion (4).

$$\begin{aligned} f < op > g = & !x \bullet (f|_{x \leftarrow 0} < op > g|_{x \leftarrow 0}) \\ & + x \bullet (f|_{x \leftarrow 1} < op > g|_{x \leftarrow 1}) \end{aligned} \tag{4}$$

For any Boolean function *f* and *g* with the same ordered variables, Shannon expansion could be applied recursively until <*op*> is applied on two constants. Because the OBDD representation of a Boolean function is a DAG, we identifies the function by the root of the DAG and the function and the root vertex are not distinguished later.

2.2.3 RESTRICT Operation

To restrict variable *x* to value *k*, we can simply delete the invalid arc for vertex *v* having *var*(*v*)=*x*. Any arc into *v* is redirected either to *lo*(*v*) for *k*=0 or to *hi*(*v*) for *k*=1.

2.3 A* search algorithm

The search problem can be defined by tuple

$$<\sum, \Theta, \text{Ops, Goal-Test, g}> \tag{5}$$

Where, $\sum$ is the search space and includes all possible solutions; Θ is the initial state of the problem; Ops is the operation of the search algorithm, Ops: $\sum \rightarrow \sum$, which maps the current state to the next state; Goal-Test: $\sum \rightarrow$ {True, False} is the test function that specifies whether the state satisfies the problem's goal. G: $\sum \rightarrow$R is the cost function that assigns a cost value for each state. The problem of state space search generally refers to the search process directed by Ops to find a minimum cost of the state from the initial state.

Ops employs A* search algorithm [9] to find the next lowest cost possible solution which is passed to Goal-Test for testing. To achieve this goal, the A* search algorithm chooses the value of each state variable one by one and a heuristic function, for example (6) in this paper, is used to direct the process.

$$f(x)=g[problem](x)+h(x) \quad (6)$$

The g[problem](x) calculates the cost of the variable that has been assigned according to the cost function g, and h(x) estimates the cost of the unassigned variable. The A* search algorithm starts with the first unassigned variable, each time expanding the node with the least cost in the node to be deployed until a fully expanded and minimal cost state is found. As long as H (x) does not estimate the cost of unassigned variables, then the algorithm returns the optimal result. Algorithm 2 gives the partial order extraction algorithm pseudo code based on A* search.

3. Off-line compilation diagnostics based on OBDD and A* search

This chapter mainly introduces how to use the OBDD compiler based on A* search based fault diagnosis scheme, the diagnosis system framework as shown in Figure 3.

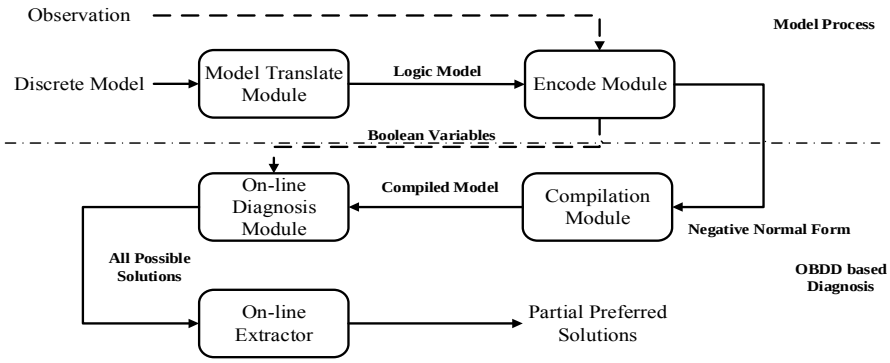

Fig.3 The Framework of OBDD based Diagnosis

The input of the diagnosis system includes the system discrete model and observation information. Among them, the system discrete model is a hierarchical model based on the system characteristics. The model transformation module converts the model into a standard model and is expressed in a logical paradigm, which records all constraints within the system. However, because the logical paradigm model may contain enumerated variable types, and OBDD requires all variables to be Boolean variables, it cannot be compiled directly into OBDD. The encoding module uses Boolean variables to encode enumerated variables and converts the logical paradigm model into a Boolean model. In addition, when the observation information is input, the coding module is responsible for assigning the corresponding Boolean variables and providing online diagnostic use. The offline compiling module compiles it into the form of OBDD according to the encoded Boolean model, and gets the compiled Boolean model. The online diagnostic module can assign all the feasible diagnosis solutions at once by the compiled Boolean model and the observed Boolean variables. The diagnosis solution is stored in the form of OBDD, and any schema variable assignment included in the path from the root node to the 1 node is a feasible diagnostic solution. Finally, the extraction module extracts the partial optimal solution according to the prior information.

