Focusing Strategies for Multiple Fault Diagnosis

Tsai-Ching Lu
HRL Laboratories, LLC
Malibu, CA 90265
tlu.fn@hrl.com

K. Wojtek Przytula
HRL Laboratories, LLC
Malibu, CA, 90265
wojtek@hrl.com

Abstract

Diagnosing multiple faults for a complex system is often very difficult. It requires not only a model which adequately represents the diagnostic aspect of a complex system, but also an efficient diagnostic algorithm that can generate effective test and repair recommendations. One way of developing such an efficient and effective diagnostic algorithm is to focus the computational resource on disambiguating a set of the most likely potential faults, called focus faults. In this paper, we apply decision theory to analyze strategies for selecting focus faults. We propose a decision-theoretic focusing strategy which is based on users' risk tolerances. The proposed focusing strategy has been applied to a large diagnostic model for locomotives, which has been deployed in the field. Our diagnostic experts found decision-theoretic focusing strategy useful and informative.

Introduction

A model-based diagnosis system requires a diagnostic model, which adequately represents the diagnostic aspect of a system, and an inference algorithm, which can generate a ranked list of suspect faults, as well as test and repair recommendations. Because of its succinct representation, Bayesian networks (BN) have become a popular choice for such diagnostic models (Darwiche 2000). Constructing adequate diagnostic BN models for complex systems is often laborious and time consuming. It is not until recently that researchers proposed effective methodologies for constructing diagnostic BNs with thousands of nodes. Since both exact and approximate inferences for belief updating in BN are NP-hard (Cooper 1990; Dagum & Luby 1993), it is expected that complex BN models will present computational challenges to existing BN inference algorithms. Nevertheless, by exploiting various network structure and node types, inference algorithms can handle most complex BN models efficiently.

When using Bayesian networks for multiple fault diagnosis, one usually generates test recommendations based on some form of value of information computation (Heckerman, Breese, & Rommelse 1995). Computing either utility or quasi-utility based value of information (VOI) for multiple fault diagnosis requires multiple belief updating, which implies that VOI computation is even harder than belief updating. In order to avoid the computational complexity and to support a more focused diagnosis, some diagnostic BN development environments, such as GeNIe (University of Pittsburgh) and WIN-DX (Knowledge Industries, Inc.), present to users a ranked list of faults, according to their posterior marginal probabilities given the evidence, and allow users to select a subset of faults to pursue in computation of recommended tests using VOI. We refer to the subset of pursued faults as *focus faults* and to the strategy used to determine the selection as a *focusing strategy*. VOI computation will provide a ranked list of tests based on their abilities to disambiguate the focus faults. Although manual selection of focus faults provides certain flexibility, it relies on human judgments regarding which faults are important. This leads to two main disadvantages: (1) manual fault selections is not feasible for autonomous diagnosis systems and (2) no decision support is provided to users to select the focus faults.

In this paper, we propose to apply decision theory to support the selection of focus faults. During system troubleshooting process the potential faults can be classified into four categories: (1) *committed faults*, which users are committed to fix, (2) *focus faults*, which users need to pursue, (3) *depleted faults*, which are of interest but are not prominent enough to be pursued, (4) *discarded faults*, which fall beyond users' interest. The decision of classifying a potential fault into one of these four categories can influence the quality and time needed for troubleshooting. For the classification, we could use utilities as suggested by decision theory; instead we are proposing to use zero-one loss function, which is less demanding in elicitation and computation. We develop a decision-theoretic focusing strategy to assist users in classifying a fault and report the probability of errors for each iteration of fault selections. We implement our focusing strategy in our inference engine and deploy it with a diagnostic system for diesel locomotives, in which a diagnostic Bayesian network consisting of 2,147 nodes and 3,650 arcs with custom layered structure and custom node type is used to represent the problem (Lu & Przytula 2005). Our diagnostic experts found not only our decision-theoretic focusing strategy useful but also the report on the probability of errors informative.

