Selection of Minimal Test Points Set for Integer-Coded Fault Wise Table

Shu-Lin Tian, Cheng-Lin Yang, and Bing Long

Abstract—Test points selection for integer-coded fault wise table is a discrete optimization problem. On one hand, traditional exhaustive search method is computationally expensive. On the other hand, the space complexity of traditional exhaustive is low. A tradeoff method between the high time complexity and low space complexity is proposed. At first, a new fault-pair table is constructed based on the integer-coded fault wise table. The fault-pair table consists of two columns: one column represents fault pair and the other represents test points set that can distinguish the corresponding faults. Then, the rows are arranged in ascending order according to the cardinality of corresponding test points set. Thirdly, test points in the top rows are selected one by one until all fault pair are isolated. During the test points selection process, the rows that contain selected test points are deleted and then the dimension of fault-pair table decreases gradually. The proposed test points selection algorithm is illustrated and tested using an integer-coded fault wise table derived from a real analog circuit. Computational results suggest show policies are better than the exhaustive strategy.

Index Terms—Fault-pair table, integer-coded fault wise table, optimization, test points selection.

1. Introduction

Fault dictionary is an important method of simulation-before-test (SBT). Comparing to simulation-after-test (SAT) method[1], this technique is a popular choice[2]. This paper focuses on the selection of test points which is an important phase of fault dictionary technique[3].

Test points set selection problem for analog fault dictionary was studied extensively in many literatures[4][11]. Varghesed[12] proposed a heuristic method to find an optimal set of measurement nodes, using a performance index called “confidence level,” determined by using distance concepts. The concept of ambiguity sets and developed logical rules to select test points was proposed by Hochwald and Bastian[3]. Stenbakken and Souders proposed a method of QR factorization of circuit sensitivity matrix[9]. Lin and Elcherif[2] proposed two heuristic methods based on two criteria proposed by Hochward and Bastian. Spaandonk and Kevenaar[10] looked for a set of test points by combining the decomposition method of system sensitivity matrix and an iterative algorithm. A set of test points, of which the size was equal to the rank of system sensitivity matrix, was selected randomly. Then, in the iterative algorithm, they randomly exchanged a test point in the set with a randomly selected integer in order to compute the determinant of covariance matrix. The new set was accepted if it was lower than the previous set. Prasad and Babu[6][8] proposed four algorithms based on three strategies for inclusive approaches and three strategies for exclusive approaches. Starzyk et al[11] proposed an entropy-based approach. Golonek and Rutkowski[5] proposed a genetic-algorithm-based test points selection method. Based on fault detection table and computing the information content of every node, Pinjala and Kim[7] proposed an approach to get the near-minimum test sets.

The integer-coded fault wise table was first proposed by Lin and Elcherif[2]. This approach was proved to be an effective tool for the optimum test points selection. Subsequent test points selection algorithms proposed by Starzyk[11], Prasad[6][8], Pinjala[7], and Golonek[5] were all based on the integer-coded fault wise table technique.

Methods mentioned above could only find a near minimum test points set. For finding the global minimum set, current exhaustive search method was proved to be NP-hard[11]. For large or medium systems, such as the dictionaries with more than 40 faults and more than 40 test points, the traditional exhaustive search is impractical[11].

This paper deals with the global minimum test points set selection problem. Based on the integer-coded fault wise table, a fault-pair table is constructed. Why and how to construct this table is illustrated in Section 2. Based on fault-pair table, a Boolean algebra method is introduced in Section 3 to select test points set. The time complexity of
proposed method is given in Section 4. Brief conclusions are given in Section 5.

Here, we define the variables in this paper as follows:

\( n_f \): F-th test point.

\( N_f \): Number of candidate test points.

\( S_{opt} \): Desired test points set.

\( f_i \): i-th fault.

\( N_f \): Number of all potential faults (including the nominal case).

\( S_{ij} \): Set of test points. Every test point in this set can isolate fault \( f_i \) and \( f_j \).