3.1 Model processing

The consistency of the diagnosis framework proposed in this thesis, using the finite state automaton to describe the transfer relationship between work mode and mode components, using propositional logic to describe the internal behavior of each mode system. Because the satellite system is large in scale, complex in structure and dynamic in component behavior, it is very difficult to describe the system behavior directly by using logical proposition or Boolean function in engineering, and the method of hierarchical modeling can be adopted.

After the layered model is flattened, the model's complete behavior model is obtained. The modeling statement describing the behavior of the system is then transformed into a standard logical paradigm (negative paradigm), and the conversion rules are shown in table 2.

Table 1. Model transformation rule

Behavior model	Standard logical paradigm
a==b	a·b+(!a)·(!b)
a!=b	(!a+!b)·(a+b)
if(a) b	!a+b
if(a) b else c	a·b+!a·c

The encoding module is responsible for encoding enumerated variables into Boolean variables. In this paper, a simple encoding scheme is adopted to treat the different values of enumerated variables as a Boolean variable. That is, when the boolean variable is true, the enumeration variable is represented as the corresponding value. In the encoding scheme, a boolean variable describing an enumeration variable with and with only one boolean variable is true.For example, for enumeration variables, the range of ***enum*** is $\{e_1, e_2, \ldots e_n\}$, then there is a boolean variable $enum@e_i=ture$ (i=1… n) indicating the value of *enum* is $\boldsymbol{e_i}$.

3.2 OBDD based compilation diagnosis

As shown in Figure 3, the OBDD based compilation diagnosis include three parts, the offline compilation module, the online diagnosis module, and the online extraction module.

offline compilation module

The offline compilation module uses the BUILD operation to compile the system Boolean function model fmodel into OBDD form Om. BUILD operations have a variety of implementation methods, the most classic methods include the direct expansion method and APPLY polymerization method.

direct expansion method

On any of the fmodel variables of v, follow v=true and v=false in turn, until you can get the value of the function fmodel, and expand all variables, that is, the OBDD representation of the fmodel. The method is clear and easy to implement, but the efficiency is very low

APPLY polymerization method

APPLY polymerization method in and or or node fmodel first construct the OBDD, then two OBDD polymerization using APPLY method, finally fmodel OBDD said.

In the worst case , the two approach requires 2n times operations (n is the number of variables)。But APPLY aggregation is usually more effective. In addition, the simplification operation can be performed after the OBDD representation of the fmodel is generated, or in the construction process, and the fmodel of the ROBDD is directly obtained. Both require the same amount of time, but construction and simplification simultaneously can significantly reduce the need for intermediate nodes in the compilation process and lower the requirements of the compiler platform

online diagnosis module

The online diagnosis module loads the observations into Om according to the observation, and uses the RESTRICT method to obtain the OBDD representation Odia of all the diagnostic solutions consistent with the observations.

```
1 Function ComputeDiagnoses (Om, obs)
2   Odia := Om
3   ForEachO(v) ∈ obs
4           Odia := restrict(OTEMP , O(v)B)
5   Return Odia
6 End Function
```

Algorithm 1. offline compilation module pseudo code

As shown in algorithm 1, according to the OBS, ***ComputeDiagnoses*** restricts system model O_m(line 4) by observation ***obs*** (line 3) and return the final diagnosis results O_{dia} where any path from the root node to the 1 node represents a consistent solution with observation. As shown in Table 1, the

BUILD operation has a worst-case complexity of $O(2^n)$, which means an exponential increase in the number of variables. So the compilation process takes a great deal of time. But because the complexity of RESTRICT operation is O (|O|), it is completed within the linear time complexity.

