THE MULTICRITERION NATURE OF DIAGNOSIS

William R. Simpson and John W. Sheppard
Advanced Research and Development

ARINC Research Corporation
Annapolis, MD

ABSTRACT

Developing diagnostic strategies is a difficult task, complicated by the multicriterion nature of diagnosis. Sequencing and evaluating tests, as well as performing subsequent diagnosis, are difficult in and of themselves, and achieving accurate diagnosis becomes an ambitious goal. Several new diagnostic tools can compute accurate decision trees using many types of assumptions, thus increasing analytical power. This increased power allows us to consider the optimization of diagnostic strategies by following several different approaches. We can also compare the various tools available. This paper provides a framework for developing diagnostic strategies or decision trees according to multiple criteria, and these criteria and decision trees fall into three broad categories—preprocessing criteria, optimization criteria, and postprocessing criteria. We also discuss guidelines for comparing diagnostic strategies for differing criteria. Interactions between many of these criteria make comparisons difficult or invalid; therefore, we provide rules for equivalence as prerequisites for making valid comparisons.

INTRODUCTION

The general problem of diagnosis is extremely complex, and to successfully perform diagnosis, a basic strategy must be imposed. A diagnostic strategy comprises performing diagnostic tests, evaluating outcomes, and undertaking repair on a system with a certain set of symptoms and history. In the recent past, widely divergent diagnostic strategies were applied to systems having the same set of symptoms and maintenance history. The variety was acceptable because it was not clear whether any particular strategy was optimal. Unfortunately, generating optimal diagnostic strategies falls in the class of NP-complete problems,[1] thus computing globally optimal trees is impractical for any moderate or large size problems. The problem of optimal design of diagnostic decision trees was addressed as early as 1960 by R. A. Johnson.[2] Johnson proposed constructing diagnostic decision trees by using the information gained per dollar expended per test for each successive diagnostic step. Subsequently, additional work in applying information theory to diagnostics has been pursued.[3,4]

All current approaches to optimal diagnostic strategies apply local search or approximate global search. Given the constraints imposed by local search, we are still interested in providing efficient diagnostic strategies, most of which today are represented by the development of decision trees. Therefore, we seek to apply several optimization techniques to continue to improve resulting trees.

Diagnosis is truly a multicriterion problem. By multicriterion we mean that for any given set of circumstances, a number of factors will affect developing diagnostic strategies. Because local search fails to provide globally optimal solutions, comparing alternatives tends to be the best and clearest way to evaluate the worth of a methodology. Applying several different criteria, however, makes comparing approaches especially difficult if we are not precise in our definitions and do not control each of the optimization factors.

The approaches to diagnosis presented in this paper have evolved over that last 12 years through more than 250 real-world applications using a set of tools called STAMP® and POINTER™.[5] Although developing these tools has shaped our thinking, the applications themselves have driven the development of the tools, and the optimization processes presented here should apply, in one form or another, to all diagnostic approaches. The context of diagnosis has led to a number of different optimization problems that occur at various stages of diagnostic strategy development and relate to three broad categories of criteria: preprocessing, optimization, and postprocessing.

Efficiency can be represented in a number of ways. For example, one interpretation of efficiency is "best" use of resources. This leaves "best" to be determined; best can include all of the factors we discuss here and, thus, any method that ignores any of these may be inefficient. For purposes of this paper, we assume efficiency is achieved when we optimize using a cost factor such as time. Thus a diagnostic strategy that isolates in an expected time of 5.0 minutes is more efficient than one that isolates in an expected time of 10.0 minutes, all other factors being equal. The last point, as we shall see is very important.

PREPROCESSING CRITERIA

Preprocessing criteria include those factors that are undertaken before formulating diagnostic strategies to condition the problem to match requirements. As we shall see, these may or may not affect the efficiency of the resulting strategy. Three preprocessing criteria discussed here are minimizing the set of tests performed, developing symptom-based strategies, and developing false alarm tolerant strategies.