### 2. Construction of Fault-Pair Table

Table 1 is a simple integer-coded fault wise table. In this table, the same integer number represents all the faults that belong to the same ambiguity group in a given column. For example, faults \( \{f_1, f_5, f_6\} \) in the second column \( (n_1) \) coded as “2” are not distinguishable and belong to the same ambiguity group. Since each test point represents an independent measurement, ambiguity groups of each test point are independent and can be numbered using the same integers without confusion. Table 1 can also be regarded as an array of 9 elements. If all the elements of this array are different, it means that each fault has a unique integer number and the integer numbers of every two of faults must be different. In the second column \( (n_1) \) of Table 1 for example, \( f_0 \) is coded as “2” and \( f_1 \) is coded as “1”. So fault pair \( (f_0, f_1) \) can be isolated by \( n_1 \). Similarly, faults \( (f_0, f_1) \) can be isolated by \( n_2, n_3 \) and \( n_4 \) separately.

Now that the aim of test points selection is to isolate all fault pairs, and all the \( C^2_9 = \frac{9 \times 8}{2} \) possible fault pairs are listed in Table 2.

There is only one test point \( n_4 \) that can isolate fault pair \( (f_5, f_6) \). It means that test point \( n_4 \) is absolutely necessary. So we should choose \( n_4 \) as the first one. Once \( n_4 \) is selected and \( S_{opt} = \{n_4\} \), all fault pairs that can be isolated by \( n_4 \) which should not be considered in later process. So these fault pairs (and corresponding rows) are deleted from Table 2. For example, from second row to fourth row, \( (f_0, f_5) \) and \( (f_5, f_6) \) should all be deleted. After this process, the dimension of Table 2 decreases dramatically. To facilitate the above process, Table 2 should be arranged in ascending order according to the cardinality of corresponding test points set \( (|S_{ij}|) \). Table 3 shows the result.

| Fault pair | \( S_{ij} \) | \( |S_{ij}| \) |
|------------|-------------|------------|
| \( f_0, f_5 \) | \( n_1 - n_4 \) | 4 |
| \( f_0, f_5 \) | \( n_2 - n_4 \) | 3 |
| \( f_0, f_5 \) | \( n_1 - n_4 \) | 4 |
| \( f_1, f_5 \) | \( n_2 - n_4 \) | 3 |
| \( f_3, f_5 \) | \( n_4 \) | 1 |
| \( f_2, f_5 \) | \( n_3 \) | 2 |
| \( f_3, f_6 \) | \( n_1, n_2, n_4 \) | 3 |
| \( f_0, f_6 \) | \( n_1, n_2, n_4 \) | 3 |
| \( f_0, f_6 \) | \( n_1 - n_4 \) | 4 |
| \( f_0, f_6 \) | \( n_1 - n_4 \) | 4 |

As illustrated above, \( n_4 \) is selected at first. Then, fault pairs (such as \( (f_0, f_2), (f_0, f_3), (f_0, f_1) \) and \( (f_0, f_5) \) etc.) that can be isolated by \( n_4 \) are deleted (the rows that contain \( n_4 \) are deleted) from Table 3. Table 4 is derived in this way.

In above table, the minimal cardinality is two. Because either \( n_1 \) or \( n_3 \) can isolate fault pair \( (f_4, f_5) \), so \( n_1 \) and \( n_3 \) should be taken into consideration separately, \( S_{opt} = \{n_1, n_3\} \) or \( S_{opt} = \{n_4, n_5\} \).

1) When \( n_1 \) is selected, \( S_{opt} = \{n_1, n_3\} \):

Fault pairs in Table 4 that can be isolated by \( n_1 \) are deleted (the rows that contain \( n_1 \) are deleted) and Table 5 is derived.
In Table 5, the only residual fault pair \((f_2, f_3)\) can be isolated by \(n_2\) or \(n_3\), so the final solution is \(S_{opt} = \{n_2, n_1, n_3\}\) or \(S_{opt} = \{n_3, n_1, n_2\}\).