on-line extractor

According to a priori probability, extractor extracts preferred diagnosis solutions. The number of faulty components or the joint probability may be the criterion. In this paper, we use the A* search (Russell and P. Norvig) to find the optimal diagnosis solution, and its loss (cost) function definition is as shown in (7) (fault component number) or (8) (joint probability):

$$\begin{cases} f(n) = g(n) + h(n) \\ g(n) = \sum_{i \in I} 1 \\ h(n) = 0 \end{cases} \tag{7}$$

$$\begin{cases} f(n) = g(n) + h(n) \\ g(n) = \sum_{i \in I} -e^{p_i} \\ h(n) = 0 \end{cases} \tag{8}$$

where I is the set of mode variables assigned *true*.

Since the loss function never "excessively" estimates ***cost,*** the results of the A* search algorithm are complete and optimal.This method reduces the workload of online offline compilation; diagnosis without exclusion of internal variables, thus less time consuming; partial extraction stage using the A* search can not only use the weights of evaluation methods based on, and the extraction efficiency and flexibility is also higher.

3.3 OBDD based diagnosis example

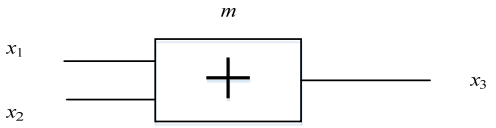

Figure 4. Adder

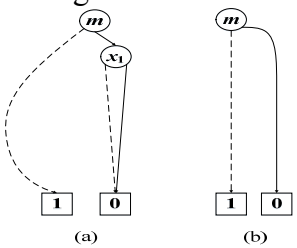

Fig.5 The diagnosis process

As shown in Figure 4, the input of the adder is x1, X2, and the output is x3. X1, X2, and X3 are both bool variables. The bool variable m in the figure is used to indicate that the system is in a normal state and when the m=false is out of order, the system is out of order. The system model can be described as (9).

if(m) x1+x2=x3 (9)

According to the encoding method described in Section 4.1, the model can be converted to a Boolean function (2), and its ROBDD model is shown in Figure 2 (c).

Assuming a certain observation can get the value of x_1 and *x3*, and *x2=true, x3=false*, diagnosis is first according to the observed reduction in Figure 2 (c) in the model, obtained in Figure 5 (a) in OBDD, then eliminate the internal variables x1, such as Figure 5 (b) in the diagnosis of solution. As shown in Figure 5 (b), any path from the root node to the 1 node in the OBDD represents the diagnostic solution. In the example, M can only reach the 1 node via the dotted line, that is, {m=false}, so the original must be faulty. When there are multiple possible paths, the A* search algorithm will seek the smallest solution of the fault element, i.e., the optimal solution.

4. Simulation and analysis

4.1 Modeling on satellite's heater control circuit

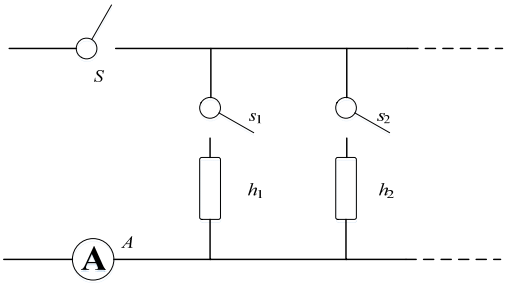

Fig.6. Satellite Thermal Control Heater Circuit

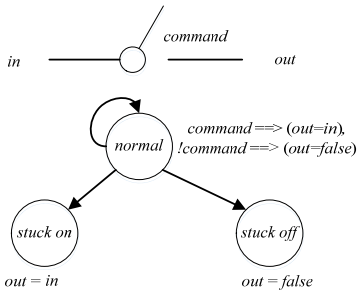

Figure 7. Switch Model

As shown in Figure 6, the experiment object is a typical satellite autonomous thermal control system. ***S*** is the safety switch for the whole thermal control circuit. When *S* is closed, the system can select to open any heater hi according to switch s_i. when ***S*** and s_i are both closed, heater h_i begins to work. On the loop, sensor ***A*** is used to measure the current. The switch, heater and current sensor are modeled as shown in Figure 7, Figure 8, and Figure 9 respectively. The interface and behavior of each component are shown in Table 3. According to the descriptions in Table 3, we first models each component separately, then models the whole system according to the connection of the components. For simplicity, this article uses only two heaters, and assumes that components are faulty and cannot be recovered.