Diagnostic Bayesian Networks

Fault diagnosis is basically a process of identifying *causes* of system defects by observing the manifested *effects*. Different from single fault diagnosis, which assumes that only one fault is presented in a defective system, multiple fault diagnosis admits the possibility that more than one fault could occur when a system is defective. Many different knowledge representations have been used to support multiple fault diagnosis (de Kleer & Williams 1987; de Kleer 1991). In this paper, we represent our diagnostic knowledge in Bayesian networks; however, the focusing strategies presented in this paper are not limited to a particular knowledge representation.

A Bayesian network (BN) describes a joint probability distribution over a set of nodes (random variables) in a directed acyclic graph. To represent diagnostic knowledge in BN, we classify each node into one of following categories: target, observation, and auxiliary. A target node usually represents a diagnostic interest (e.g., the health status of a fuel injector). A target node has at least one target state, representing a failure mode (fault) of a component (e.g., a state "plugged" as a failure mode of a fuel injector), and at least one non-target state, representing a normal operational mode of a component (e.g., a state "ok" for an operational fuel injector). An observation node usually represents a symptom (e.g., observing an excessive smoking in engine exhaust), an built-in error message (e.g., the status of a power supply which is monitored by a feedback signal), or a test (e.g., measuring the voltage of a battery). An error message based observation is normally recorded in an archive when it obtains an abnormal state (e.g. power supply status is failed). When an error message, which is continuously monitored by a signal, is not recorded in an archive, one could assume that the error message is in its *default* ok state. This is to account for unreported observations (Peot & Shachter 1998). A node which is neither a target nor an observation is classified as an auxiliary node, which is usually used to represent intermediate relations between targets and observations. An observation node is further annotated with a Boolean flag, *ranked*, to specify whether a node will be ranked in the VOI computation. We normally annotate a test, but not an error message or a symptom, as *ranked*, since the states of symptoms and error messages are usually available before a diagnostic session is started and do not need to be recommended. We call such an annotated Bayesian network a diagnostic Bayesian network (dBN).

Troubleshooting Procedure

Figure 1 illustrates steps involved in a troubleshooting procedure, which include selection of faults to focus on and selection of next test to perform:

1. Instantiate the initial set of observations, such as error messages or reported symptoms;
2. Compute posterior probabilities of faults and generate a ranked list of faults based on their posterior probabilities;
3. Check if available diagnostic information is sufficient to perform repairs; if yes, stop to repair; otherwise, continue;

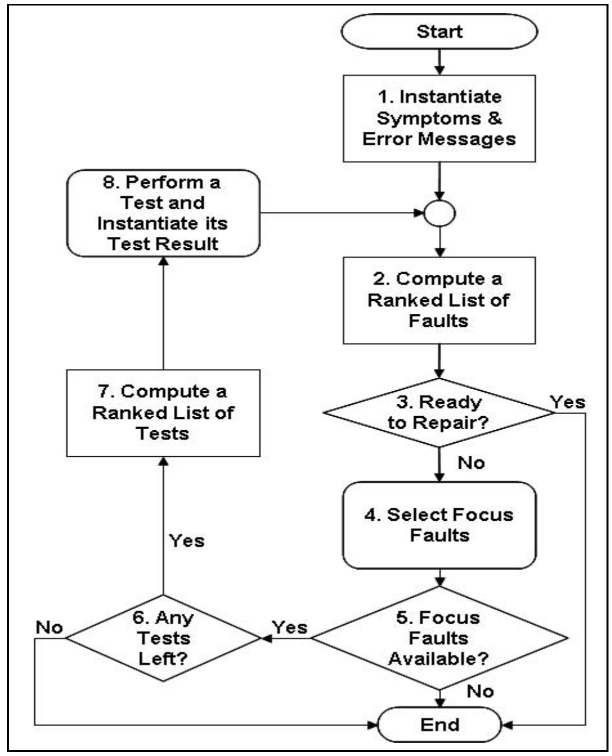

Figure 1: A procedure for multiple faults diagnosis with a diagnostic Bayesian network.

