Functional Test-Cost Reduction Based on Fault Tree Analysis and Binary Optimization

Xiaojie Zuo[1,2], Kangcheng Wang*[1,2,3], Yun-Bo Zhao[1,2,3,4], Yu Kang**[1,2,3,4], Peng Bai[3,5]

***Abstract*—With the rapid increase in the complexity of electronics, the cost of the functional testing process used to ensure product functionality continues to increase. Optimization modeling based on reliability analysis is an effective approach to reduce the cost of testing. However, the reliability calculations of existing methods often exhibit significant deviations, making it challenging to guarantee the effectiveness of the resulting strategies in practical applications. To solve this problem, this article proposes an optimization modeling method that integrates statistical analysis and reliability analysis. On one hand, statistical analysis is utilized to analyze the process data and determine the probability of occurrence of the basic event in the fault tree tailored to the object of research. On the other hand, reliability is calculated based on the fault tree structure. On this basis, a binary optimization model is established to determine the testing strategy. The effectiveness of the proposed method is verified through simulation experiments.**

***Index Terms*—Intelligent manufacturing, optimization modeling, fault tree analysis, functional testing, cost reduction.**

I. Introduction

Functional testing is a crucial process in electronics manufacturing for evaluating product quality. With the rapid increase in the complexity of electronics, the cost of functional testing is increasing, and its impact on the overall cost of manufacturing cannot be ignored [1]. To reduce testing costs, production often divides the functional testing session into two sub-sessions: board-level functional testing and system-level functional testing. Board-level testing reduces rework costs by pre-testing some of the main functions of the motherboard. It usually determines the items to be tested or the percentage of the motherboard to be tested according to a certain strategy. On the other hand, system-level testing comprehensively tests all the functions of the finished product to maximize the shipping yield rate [2]. Since the cost of the system-level functional testing session cannot be reduced, the core issue lies in the strategy design of the board-level testing session to minimize testing costs [3].

Existing methods for designing motherboard test strategies are broadly categorized into two types [4]: test ordering and test selection. The former adjusts the test order of the testing items, thereby reducing the test cost by terminating the testing process of faulty motherboards in advance [5]; the latter selects specific testing items from each motherboard for testing, thus reducing the overall testing cost for all motherboards. The latter often outperforms the former in testing high-yield products [6]. As product yield rates continue to improve, test selection is widely used for functional testing of various circuits [7]–[9].

However, existing test selection methods optimize test strategies based solely on the conventional attributes of the testing items, often ignoring the reliability information of the test object. Since the board fabrication process often has frequent variations in working conditions, neglecting reliability information can easily lead to a large deviation between the established optimization model and the actual process. The optimized testing strategy is prone to overfitting in practical applications, making it difficult to effectively reduce the test cost [10].

Enhancing pure optimization modeling by incorporating reliability analysis is an effective approach to address the aforementioned issues. Existing optimization modeling methods based on reliability analysis typically utilize system reliability to establish the objective function or constraints of the optimization problem [11]–[13]. However, the reliability analysis methods in the existing studies have the following issues: Firstly, the current methods often lack the utilization of an effective reliability analysis model tailored to the research object, leading to challenges in ensuring the validity of qualitative reliability analysis and the accuracy of quantitative reliability calculations. Secondly, the reliability of the existing methods is often determined by a limited amount of data collected over a short timeframe, which makes it difficult to accurately reflect the reliability of the system and its components. The aforementioned issues result in a significant deviation between the calculated reliability and the actual value in practical applications of existing methods. This, in turn, impacts the effectiveness of optimization models based on reliability analysis.

To address the aforementioned issues, this article proposes an optimization modeling method that integrates statistical analysis and reliability analysis. It utilizes statistical analysis to analyze process data and determine the probability of the basic event occurrence in the fault tree for the research object. Simultaneously, it employs the fault tree structure to calculate the reliability. On this basis, a binary optimization model is established to obtain a testing strategy with strong generalization ability and reduce the cost of functional testing.