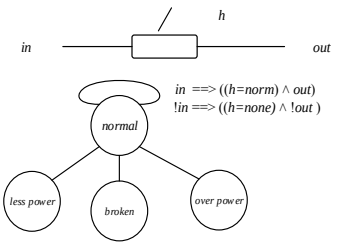

Figure 8. Heater Model

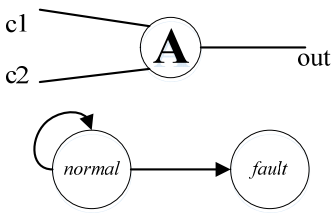

Figure 9. Current Telemetry Sensor

As described in Table 3, a model is first built for each element, and then the whole system is modeled in terms of the connection between the components.

4.2 Typical fault mode diagnosis simulation analysis

In the test scene, the thermal control system is diagnosis object. The observation values include: ***n***(the total input current), ***cs***(total switch), ***cs1***(branch no.1 switch), ***cs2***(branch no.2 switch),

h1(heater 1 heat output), ***h2***(heater 2 heat output)and ***C*** (current telemetry). The observation values are shown in Table 4.

Sets the maximum possible diagnostic solution for the two possible diagnostic results, as shown in figure 12. The "=" in the figure is in front of the system model variable, followed by the value of the variable. The name of the system mode variable is “component name.Mode".

In the first diagnosis solution, elements s1, s2, and A are normal, component h1 and h2 fault (less_power_fault). In the second diagnosis solution, elements s1 and A are normal, components s2, h1, and h2 faults (stuck_on_fault, less_power_fault, and less_power_fault).

In combination with system behavior, it is not difficult to see that both of them can give diagnosis solutions according to the observed values, and they are the two best solutions, which can prove the correctness of the method. Repeat the test 10 times, and the length of the diagnosis is as shown in Table 5. The average compile time is about 5.8s, and the average diagnosis time is about 0.0069s (the time of online diagnosis add the time of partial order extraction).

4.3 Diagnosis efficiency comparison analysis

In order to effectively verify the validity of the proposed diagnostic method, this paper compares the diagnosis efficiency between the traditional OBDD compiling diagnostic method and the classical CDA* method in the presence of two faults. The diagnosis is long and the results are shown in Table 6.

It can be seen that the efficiency of Torta’s diagnosis method is even worse than that of CDA* online inference method, because there are many internal variables in the system. But the OBDD diagnosis based on A* search is much better than the classical method proposed by Torta’s diagnosis method and the CDA* online inference diagnosis method.

Table 2. Omponent Model and Behavior of Satellite Thermal Control Circuit

Component	Input (bool)	Output (bool)	Component mode	Mode type	Description
Switch	Command, represents close or open. In, represents the existence of voltage input.	Out, represents the existence of voltage output.	Normal	Normal	When command=true, out=in. When command=false, out=false.
			Stuck on	Fault	Out=in.
			Stuck off	Fault	Out=false.
Heater	In, represents the existence of voltage input.	Out, represents the existence of voltage output. Tvar t, indicates the type of temperature increase (tvar is an enum type, its value includes none, normal, less and over, represents there is no temperature increase, normal temperature increase, temperature increases too slow and temperature increases too fast respectively.	Normal	Normal	When in=true, out=true and h=normal. When in=false, out=false and h =none.
			Broken	Fault	Out=false and h=none.
			Over power	Fault	When in=true, out=true and h=over. When in =false, out=false and h=none.
			Less power	Fault	When in=true, out=true and h=less. When in=false, out=false and h=none.
Ammeter	C1, represents the existence of current in loop 1. C2, represents the existence of current in loop 2.	Out, represents the existence of current output and the output type(current is an enum type, its value includes none, one and two, represents there is no current, one current and two current respectively.	Normal	Normal	When c1=false and c2=false, out=none. When c1=true and c2=true, out=two. Else out=one.
			Fault	Fault	Out=none.