4. Select a set of focus faults; '
5. Check if there are available focus faults, if yes, continue; otherwise, stop.
6. Check if there are still unperformed tests, if yes, continue; otherwise, stop.
7. Compute the VOI for all unperformed tests relative to the selected focus faults and generate a ranked list of tests based on their VOI;
8. Perform one of the recommended tests and instantiate its test result;
9. Go to Step 2.

Notice that observation instantiations in Step 1 and 8 constitute on input to our diagnostic system, which could be either provided manually by users or automatically loaded by other programs. Step 2 produces a ranked list of faults based on the posterior probabilities of faults computed by standard belief updating in BN.

In the following section, we will first outline the value of information computation for multiple faults, which is used in Step 7 to generate a ranked list of tests. We will then present our focusing strategies on providing decision support for selecting the set of focus faults in Step 4.

Value of Information

The value of information is a measure for quantifying the value of obtaining an item of information (e.g., result of a test) for our decision problem (e.g., differentiating a set of faults) (Jensen 2001). It starts with defining a value function which maps the probability distribution of a hypothesis into a real value: $V(P(H|\mathbf{e})) : [0;1]^{|H|} \rightarrow \mathbb{R}$, where H is a hypothesis with $|H|$ number of mutually exclusive states and $\mathbf{e}$ is the set of evidences. The expected value for performing a test T is

$$EV(T) = \sum_{t \in T} V(P(H|\mathbf{e}, t))P(t), \quad (1)$$

where t is a result of the test T. The expected benefit for performing a test T is

$$EB(T) = EV(T) - V(P(H|\mathbf{e})). \quad (2)$$

Decision theory recommends using utility function as the value function. In situations where utility function is hard to elicit, one can use quasi-utility value functions (Glasziou & Hilden 1989). In this paper, we will use entropy function as our value function:[1]

$$V(P(H|\mathbf{e})) \triangleq \mathcal{H}(H|\mathbf{e}) = -\sum_{\mathbf{h} \in H} P(h|\mathbf{e}) \log_2 P(h|\mathbf{e}), \quad (3)$$

where h is a state of the hypothesis H, and the expected benefit as

$$EB(T) = \mathcal{H}(H|T, \mathbf{e}) - \mathcal{H}(H|\mathbf{e}) = I(H; T|\mathbf{e}), \quad (4)$$

where $I(H; T|\mathbf{e})$ is the mutual information between H and T. In order to rank different hypotheses using $EB(T)$, we will normalize the expected benefit by $\mathcal{H}(H|\mathbf{e})$ and define the value of information of performing a test T for a hypothesis H given evidences $\mathbf{e}$ as

$$\mathsf{VOI}(H, T|\mathbf{e}) = \frac{EB(T)}{\mathcal{H}(H|\mathbf{e})} - \alpha C(T), \quad (5)$$

where $C(T)$ is the cost of performing the test T and α is a scaling ratio[2].

When performing the single fault diagnosis, we are interested in differentiating a selected focus fault against the rest. In other words, we define a hypothesis variable H as $H = \{f, \overline{f}\}$, where f is a target state (fault) of a target variable and $\overline{f}$ is the negation of the fault f, i.e., the rest of the states of the target variable.

Recall that a dBN may consist of many target nodes and each target node may have more than one target states. Consider for example a dBN which includes, in addition to other node types, two target nodes F_1 and F_2 with states $\{f_{11}, f_{12}, ok\}$ and $\{f_{21}, ok\}$ respectively. This dBN allows us to investigate three single fault hypotheses $H_1 = \{f_{11}, \overline{f}_{11}\}, H_2 = \{f_{12}, \overline{f}_{12}\}$, and $H_3 = \{f_{21}, \overline{f}_{21}\}$. If we decide to pursue the fault f_{11}, i.e., selecting the f_{11} as the focus fault from the ranked list of faults, we will compute $\mathsf{VOI}(H_1, T|\mathbf{e})$ for each unperformed test T to generate a ranked list of tests, i.e., ranking the values of VOI of all unperformed tests.

[1]Readers are recommended to read (Glasziou & Hilden 1989) for the appropriate use of different quasi-utility functions.