*Corresponding author. E-mail: kcwang@ustc.edu.cn
**Corresponding author. E-mail: kangduyu@ustc.edu.cn
This work was supported by the Key Research and Development Program of Anhui (No. 202104a05020064).
[1]AHU-IAI AI Joint Laboratory, Anhui University, Hefei, 230001, China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China
[3]Department of Automation, University of Science and Technology of China, Hefei, China
[4]Institute of Advanced Technology, University of Science and Technology of China, Hefei, China
[5]LCFC, Hefei, China

The main contents of the rest of this article are as follows: Section II provides a brief description of the functional testing process; Section III proposes an optimization modeling method that integrates statistical analysis and reliability analysis and introduces method evaluation indexes; Section IV carries out simulation experiments to verify the effectiveness of the method; Section V provides a summary of the full article.

II. Background

To ensure that electronics, especially their motherboards, function properly, board-level and system-level functional testing sessions are performed sequentially after the board fabrication process. Both sessions assess several functional items serially.

The board-level functional testing session usually determines the test status or test order of the functional items by developing a testing strategy. If all testing items of a motherboard pass the test, the motherboard is regarded as a qualified product. However, if one item fails the test, the motherboard is regarded as defective and sent to a repair center. The repair center will conduct more detailed tests on the faulty motherboard, determine the underlying causes of the failure based on the testing outcomes and practical experience, repair the relevant components, and record it in the repair log.

Motherboards that pass the board-level functional testing will be assembled into a finished product. All components of the finished product will then be tested in the system-level functional testing session. If all items of the product pass the test, the product will be considered qualified. If not, the product is sent to the repair center for disassembly, additional testing, and repair.

If a motherboard has untested faulty components, it is considered a false positive motherboard, meaning it is incorrectly classified as a functional one. The motherboard will then be assembled into a finished product. During the system-level functional testing session, the finished product will be detected as defective and sent for repair.

The testing outcomes are binary data, where '0' indicates that the motherboard or finished product failed the test on that item, and '1' indicates that it passed the test. Let n and m denote the number of motherboards and the number of tested items, respectively, the testing outcomes of the board-level functional testing session can be represented as a data set $D \in \{0,1\}^{n\times m}$. If there are untested functional items in the board-level functional testing session, the values of the corresponding elements of D are not immediately available. However, since all functional testing items will be tested in the system-level functional testing session to ensure product quality, the testing outcomes from the system-level session can be used to determine the values of all elements in the data set D.

The cost of functional testing mainly consists of the time cost of testing items and the cost of repairing defective products. Except for the time cost of the system-level functional testing session, the time cost, as well as the rework cost of the false positive boards and final products, are all determined by the board-level testing strategy. Therefore, designing an efficient board-level functional testing strategy is the key to reducing the overall cost of functional testing.

III. Method

A. Reliability Analysis of Functional Testing

The cost of reworking false positive motherboards accounts for a relatively high percentage of the overall cost of testing. The cost is essentially determined by the reliability of the board-level functional testing session with respect to the system-level functional testing session. Therefore, this paper starts with analyzing the reliability and designing an effective method to reduce the overall cost of testing.

When the test selection strategy is used, the board-level functional testing items can be categorized into two types according to the strategy: testing items and untested items. Since the testing items are tested in series, the testing items form a serial subsystem, while the untested items form another serial subsystem. Assuming that testing the faulty components of a defective motherboard will certainly identify it as faulty, then the presence of a false positive motherboard is caused by the untested items. Therefore, for the system-level functional testing, the reliability of the subsystem consisting of testing items from the board-level functional testing session is 100%, while the unreliability arises solely from the subsystem comprising untested items. Therefore, the reliability of the board-level test session with respect to the system-level test session can be calculated as follows:

$$R(\boldsymbol{S}) = \prod_{j=1}^{m} R_j|(S_j = 0) \tag{1}$$

where R_j denotes the reliability of the motherboard component corresponding to the jth testing item.

To determine the reliability of a motherboard component, it is necessary to analyze the inherent causes of its failure. Fault tree analysis is an effective tool for analyzing the possible causes of failures of a system and its components and their probability of occurrence. In our previous work [14], we used this tool to build a fault tree tailored to the functional testing session of a laptop motherboard. In this fault tree, the top event is the motherboard failure, the first-level intermediate events are the failures of the motherboard components, the basic events are the underlying causes of the motherboard component failures, and the basic events are independent of each other. For the sake of brevity, the relevant diagrams and tables will not be repeated here; interested readers should refer to Ref [14].