MINIMUM TEST SET

In assessing the value of a set of tests to perform diagnosis, one major objective may be to minimize the number of tests used in the tree. Because each test must be specified, developed, documented, and validated, reducing the number of tests can significantly reduce cost. Frequently, *ad hoc* methods for developing diagnostic strategies result in overtesting. Determining the minimum test set should reduce development cost, but may increase life- cycle cost. Minimum test strategies may be less efficient. Test cases on a moderate-size system indicate that as much as 10% loss in efficiency (as measured in time to test or number of tests to achieved isolation) may occur. Some systems examined have no degradation.

The degree of development savings or life-cycle-cost losses depend on several factors, and both types of strategies need to be developed to project these costs. In a test case, based upon an environmental control system with more than 400 fault isolation conclusions and 300 tests,[8] the minimum test set was 175 tests and required 9.77 tests on average for diagnosis. When the full test set was available, 217 of the these tests were used with an average of 9.66 tests used for diagnosis. The up-front savings of not developing 43 tests may well be worth the difference in isolation efficiency.

The approach to computing the minimum test set depends on the data structure and underlying optimization philosophy. In STAMP, this is a selectable feature, and the analysis simply evaluates whether eliminating a test will change isolation capabilities. (Note that the tests are ranked by cost first.) If the answer is no, the test is eliminated, and the question is asked of the next test. If the answer is yes, the test is not eliminated, and the question is asked of the next test. Precise algorithms for this analysis are given in reference 5.

SYMPTOM-BASED DIAGNOSIS

For systems that exhibit well-defined symptoms, a large reduction in diagnostic effort can be achieved by providing detailed strategies for each symptom. We can usually fault isolate a system more efficiently by beginning with a symptom. Beginning with the symptom actually eliminates many possibilities while pointing to others, thus reducing the search space tremendously. Sometimes, a symptom is so precise that it points directly to the answer and eliminates diagnosis completely.

This type of diagnostic savings is usually accompanied by a sizable increase in up-front diagnostic development costs because we must generate, document, and verify more strategies, including one to use when no defined symptoms are present. These costs can be avoided by using a reasoner in devices that can accommodate symptoms and adjust strategies. When developing portable maintenance aids, updating the symptoms should be possible in the field.

FALSE ALARM TOLERANT STRATEGIES

False alarms can become a severe problem for many systems that are state-of-the-art or that perform diagnostics with inaccurate test equipment. A number of other factors also influence the significance of false alarms.[6] Usually three approaches are possible for compensating for false alarms:

- Open test tolerances (this runs the risk of missed detections)
- Repeat poll (allows transients to die out, but also increases the chance of missed detections because intermittent indications of real failures may go away)
- Seek confirmation of detection by performing additional tests

Such approaches, of course, decrease efficiency because additional testing is being performed. One approach to computing what additional testing is required is by hypothesis verification.[5] Hypothesis verification methods chose an additional test based upon its ability to verify or deny the diagnostic conclusion. One such approach is dealt with in detail in reference 5.

OPTIMIZATION CRITERIA

Optimization criteria apply, in the mathematical sense, after preprocessing is completed. We can apply a cost function to our data and try to minimize the value of that cost function. In diagnosis, we are concerned with three basic types of cost parameters:

- Information content of a test
- Cost directly affecting tests
- Costs indirectly affecting tests

Information content of tests is addressed by using information measures for decision processes. Reference 5 provides an information entropy based approach.

After preprocessing the criteria, we can derive diagnostic strategies that account for both direct and indirect cost parameters affecting diagnosis:

- *Direct parameters* are values assigned to those factors that are tied to testing.
- *Indirect parameters* are values assigned to factors tied to conclusions.