[2]Please note that we use the linear transformation as an example, however, one can have more elaborated transformation function.

Table 1: Configurations of (F_1, F_2).

$c_1 = (f_{11}, f_{21})$	$c_2 = (f_{12}, f_{21})$	$c_3 = (ok, f_{21})$
$c_4 = (f_{11}, ok)$	$c_5 = (f_{12}, ok)$	$c_6 = (ok, ok)$

When performing the multiple fault diagnosis, we first select a set of focus faults that we wish to pursue. There are many ways to construct a hypothesis variable H for the selected focus faults F (Jagt 2002). We consider three common ways of constructing a hypothesis variable: conjunction ($\wedge$), disjunction ($\vee$), and unique existence ($\oplus$). Continuing on our example, there are six configurations (c_i) for our two target variables (Table 1). Assume that we select f_{11} and f_{21} as our focus faults F. If we are interested in differentiating $f_{11} \wedge f_{21}$ from the rest, we will repartition the configuration of (F_1, F_2) to derive the hypothesis variable $H = \{h_1, h_2\}$, where $h_1 = c_1$ and $h_2 = \{c_2, \ldots, c_6\}$. If we are interested in differentiating $f_{11} \vee f_{21}$ from the rest, we will have $H = \{h_1, \ldots, h_5\}$, where $h_i = c_i$ for $i = 1, \ldots, 4$, and $h_5 = \{c_5, c_6\}$. If we are interested in differentiating $f_{11} \oplus f_{21}$ from the rest, we will have $H = \{h_1, h_2, h_3\}$ where $h_1 = c_3, h2 = c_4$, and $h_3 = \{c_1, c_2, c_5, c_6\}$.

In this paper, we will use the disjunction ($\vee$) to compose hypothesis states, i.e., each configuration which is satisfied with the disjunction of the selected focus faults will become a state of H and those unsatisfied configurations will be grouped into one state of H. We will then compute $\mathsf{VOI}(H, T|\mathbf{e})$ for each unperformed test T to generate a ranked list of tests. In other words, we are ranking the values of VOI for all available tests on differentiating the states of H derived from the disjunction of focus faults.

Unlike computing VOI for single fault diagnosis, where all required probabilities of H are available from standard belief updating in BN, computing VOI for multiple fault diagnosis require us to derive the probabilities of H from the joint probabilities of the selected focus variables, which are not directly available in standard belief updating in BN. Although there are methods for computing $P(H|\mathbf{e})$, where H is technically a set of target variables F in dBN (Xu 1995; Smith 2001), it soon becomes intractable since the number of configurations of F grows exponentially. Instead, we will approximate $P(H|\mathbf{e})$ by the marginal probabilities of F. In other words, we assume that target variables in F are independent. Continuing on our example, we will have $P(F|\mathbf{e}) = P(F_1|\mathbf{e})P(F_2|\mathbf{e})$ to derive $P(H|\mathbf{e})$.

Focusing Strategies

A focusing strategy is used to decide which fault will be included in the set of focus faults. The set of focus faults is then used to form the states of the hypothesis variable for VOI computation. Since the number of the states of the hypothesis variable grows exponentially in the number of selected focus faults, it is impractical to include all faults as focus faults when diagnosing a complex system. On the other hand, applying an ad-hoc strategy, such as using a predetermined small number of focus faults, is hard to generalize to different kinds of system failures.

For example, in model-based diagnostic systems (de Kleer & Willams 1989; de Kleer 1991), a *diagnosis*, also called a *candidate*, is a conjunction of faults. They focus the diagnostic reasoning on the subset of diagnoses (called *leading diagnoses*) that satisfy the following conditions:

- There are no more than k_1 (usually $k_1 = 5$) leading diagnoses.
- Candidates with probability less than $\frac{1}{k_2}$th usually $k_2 = 100$) of the best diagnosis are not considered.
- The diagnoses need not include more than k_3 (usually $k_3 = .75$) of the total probability mass of the candidates.