2) When \(n_3\) is selected, \(S_{opt} = \{n_1, n_2\}\)

In Table 4, the rows that contain \(n_3\) are deleted and Table 6 is derived.

In Table 6, the residual fault pair \((f_4, f_8)\) can be isolated by either \(n_1\) or \(n_2\). No matter \(n_1\) or \(n_2\) is selected, all fault pairs are isolated. So the final solution can be \(S_{opt} = \{n_4, n_1, n_2\}\) or \(S_{opt} = \{n_4, n_2, n_1\}\).

Considering above analysis, the final solution can be \(S_{opt} = \{n_1, n_2, n_4\}\), \(S_{opt} = \{n_1, n_3, n_4\}\) or \(S_{opt} = \{n_2, n_3, n_4\}\).

The proposed algorithm can be summarized as follows.

Step 1: Construct fault-pair table.
Step 2: Rearrange the fault-pair table in ascending order according to the value of \(S_{ij}\).
Step 3: Consider \(S_{ij}\) of first row, suppose \(S_{ij} = \{n_{p1}, n_{p2}, \ldots\}\). For each \(n_{p,k}\) in \(S_{ij}\), execute the following process.
Step 4: Delete rows that contain \(n_{p,k}\) and the new fault-pair table is associated with \(n_{p,k}\). For the new fault-pair table, execute Step 3 to Step 4 until an empty fault-pair table is generated (viz. all faults are isolated).

In fact, Step 3 to Step 4 are nested call procedure.

### 3. Abstract and Index Terms

Fig. 1 is a band-pass filter circuit. Totally, there are eighteen potential catastrophic faults \(f_0\) to \(f_{18}\) (including the nominal case) and eleven test points \(n_1\) to \(n_{11}\). Voltage values at all nodes for different faulty conditions are obtained by simulation and the integer-coded fault wise table is constructed by procedures introduced in [6]. The results are shown in Table 7.

According to Step 1 and Step 2, construct fault-pair table and rearrange this table in ascending order according to \(S_{ij}\), and Table 8 is derived. In this table, there is only one test point \(n_{11}\) contained in \(S_{0,14}\). Following Step 4, set \(S_{opt} = \{n_{11}\}\) and the rows that contain \(n_{11}\) are deleted. Then Table 9 is generated.

![Fig. 1. A band pass filter circuit.](image-url)
To delete rows that contain selected test point is with $O\left( \frac{N_F(N_F-1)}{2} \right)$. So the total time complexity of proposed algorithm is

$$O\left( \frac{N_F(N_F-1)}{2} + \frac{N_F(N_F-1)}{2} \log \frac{N_F(N_F-1)}{2} + \frac{N_F(N_F-1)}{2} \right)$$

where $p$ is the number of test points that make up of the final solution.

Considering the extreme condition, $p = N_T$,

$$O\left( \frac{N_F(N_F-1)}{2} + \frac{N_F(N_F-1)}{2} \log \frac{N_F(N_F-1)}{2} + \frac{N_F(N_F-1)}{2} \right) = O\left( \frac{N_T^2N_F^2}{2} \right)$$

$$= O\left( N_TN_F^2 \right), \quad N_T >> \log \frac{N_F}{\sqrt{2}} \gg 1.$$

Obviously, $O\left( N_TN_F^2 \right) << O\left( 2^{N_F}N_F \log N_F \right)$, i.e., the proposed algorithm is more time efficient than the exhaustive method. The space complexity of proposed algorithm is $O\left( \frac{N_F(N_F-1)}{2}N_F \right)$.

5. Conclusions

Modern densely loaded circuit boards have posed problems for fault diagnosis with in-circuit testers because only limited physical access to the boards is allowed. Minimum set of test points selection method is, therefore, badly needs to reduce the number of test point. The exhaustive method is space efficient but time expensive. The proposed algorithm makes use of some feasible memory to accelerate the test points selection process. The proposed method can find the optimum test points set with low time complexity and feasible space complexity. Its efficiency and accuracy is proved by real circuit experiment; therefore, it is a good solution to minimize the size of test points set.

References