Examples of direct parameters include test cost, test time, and skill level. In general, the larger the value the less desirable the test. For such factors, we can define a simple, normalized weight given by:

$$w_i = \frac{\kappa_1}{d_i}$$

where

w_i = weight applied to the ith test
d_i = the direct cost parameter (inverse proportionality)
κ_1 = a normalizer defined as:

$$\kappa_1 = \left(\sum_{i=1}^{|T|} \left(\frac{1}{d_i} \right) \right)^{-1}$$

where

$|T|$ = the size of the set of all tests being considered
w_i = value always lying between 0 and 1, increasing in value for decreasing values of d_i.

Indirect parameters are so named because they have an indirect effect on the tests. Examples include failure rate, safety, and criticality. A higher value usually indicates a more desirable conclusion to consider, thus tests that examine conclusions with high values are more desirable. Because the parameters are indirect, we must first compute a contribution of conclusion weights on each test as:

$$\chi_i = \sum_{j=1}^{|F|} a_j$$

where

$$a_j = \begin{cases} e_j, & \text{if the test examines conclusion } j \\ 0, & \text{otherwise} \end{cases}$$

χ_i = contribution of conclusion weights on the ith test
a_j = intermediate value (either the indirect e_j or 0) of the jth conclusion
$|F|$ = the set of all appropriate conclusions

The actual weight can be then computed as:

$$w_i = \kappa_2 \chi$$

where

w_i = weight applied to the ith test
χ_i = intermediate value (direct proportionality)
κ_2 = a normalizer defined as:

$$\kappa_2 = \left(\sum_{i=1}^{|T|} \tau_k \right)^{-1}$$

A third parameter, I_k, relates the information content of a test and must be computed by a method consistent with the modeling approach. In STAMP, we use an entropy measure.[5] The value of the ith test at any time is given by:

$$val_i = \prod_{j=1}^{n} (w_{ji})^{\alpha_j}$$

where

Product = performance over n weighting factors (including information)

α_j = emphasis exponent applied to factor j

ISOLATION SEQUENCE CUTOFFS

In conjunction with the indirect parametric weighting criteria, a number of cutoff criteria may be applied. For example, time may be limited, and we may need to terminate diagnosis after time expires. We apply these cutoffs to isolation strategies with any weighting schema. Other cutoffs may include acceptable ambiguity size or numbers of steps in a sequence. In fact all may be active in any diagnostic strategy (i.e., quit after 22 minutes or 7 steps or when the ambiguity is reduced to n or fewer replaceable units). Any premature termination of a diagnostic sequence will cause a change in the resolution. Combining isolation sequence cutoffs and multicriterion optimization makes comparing competing techniques almost impossible because of the resolution problem to be discussed later.

POSTPROCESSING CRITERIA

Once we achieve a complete diagnostic strategy, we can begin to ask trade-off questions such as "What if I were willing to sacrifice diagnostic resolution for isolation efficiency?" or "What if I were willing to give up some detection capability to get a quicker go path?" A prime example of this type of analysis is given by decision tree pruning to achieve built-in tests (BIT) objectives. In spite of the availability of more and better analysis tools, few if any formal methods exist for providing optimal BIT specifications.

In response to this need for optimal BIT specification, at least two approaches to specifying BIT have been developed. Both approaches are based on the assumption that BIT resources must be minimized, and both approaches provide methods for eliminating "unneeded" tests from the BIT specifications follows:

- Test point utilization (TPU): Derived from examining the frequency of test and test point use in a decision tree to determine whether or not to include a specific test in BIT.
- Optimized resolution analysis (ORA): Focuses on several requirements of BIT to provide complete detection and maintain maximum expected ambiguity resolution.

In general, the ORA method provides a more robust answer than TPU, but with increased mathematical complexity. A detailed comparison of two methods is provided in reference 7.

TEST POINT UTILIZATION

To specify BIT for a system, we assume that we have developed a set of candidate tests to be used in BIT. The TPU metric is used to determine which of the tests will be BIT as follows:

1. A diagnostic decision tree is developed with the candidate BIT tests using some optimization procedure (e.g., information gain). The tree may include various cost weights, but the use of weights at this point should be considered with care.