In this article, we still take this laptop computer as an example to study the functional testing process and analyze the reliability of its motherboard using the established fault tree. The reliability $R(\boldsymbol{S})$ of the top event in Eq. (1) is derived according to the structure of the fault tree as follows:

First, a large number of historical testing outcomes of this motherboard are constructed as a dataset $D^h \in \{0,1\}^{n^h\times m}$, where n^h denotes the number of motherboards that have undergone the board-level functional testing session. Then, a repair log corresponding to D^h for the motherboard and its

finished product is obtained from the repair center. This log includes diagnostic and repair information for the faulty motherboard in both the board-level and system-level functional testing sessions. Next, another dataset is constructed using the repair logs and denoted as $D^r \in \{0,1\}^{n^r \times q}$, where q denotes the number of basic events. In this dataset, $D^r_{s,k} = 1$ denotes that the sth motherboard is repaired due to the occurrence of the kth basic event. D^r is a sparse matrix, and typically each row of D^r has only one element equal to 1. Use the following formula to calculate the number of occurrences of the kth basic event:

$$n^r_k = \sum_{s=1}^{n^r} (D^r_{s,k} = 1), k = 1, \cdots, q \tag{2}$$

Then the reliability of the kth basic event can be estimated as follows:

$$R(X_k) = \frac{n^h - n^r_k}{n^h}, k = 1, \cdots, q \tag{3}$$

Since the number of faulty motherboards is very limited in the high-yield board fabrication process, to improve the accuracy of the basic event reliability estimation, a large number of tests are required to collect enough repair logs. Therefore, the value of n^h in Eq. (3) should typically be no less than 1e7.

Next, based on reliability theory, the reliability of the components corresponding to the first-level intermediate events is calculated as follows:

$$R(M_j) = \prod_{X_k \in B_j} R(X_k), j = 1, \cdots, m \tag{4}$$

Since all first-level intermediate events correspond to board-level functional testing items, the reliability of each testing item can be obtained as follows:

$$R_j = R(M_j), j = 1, \cdots, m \tag{5}$$

Finally, the reliability of the top event is calculated using Eq. (1) and Eqs. (3)–(5).

B. Optimization Modeling Integrating Statistical Analysis and Reliability Analysis

Based on the previous analysis, it can be seen that increasing the reliability of the board-level functional testing session with respect to the system-level functional testing session may lead to an increase in false positive motherboards and a decrease in testing time. Considering that the increased rework cost due to an increase in the number of false positive motherboards tends to be higher than the cost savings from a decrease in testing time, maximizing the reliability of the board-level functional testing session with respect to the system-level functional testing session is a potentially effective strategy for reducing the overall cost of testing. Thus, the objective of the optimization problem in this section is expressed in the following form:

$$\max_{\boldsymbol{S}} R(\boldsymbol{S}) \tag{6}$$

The testing time constraint is imposed on the optimization problem to avoid the accumulation of motherboards. To mitigate overfitting, the average testing time for each testing item is obtained from the historical testing time $\boldsymbol{T}^h \in \mathbb{R}^{n^h \times m}$ by the following equation:

$$\bar{t}^h_j = \frac{1}{n^h} \sum_{i=1}^{n^h} T^h_{i,j}, j = 1, \cdots, m \tag{7}$$

where $T^h_{i,j}$ denotes the time cost of testing the jth item on the ith historical motherboard.

Let t^0 be the testing time threshold for each motherboard, which is determined by the motherboard fabrication schedule. The time constraint for the test can be expressed as:

$$\sum_{j=1}^{m} \bar{t}^h_j | (S_j = 1) \leq t^0 \tag{8}$$

Based on Eq. (6) and Eq. (8), a binary optimization problem is formulated as follows:

$$\max_{\boldsymbol{S}} R(\boldsymbol{S}) \tag{9}$$

$$\text{s.t.} \sum_{j=1}^{m} \bar{t}^h_j | (S_j = 1) \leq t^0 \tag{10}$$

$$\boldsymbol{S} \in \{0,1\}^m \tag{11}$$

The optimal testing strategy can be obtained by solving the optimization problem (9)–(11).