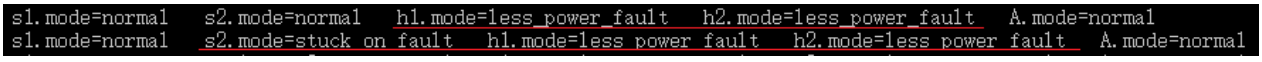

Figure 10. Diagnosis Result

Table 3.Observation variables and observations

Variables	in	cS	cs1	cs2	h1	h2	C
Observations	true	true	true	true	less	less	two

Table 4. Compile diagnostic method based on A* search time statistics

NO.		1	2	3	4	5	6	7	8	9	10	Average
Time of Compilation		5.81	5.644	5.783	5.645	5.721	5.972	5.927	5.868	5.913	5.851	5.8134
Time of Diagnosis	Restriction	0.001	0.001	0.001	0	0.001	0.001	0.001	0.001	0.001	0	0.0008
	Extraction	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.007	0.0061

Table 5.The working time of Offline-compilation and Online-reasoning

	1	2	3	4	5	6	7	8	9	10	Average
Used A* Search and OBDD	0.007	0.007	0.007	0.006	0.007	0.007	0.007	0.007	0.007	0.007	0.0069
Torta's OBDD method	0.051	0.05	0.041	0.051	0.043	0.05	0.048	0.05	0.051	0.051	0.0486
CDA* Online-reasoning	0.01	0.016	0.069	0.031	0.016	0.028	0.015	0.015	0.02	0.031	0.0251

5. Conclusion

This paper studies the fault diagnosis method of satellite system based on discrete model.

1) for satellite fault diagnosis modeling process, put forward a set of diagnostic framework, using hierarchical mode of rapid modeling system, and then transfer standard form of negation paradigm;

2) based on fault diagnosis theory of OBDD, puts forward A* search algorithm based on partial solution quickly, improve the efficiency of with the real-time on-line diagnosis;

3) control loop in satellite thermal control system as an example, a complete presentation of the whole process to verify the diagnosis, diagnosis framework and algorithm in satellite system.

Theoretical analysis and experimental results show that the proposed method can easily and quickly build the model of the system, without the artificial reasoning complex that can quickly and accurately complete fault diagnosis, laid the foundation for the rapid diagnosis of complex satellite system.

References

[1]. Cui Ziqian. The Research On Satellite Thermal Control System Diagnosis Based On Qualitative Model [D].Harbin：Harbin Institute of Technology, 2007.

[2]. B. Williams and P. Nayak, "A Model-based Approach to Reactive Self-Configuring Systems," In Proceedings of AAAI-98, 971-978, 1996.

[3]. Darwiche A, Marquis P. A knowledge compilation map[J]. Journal of Artificial Intelligence Research, 2002, 17(1): 229-264.

[4]. Torasso P, Torta G. Compact diagnoses representation in diagnostic problem solving[J]. Computational Intelligence, 2005, 21(1): 27-68.

[5]. Torta G. Compact representation of diagnoses for improving efficiency in model based diagnosis[D]. Dipartimento di Informatica, Universita di Torino, Italy, 2005.

[6]. Torasso P, Torta G. Model-based diagnosis through obdd compilation: A complexity analysis[M]//Reasoning, Action and Interaction in AI Theories and Systems. Springer Berlin Heidelberg, 2006: 287-305.

[7]. R. E. Bryant, "Symbolic Boolean manipulation with ordered binary-decision diagrams," ACM Comput. Surv., vol. 24, no. 3, pp. 293–318, 1992.

[8]. Gu Tianlong, Xu Zhoubo. Ordered Binary Decision Diagraph and Its Application. Beijing: Science Press, 2009.
[9]. S. Russell and P. Norvig, Artificial Intelligence A Modern Approach. 2013.