2. Once the decision tree has been generated, the number of times each test is used in an isolation sequence is counted, and the tests are sorted based on these counts. If only n tests can be used in BIT, then the n tests with the highest counts (i.e., the most frequently used tests in the tree) are selected.

The advantage of this approach is its apparent easy implementation. The disadvantage may be that its simplistic approach may yield a less robust answer than ORA.

OPTIMIZED RESOLUTION ANALYSIS

The ORA method proceeds in three steps:

1. Perform an excess test analysis to eliminate tests providing redundant or excess information (i.e., elements redundant and excess tests).
2. Generate a decision tree using all of the remaining BIT candidate tests.
3. Prune the decision tree except for the go path.

In the first step, redundant test is defined to be a test that provides exactly the same information as another single test. An excess test is defined to be a test that provides exactly the same information as a combination of two or more other tests. This is precisely the analysis for minimum test set, previously described.

In completing the second step in the ORA algorithm, generate a decision tree using all of the remaining BIT candidate tests, the tree may be optimized according to whatever cost criteria are appropriate. However, BIT usually has near uniform cost and time. This means that the most often used criterion would be failure rate, thus emphasizing failure probability in deriving the decision tree. In addition, the tree should be generated in such a way that the fewest expected number of tests on the go path are generated. After the decision tree is generated, the number of tests on the go path is determined. If this number exceeds the number allowed, all other tests are eliminated from the tree, and tests are eliminated from the go path starting at the end of the path until the number of tests remaining equals the allowance specified.

If available tests are sufficient to construct a complete go path but insufficient to produce a complete decision tree, then ambiguity must increase as a result of eliminating tests to meet the allowance. In order to minimize expected ambiguity, the ORA algorithm prunes the tree starting at the leaves, excluding the go path from consideration.[3,4] The pruning algorithm maintains minimum expected ambiguity by examining the failure probability at the internal nodes of the trees and pruning the nodes with the lowest probability of failure.

COMPARING DIAGNOSTIC STRATEGIES

With a number of tools now capable of generating optimized strategies, comparisons are possible. The general problem of developing an optimal diagnostic strategy, however, is NP-complete, making global optimization algorithms computationally complex. Indeed, because the existing tools use local, heuristic, or estimating approaches and generate different answers, they invite comparison. Comparison, however, is not easy to do, as seen from the different types of criteria enumerated here. Comparison becomes more reliable and meaningful if a few basic rules are applied.

Rule 1: Tools must be accurate. Accuracy is a requirement, not a basis, for comparing tools. Any loss in accuracy invalidates further comparison because strategies are incorrect and further differences are irrelevant. Improper diagnosis costs dearly in any environment. Approaches applying reasoning under uncertainty further complicate the accuracy issue.

Rule 2: Resolution is more important than efficiency. The system with the tighter fault isolation (smaller ambiguity size) probably has a better inference engine. In fact, for a given system with a fixed set of tests, resolution should be computable and independent of isolation strategy; resolution should be identical for the algorithms compared. The resolution should not only be identical, but the ambiguity groups should be identical. No further comparisons can be made if the resolutions are different. (Again, probabilistic or uncertainty-based approaches complicate this rule.)

Rule 3: Given the same resolution, criteria must be closely similar. An important parameter when developing diagnostic strategies is the size of the test set (directly relatable to development cost). If the size of the test set is not a concern, be sure that test set minimization is not invoked in any of the tools. All other factors should remain constant, such as the need for false alarm tolerance.

Rule 4: Expected values of time, cost, or other factors must be compared over the entire test and conclusion set. There are specific rules that must be followed to have a statistically valid comparison of subsets, and they are usually harder to satisfy than just taking the whole set.

For example, a sample subset must be identically distributed and representative of the whole set (a certain number of distribution moments must be the same).