C. Evaluation Criteria

This subsection introduces three criteria for evaluating the performance of the selected testing strategy, namely false positive rate (FPR), average testing time cost, and average cost of testing.

FPR denotes the ratio of faulty motherboards not detected by the testing strategy $\boldsymbol{S}$ to the total number of actual faulty motherboards. Based on Section **??**, The detailed steps for calculating FPR are as follows:

The actual number of qualified motherboards can be obtained by logically analyzing the testing outcomes:

$$n_P = \sum_{i=1}^{n} \left[\left(\bigwedge_{j=1}^{m} (D_{i,j} = 1) \right) \equiv \top \right] \tag{12}$$

where $\top$ denotes the logical truth.

The actual number of defective motherboards can be obtained by subtracting the number of qualified motherboards from the total number of motherboards.

$$n_N = n - n_P \tag{13}$$

By analyzing the logical relationship between the testing outcomes, the number of faulty motherboards detected using the testing strategy $\boldsymbol{S}$ can be calculated as follows:

$$n_{TN}(\boldsymbol{S}) = \sum_{i=1}^{n} \left[\left(\bigvee_{j=1}^{m} (D_{i,j} = 0) | (S_j = 1) \right) \equiv \top \right] \tag{14}$$

The number of false positive motherboards is calculated as follows using Eq. (13–14):

$$n_{FP}(\boldsymbol{S}) = n_N - n_{TN}(\boldsymbol{S}) \tag{15}$$

Then the FPR can be calculated as follows:

$$\text{FPR}(\boldsymbol{S}) = \frac{n_{FP}(\boldsymbol{S})}{n_N} \tag{16}$$

The cost of repairing a false positive motherboard in the system-level functional testing session is determined by $n_{FP}(\boldsymbol{S})$ and the cost of repairing each motherboard in the finished product. The cost of rework depends on a variety of factors and varies for each motherboard. For simplicity, assuming that the repairing cost is the same for each motherboard and converted to repair time, the rework cost of the process can then be simplified as a constant and denoted as $c_{r,FP}$. Similarly, the rework cost of a faulty motherboard detected by the board-level functional testing session is determined by $n_{TN}(\boldsymbol{S})$ and the rework cost per motherboard, assuming that the rework cost per motherboard for this session is $c_{r,TN}$, which is usually significantly lower than $c_{r,FP}$. The exact values of both $c_{r,FP}$ and $c_{r,TN}$ are determined by practical experience.

The average testing time per motherboard is another important criterion for evaluating the effectiveness of the testing strategy. This cost represents the test speed, which can be calculated as follows:

$$c_t(\boldsymbol{S}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} T_{i,j}|(S_j = 1) \tag{17}$$

Assume that the total cost of functional testing is the sum of testing time cost and rework cost. Then the testing time cost and rework cost can be integrated into a comprehensive criterion based on the number of motherboards tested and repaired. This criterion is defined as the average cost of testing per motherboard and can be expressed as follows:

$$c(\boldsymbol{S}) = \frac{nc_t(\boldsymbol{S}) + c_{r,TN}n_{TN}(\boldsymbol{S}) + c_{r,FP}n_{FP}(\boldsymbol{S})}{n} \tag{18}$$

The testing strategy with the lowest $c(\boldsymbol{S})$ value is the one that can effectively balance test quality and test speed.

IV. Case Study

This section validates the effectiveness of the proposed test cost reduction method using the functional testing process of a laptop motherboard mentioned in Section III.