We could adapt deKleer's selection of leading diagnoses as focus fault selections. However, we are still lacking a way to analyze the consequence of their focusing strategy.

To evaluate focusing strategies, we apply decision theory. We first assume that each decision of selecting a fault as focus can be made independently[3]. Let $\lambda_i(f_{ij}|f_{ik})$ be the loss function associated with selecting a target (fault) state f_{ij} of a target node F_i as a focus fault, when actually a state f_{ik} of F_i should be selected[4]. The expected loss (risk) of selecting the fault f_{ij} as focus is defined as follows:

$$R_i(f_{ij}|\mathbf{e}) = \sum_{ik} \lambda_i(f_{ij}|f_{ik})P(f_{ik}|\mathbf{e}). \qquad (6)$$

The optimal decision f^*_{ij} is derived from minimizing the risk $R_i(f_{ij}|\mathbf{e})$:

$$f^*_{ij} = \operatorname*{argmin}_{ij} \sum_{ik} \lambda_i(f_{ij}|f_{ik})P(f_{ik}|\mathbf{e}). \qquad (7)$$

Assume the linear additivity among the risks, the total risk of selecting a set of focus faults F_l is defined as follows:

$$R(F_l) = \sum_{i \in l} \omega_i R_i(f_{ij}|\mathbf{e}), \qquad (8)$$

where ω_i is the weighting factor for R_i. We can compute $R(F_l)$ for any non-empty set of focus faults F_l derived from different focusing strategies. However, the optimal strategy is the one minimizing $R(F_l)$:

$$F^*_l = \operatorname*{argmin}_{l} \sum_{i \in l} \omega_i R_i(f_{ij}|\mathbf{e}). \qquad (9)$$

When loss functions $\lambda_i(f_{ij}|f_{ik})$ and weight factors ω_i are hard to obtain, we may assume the zero-one loss function ($\lambda_i(f_{ij}|f_{ik}) = 1$, if $f_{ij} \neq f_{ik}$; $\lambda_i(f_{ij}|f_{ik}) = 0$, otherwise) and the equal weighting factor ($\omega_i = 1$) for all i. Consequently, the risk for deciding on f_{ij} reduces to the probability of error, i.e., $R_i(f_{ij}|\mathbf{e}) = 1 - P(f_{ij}|\mathbf{e})$.

To minimize the risk, we will choose f^*_{ij} with the maximum $P(f_{ij}|\mathbf{e})$ among all j. In other words, the probability of correctness is $P(f^*_{ij}|\mathbf{e})$. Since all faults in F_l are assumed to be jointly independent, we will have the total probability of correctness as $\prod_i P(f^*_{ij}|\mathbf{e})$ and the total probability of error as $1 - \prod_i P(f^*_{ij}|\mathbf{e})$. These assumptions will lead us to the optimal F^*_l, which contains only one fault with the maximum $P(f^*_{ij}|\mathbf{e})$. In general, we have derived a decision-theoretic framework to evaluate the total risk for any F_l as in Equation 8.

In practice of multiple fault diagnosis, it is convenient to classify faults into four categories: (1) *committed faults*, which users are committed to fix, (2) *focus faults*, which users need to pursue, (3) *depleted faults*, which are of interest but are not prominent enough to be pursued, (4) *discarded faults*, which fall beyond users' interest. One way of classifying a fault into one of these categories is to define the probability thresholds: committed fault threshold (p_c), focus fault threshold (p_f), and discarded fault threshold (p_d), such that a fault f_{ij} is considered committed ($p_c \leq P(f_{ij}|\mathbf{e}) \leq 1$), focus ($p_f \leq P(f_{ij}|\mathbf{e}) < p_c$), depleted ($p_d \leq P(f_{ij}|\mathbf{e}) < p_f$), or discarded ($0 \leq P(f_{ij}|\mathbf{e}) < p_d$)[5]. Ideally, we can define separately the set of probability thresholds for each fault, because we may see the risk for each fault differently; for example, the committed fault threshold, p_c, for a mission critical fault will be smaller than the one for a fault of an auxiliary component. However, when such information is hard to obtain, we can define one set of thresholds for all faults. Once we have classified all the faults into their categories, we can compute the total risks for each category of faults so that users are informed about the consequences of their decisions.