Rule 5: Differences in efficiency should be evaluated using significance tests, such as a *t*-test. This, of course, requires a sufficient number of comparisons to yield a statistically valid sample. As a rule-of-thumb, however, differences in efficiency of 5% or less would tend to indicate essentially equivalent approaches.

Rule 6: When varying parameters to evaluate performance, only one parameter should be changed at any one time to determine its effects on the outcome. Otherwise, it is difficult (if not impossible) to account for interacting criteria.

SUMMARY

Diagnostic strategies can be developed that incorporate a large number of optimization criteria. Several criteria can be used at the beginning of diagnosis. One criterion is a minimum number of different tests in the tree. Often when developing diagnostic strategies, a single test has substantial cost associated with developing technical orders, documentation, verification, or test program sets. This type of analysis minimizes the tests that are used, but often at the expense of efficiency during optimization. Other preprocessing analyses include developing symptom-based strategies and false alarm tolerant strategies.

A large number of criteria can be used during strategy development. These include minimizing cost and accounting for other criteria such as time not to exceed x, steps not to exceed y, or ambiguity size not to exceed z. The last case may actually trade off ambiguity resolution for efficiency. Applying these criteria is not a straightforward mathematical process.

Poststrategy analyses include decision tree pruning algorithms and may be directly related to developing BIT where a finite capacity exists on board, and we must prune the resulting tree to match that capacity.

Finally, because of the complexity of developing diagnostic strategy based on multiple criteria, comparing tools and methodologies employing these techniques is not straightforward. Several rules based on applying sound empirical analysis. Most important of these rules is that tools must provide equivalent resolution and accurate answers in order to have a valid basis for comparison.

REFERENCES

1. Laurent Hyafil and Ronald L. Rivest, "Constructing Optimal Binary Decision Trees is NP-Complete," *Information Processing Letters*, Vol. 5, No. 1, May 1976, pp. 15-17.

2. R. A. Johnson, "An Information Theory Approach to Diagnosis," *Proceedings of the 6th Annual Conference on Reliability and Quality Control*, 1960, pp. 102-109.

3. C. P. Hartman et al., "Application of Information Theory to the Construction of Efficient Decision Trees," *IEEE Transaction on Information Theory*, IT-28, No.4, July 1982, pp. 565-577.

4. T. R. Addis, "Knowledge Refining for a Diagnostic Aid (An Example of Applied Epidemics)," *The International Journal of Man-Machine Studies*, Vol. 17, 1982, pp. 151-164.

5. W. R. Simpson and J. W. Sheppard, *IEEE Design and Test of Computers*, Series on integrated diagnostics in six parts:

   "System Complexity and Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 8, No. 3, September 1991, pp. 16-30.

   "A Mathematical Model for Integrated Diagnostics, " *IEEE Design and Test of Computers*, Vol. 8, No. 4, December 1991, pp. 25-38.

   "System Testability Assessment for Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 9, No. 1, March 1992, pp. 40-54.

   "Assessing Test Resources Within an Integrated Diagnostic Framework," *IEEE*

Design and Test of Computers, Vol. 9, No. 3, September 1992, pp. 65-78.

"Fault Isolation in an Integrated Diagnostic Environment," *IEEE Design and Test of Computers*, Vol. 10, No. 1, March 1993, pp. 52-66.

"Performing Effective Fault Isolation in Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 10, No. 2, June 1993, pp. 78-90.

6. W. R. Simpson and J. W. Sheppard, "Analysis of False Alarms During System Design," *Proceedings of National Aerospace Electronics Conference*, Dayton, Ohio, May 1992, pp. 657-661.

7. J. W. Sheppard and W. R. Simpson, "A Systems Approach to Specifying Built-In-Tests," *Test Facility Working Group Conference* (TFWGCON '93) Las Vegas, Nevada, June 1993.

8. D. L. Gartner, M. A. Phillips, D. L. Wuerflein, and R. W. White, "Reports on Evaluation of Dependency Modeling and Advanced Diagnosis," *General Dynamics*, Fort Worth, August 1992.