A. Data Preparing

To protect the trade secrets of the electronics manufacturer, the following method is used to generate simulation data that is similar to the actual data:

The number of board-level functional testing items is set to 16, and the motherboard components tested are the same as those listed in Ref [14]. According to the actual value of the yield rate of each testing item, the yield rate of each testing item is set as a random number uniformly distributed on the interval [0.999, 1]. The number of motherboards n^h in the historical test log is set to 2e7 to capture enough repair logs. Historical functional testing outcomes are randomly generated based on the yield rates of test items and form a matrix $D^h \in \{0,1\}^{2e7\times 16}$. From D^h, the number of faulty motherboards detected in the historical functional testing n^r is 135,048. The repair logs of the faulty motherboards form a matrix $D^r \in \{0,1\}^{135{,}048\times 24}$, where the elements of each column of the matrix are randomly generated based on the probability of occurrence of the corresponding basic event. Set the number of motherboards currently under test, n, to 5e4. Generate the current functional testing outcomes randomly according to the yield rate and form a matrix $D \in \{0,1\}^{5e4\times 16}$.

According to the actual testing procedure, the mean testing time for each item $\bar{t}_j$ is set as a random number uniformly distributed between 0.1 and 11 seconds, with the standard deviation of the testing time set as $\bar{t}_j/100$. Two datasets containing the testing time of the motherboard are generated based on the mean and standard deviation mentioned above. The first dataset corresponds to D^h and is denoted as $\boldsymbol{T}^h \in \mathbb{R}^{2e7\times 16}$; the second dataset corresponds to D and is denoted as $\boldsymbol{T} \in \mathbb{R}^{5e4\times 16}$. Based on the current progress of the board fabrication process, the testing time threshold for each motherboard is set to 75 seconds. In addition, based on the actual repairing cost, the repairing costs $c_{r,TN}$ and $c_{r,FP}$ are set at 0.5 hours and 5 hours, respectively.

The first 60% of the dataset D is used as a training set for model training, while the remaining 40% is used as a test set to validate the effectiveness of the testing strategy. Table I presents the yield rates of the testing items on the training and test sets, along with the relative changes of the yield rates in the test set with respect to the training set (denoted as Δ). Bold values highlight significant yield variations in the corresponding testing items.

TABLE I
VALUES AND RELATIVE CHANGES OF YIELD RATES ON THE TRAINING AND TEST SETS

No. of item	Training set's yield	Testing set's yield	Δ
1	0.99923	0.99860	**-0.0634%**
2	0.99957	0.99920	**-0.0367%**
3	0.99963	0.99925	**-0.0383%**
4	0.99993	1.00000	0.0067%
5	0.99943	0.99945	0.0017%
6	0.99980	0.99960	-0.0200%
7	0.99993	0.99990	-0.0033%
8	0.99957	0.99945	-0.0117%
9	1.00000	0.99930	**-0.0700%**
10	0.99943	0.99925	-0.0183%
11	0.99947	0.99890	**-0.0567%**
12	0.99953	0.99930	-0.0233%
13	0.99947	0.99925	-0.0217%
14	0.99980	1.00000	0.0200%
15	0.99943	0.99950	0.0067%
16	0.99943	0.99920	-0.0233%

B. Ablation Study

To fully verify the effectiveness of the proposed method, two sets of ablation experiments are designed as follows:

1) Modify the way of obtaining reliability in the objective function (9): use the testing outcomes of the training set to calculate the yield rate of each testing item, as an approximation of the reliability of the corresponding component. Solve the modified optimization problem

and denote the resulting optimal testing strategy as $\boldsymbol{S}_{\text{approx}R}$.

2) Modify the objective function (9) to minimize the average cost of testing on the training set $c_{train}(\boldsymbol{S})$. Solve the modified optimization problem and denote the resulting optimal testing strategy as $\boldsymbol{S}_{\min c}$.

The first ablation experiment aims to verify the effectiveness of the strategy obtained when the reliability is inaccurately calculated using the reliability analysis model tailored to the research object. The second ablation experiment aims to verify the effectiveness of the strategy obtained when the reliability information of the test object is not effectively utilized. For simplicity, the methods corresponding to the two comparative experiments are referred to as the "reliability approximation method" and the "cost minimization method", respectively. In addition, the strategy obtained by the proposed method is denoted as $\boldsymbol{S}_{\text{ours}}$.

This simulation case study was carried out on a Windows 10 PC equipped with an Intel Core 2.5GHz CPU and 32GB of RAM using MATLAB R2023a. Monte Carlo simulation was used to solve each optimization problem. The number of simulation iterations for each round of optimization is set to 5000. The running time of the proposed method and the methods in the ablation study are shown in Table II.