Instead of specifying probability thresholds, users can specify the model-wide total risk thresholds for committed (tr_c), focus (tr_f), and depleted (tr_{dp}) faults. Given a list of faults F ranked by their $P(f_{ij}|\mathbf{e})$ in descending order, we can classify each fault f_{ij} into its category according to the procedure $\mathit{ClassifyFaults}(F, tr_c, tr_f, tr_{dp})$ outlined in Figure 2, where we assume zero-one loss functions and equal weighting factors. The procedure takes a ranked list of faults F with their $P(f_{ij}|\mathbf{e})$ as inputs and outputs a *partition* of F into: committed faults (F_c), focus faults (F_f), depleted faults (F_{dp}), and discarded faults (F_{di}). The procedure loops through the list of faults in F (Line 5-18). For each fault, the procedure starts with computing the accumulated risk (probability of errors) of including the fault (Line 6-7). If the accumulated risk is smaller than the total risk threshold for the current fault category, the fault is added into the category (Line 9). Otherwise, the procedure checks if it has reached the last category (Line 14) , if yes, all the remaining faults will be added into discarded faults (Line 19-21); if not, the procedure advances to the next fault category (Line 11).

Evaluation

To test the performance of different focusing strategies, we conducted experiments on two proprietary networks, tcc4g and emdec6h, constructed by HRL for diagnosing two subsystems of locomotives. In tcc4g network, there are 36 target

[3] If this assumption is not valid, we need to consider the utility (loss) function over the dependent faults.

[4] f_{ik} could be an ok state.

[5] These thresholds are in fact the probabilities of correctness of classifying f_{ij} into one of the categories, if we use the zero-one loss function and the equal weighting factor. In other words, one can use risk thresholds instead of probability thresholds, if information is available.

Procedure $\mathit{ClassifyFaults}(F, tr_c, tr_f, tr_{dp})$

Input: A list of faults F ranked by their $P(f_{ij}|\mathbf{e})$ in descending order, a total rsik threshold for committed faults tr_c, a total risk threshold for focus faults tr_f, and a total risk threshold for depleted faults tr_{dp},

Output: F_c: the set of committed faults; F_f: the set of focus faults; F_{dp}: the set of depleted faults, and F_{di}: the set of discarded faults.

```
k := 0; // index of TR and SF
tpc := 1.0; // total probability of correctness
TR := [tr_c, tr_f, tr_dp];
SF := [F_c, F_f, F_dp];
for (ij := 0; ij < |F|; ij++)
  tpc := tpc * P(f_ij|e);
  tpe := 1.0 - tpc; // total probability of error
  if (tpe < TR[k])
    SF[k] := SF[k] ∪ f_ij;
  else
    k++; // move to the next fault category
    tr := 1.0; // reset total risk
    ij --; // retreat fault index
    if k > 2
      break; // break for loop
    end if
  end if
end for
for (;ij < |F|; ij++)
  F_di := F_di ∪ f_ij;
end for
```

Figure 2: A procedure for classifying a list of faults into their fault categories.

nodes and 69 observations (29 error messages and 40 tests). In emdec6h network, there are 47 target nodes and 117 observations (53 error messages and 54 observations). We decided not to run our performance evaluation on any of the publicly available BN, since we have no domain knowledge of annotating those networks into dBN.

For each network, we randomly generate n diagnostic cases and run three focusing strategies (deKleer, probability-threshold, risk-threshold) on them. In each case, we first generate the "real" target states by randomly selecting 10 percent of target nodes to fail, and each of which is randomly assigned with one of its target states. The rest of target nodes are randomly assigned with one of their non-target states.[6] We then plug in these "real" target states into the network and update the belief for observations. We generate the "real" states of observations by casting the states of the modes of their posterior distributions. These "real" states of observations will be used in diagnosis as simulated test results or the initial states of error messages.

Once we generate the "real" states for all the cases, we start the diagnosis procedure as outlined in Figure 1 to generate "diagnosed" states. For each case, we instantiate all error messages into their "real" states in dBN as in Step 1. The diagnostic procedure will perform each step iteratively until there is no test to perform (Step 6), no focus faults available (Step 5), or ready to repair (Step 3). We assume that we are ready to repair, when we perform tests up to three times the number of failed targets. Once the diagnostic procedure is stopped, we record the "diagnosed" state of a target node by casting the mode of its posterior distribution.

For each case, we compute the scores of sensitivity (Sen.), specificity (Spe.), and accuracy (Acc.) to account for the quality of diagnosis. Recall that each target node has a "real" state and "diagnosed" state in our simulation. If the "real" state is a target (non-target) state and its diagnosed state is the same target (non-target) state, we count it as one of the correctly diagnosed defects (non-defects). The sensitivity (specificity) is the ratio of correctly diagnosed defects (non-defects). The accuracy is the ratio of overall correct diagnosis. For each combination of simulation parameters, we further compute the mean and standard deviation of sensitivity, specificity, and accuracy scores for each focusing strategy.

To ensure that the number of randomly generated diagnostic cases does not bias our evaluation results, our exploratory experiments indicate that 5000 cases seem to be sufficient for both tcc4g and emdech6h. Hence we will only report our experiment results of 5000 cases with some fixed parameters for each focusing strategy. For de Kleer's focusing strategy, we fix $k_1 = 5$ and $k_3 = .75$ as suggested in de Kleer & Willams (1989) and vary the $k2 = 1000, 100, 10$. For probability-threshold strategy, we fix $p_c = 0.9$ and $p_d = 0.00001$ and vary the $p_f = 0.001, 0.01, 0.1$ with respect to the variation of k_2. For risk-threshold strategy, we fix $tr_c = 0.271$ and $tr_{dp} = 0.9999$ corresponding to the selected p_c and p_d, and vary the $tr_f = 1, 0.9999, 0.999$ with respect to the variation of p_f. We further introduced the k_1 parameter into both decision-theoretic focusing strategies, i.e., both strategies will not pursue more than $k_1 = 5$ targets, to reduce the built-in bias in de Kleer's focusing strategy. The results of evaluation for tcc4g and emdec6h are shown in Table 2 and 3. We did see both probability-threshold and risk-threshold strategies perform slightly better than de Kleer's strategy.

Conclusion

The major contributions of our paper are: (1) introduction of the concept of fault categories, (2) application of decision theory to analyze the problem of focus fault selections, (3) development of an informative decision-theoretic focusing strategy, (4) reporting experiment results of evaluating different focusing strategies, and (5) deploying the implementation of our focusing strategies into the field.

Although focusing is not novel in diagnosis, applying decision theory to analyze the consequence of focusing is

[6]This assignment scheme does not lead to inconsistent target states because target nodes in both tcc4g and emdec6h are jointly independent. When target nodes are dependent in a dBN, we may use the forward sampling to generate consistent states.

Table 2: Evaluation Results for TCC4G: Average Sensitivity, Specificity, and Accuracy - 5000 cases.

de Kleer's				Probability-Threshold				Risk-Threshold			
k_2	Sen.	Spe.	Acc.	p_f	Sen.	Spe.	Acc.	tr_f	Sen.	Spe.	Acc.
0.001	0.6654	0.9940	0.9666	0.001	0.6733	0.9940	0.9673	1	0.6670	0.9939	0.9667
0.01	0.6313	0.9941	0.9639	0.01	0.6378	0.9941	0.9644	0.9999	0.6439	0.9937	0.9646
0.1	0.5880	0.9927	0.9590	0.1	0.5709	0.9929	0.9577	0.999	0.6255	0.9936	0.9630

Table 3: Evaluation Results for EMDEC6H: Average Sensitivity, Specificity, and Accuracy - 5000 cases.

de Kleer's				Probability-Threshold				Risk-Threshold			
k_2	Sen.	Spe.	Acc.	p_f	Sen.	Spe.	Acc.	tr_f	Sen.	Spe.	Acc.
0.001	0.6579	0.9985	0.9695	0.001	0.6742	0.9982	0.9707	1	0.6752	0.9981	0.9706
0.01	0.6469	0.9985	0.9686	0.01	0.6800	0.9983	0.9712	0.9999	0.6842	0.9980	0.9713
0.1	0.6342	0.9984	0.9674	0.1	0.5711	0.9979	0.9616	0.999	0.6834	0.9980	0.9712

novel. Such analysis leads us to the development of informative focusing strategies. Furthermore, the concept of fault categories (especially the category of committed fault) is not found in the literature to the best of our knowledge. This makes our focusing strategies potentially perform better than the strategy adapted from de Kleer's (1989), because their strategy might wrongly focus on differentiating committed faults.

When applying focusing strategies to a particular problem, we recommend users to validate assumptions in our decision-theoretic focusing strategies. For example, one may want to use different loos function for each individual fault, if dBN include dependent faults or zero-one loss function is not appropriate. One may want to use different total risk function, if linear additivity is not valid. Furthermore, one may consider using different hypothesis formation operator in VOI computation. We also recommend users to fine-tune the parameters of selected focusing strategy with respect to the problem.

In the future, we would like to extend our evaluation methods to attribute the diagnosibility to its sources. In addition to focusing strategies, there could be many other factors affecting our diagnostic scores. For example, it could be the case that we have a perfect model describing the system, but the system does not provide enough observability to separate the faults. It could also be the case that we did not model the system correctly into dBN. We are currently investigating different evaluation methods.

Acknowledgments

The core of our inference engine is based on the SMILE reasoning engine by the Decision Systems Laboratory, University of Pittsburgh (http://dsl.sis.pitt.edu). We thank the reviewers for their comments, which helped us improve our paper.

References

Cooper, G. F. 1990. The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. *Artificial Intelligence* 42(2–3):393–405.

Dagum, P., and Luby, M. 1993. Approximating Probabilistic Inference in Bayesian Belief Networks is NP-hard. *Artificial Intelligence* 60(1):141–153.

Darwiche, A. 2000. Model-Based Diagnosis under Real-World Constraints. *AI Magzazine* 21(2):57–73.

de Kleer, J., and Willams, B. C. 1989. Diagnosis With Behavioral Modes. In *Proceedings IJCAI-89*, 104–109.

de Kleer, J., and Williams, B. C. 1987. Diagnosing Multiple Faults. *Artificial Intelligence* 32(1):97–130.

de Kleer, J. 1991. Focusing on Probable Diagnoses. In *Proceedings of the 9th National Conference on Artificial Intelligence, AAAI–91*, 842–848. Los Angeles, CA: American Association for Artificial Intelligence.

Glasziou, P., and Hilden, J. 1989. Test Selection Measures. *Medical Decision Making* 9(2):133–141.

Heckerman, D.; Breese, J. S.; and Rommelse, K. 1995. Decision-Theoretic Troubleshooting. *Communications of the ACM* 38(3):49–57.

Jagt, R. M. 2002. Support for Multiple Cause Diagnosis with Bayesian Networks. Master's thesis, Delft University of Technology.

Jensen, F. V. 2001. *Bayesian Networks and Decision Graphs*. New York, NY: Springer.

Lu, T.-C., and Przytula, K. W. 2005. Methodology and Tools for Rapid Development of Large Bayesian Networks. In *Proceedings of the 16th International Workshop on Principles of Diagnosis (DX-05)*, 107–112.

Peot, M., and Shachter, R. 1998. Learning From What You Don't Observe. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–98)*, 439–446. San Francisco, CA: Morgan Kaufmann Publishers.

Smith, D. 2001. The Efficient Propagation of Arbitrary Subsets of Beliefs in Discrete-Valued Bayesian Belief Networks. In *Proceeding of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTATS 2001)*.

Xu, H. 1995. Computing Marginals for Arbitrary Subsets from Marginal Representation in Markov Trees. *Artificial Intelligence* 74(1):177–189.