TABLE II
RUNNING TIME OF DIFFERENT METHODS

Methods	Running time/sec
Proposed method	**0.0031**
Reliability approximation method	0.0034
Cost minimization method	6.8910

The test strategies obtained from each method are shown in Table III; the values of the evaluation criterion of each method, together with the relative changes of the methods in the ablation study with respect to the proposed method on the training and test sets are shown in Table IV and Table V, respectively.

C. Result Analysis

From Table I, it can be seen that the working conditions of the test set have changed compared to the training set, resulting in relatively large variations in the yield rates of several testing items.

From Table IV and Table V, it can be seen that the strategy $\boldsymbol{S}_{\text{ours}}$ is similar to that of $\boldsymbol{S}_{\text{approx }R}$ in the training set. However, in the test set, the values of FPR and the average cost of testing for strategy $\boldsymbol{S}_{\text{ours}}$ are significantly better than those of strategy $\boldsymbol{S}_{\text{approx}R}$. This suggests that the improvement of computational accuracy of the reliability helps to obtain a more reliable test strategy, which improves the generalization performance of the strategy when working conditions change. It can also be seen from Table III that the strategy $\boldsymbol{S}_{\text{approx}R}$ has only one testing item (the No. 9) with a different test state compared to $\boldsymbol{S}_{\text{ours}}$. Based on the analysis in Section III-A, it can be seen that this issue leads to a decrease in the reliability of the board-level functional testing session, resulting in the tendency of performance degradation on the test set.

It can also be seen from Table IV and Table V that the average cost of testing for the strategy $\boldsymbol{S}_{\min c}$ is the lowest among the three strategies on the training set. This is because the goal of the cost minimization method is to minimize the average cost of testing on the training set. However, on the test set, the average cost of testing for this strategy is higher than that of the other two strategies. It can be seen that the cost minimization approach suffers from more significant overfitting when the working conditions change, as it does not utilize the inherent reliability information in the historical testing outcomes. It can also be seen from Table III that the number of functional items tested by $\boldsymbol{S}_{\min c}$ is fewer than that of the other strategies. This leads to a significantly lower average testing time than that of the other strategies. However, it also results in a significantly higher value of the FPR criterion than the other strategies, resulting in a significantly greater number of false positive motherboards and causing higher rework costs and total cost of testing.

In addition, according to Table II, the running time of the proposed method is the lowest among the three methods. It is close to that of the reliability approximation method. Since these two methods do not require frequent computation of cost function values, their running time is significantly lower than that of the cost minimization method. In practical applications, a lower running time helps in making timely adjustments to the test strategy, thereby further reducing the test cost.

Based on the above analysis, it can be seen that the proposed method, which integrates statistical analysis and reliability analysis, can select test strategies with strong generalization ability, perform best on the test set, and acquire test strategies with the highest efficiency. This approach has the potential to significantly reduce the cost of functional testing in practical applications.

V. CONCLUSION

This article proposes an optimization modeling method that integrates statistical analysis and reliability analysis. Statistical analysis is used to analyze the process data and determine the probability of occurrence of the basic event occurrence in the fault tree tailored to the object of research. This information is then used to calculate the reliability of the fault tree structure. On this basis, the testing strategy is determined by establishing a binary optimization model. The proposed method has been verified to be effective in reducing the cost of functional testing by over 10% using simulation data similar to the actual data.

REFERENCES

[1] M. Liu, R. Pan, F. Ye, X. Li, K. Chakrabarty, and X. Gu, "Fine-grained adaptive testing based on quality prediction," *ACM Transactions on Design Automation of Electronic Systems*, vol. 25, no. 5, pp. 1–25, jul 2020.

[2] I. Polian, J. Anders, S. Becker, P. Bernardi, K. Chakrabarty, N. El-Hamawy, M. Sauer, A. Singh, M. S. Reorda, and S. Wagner, "Exploring the mysteries of system-level test," in *2020 IEEE 29th Asian Test Symposium (ATS)*. IEEE, nov 2020.

[3] S. Biswas, H. Wang, and R. D. S. Blanton, "Reducing test cost of integrated, heterogeneous systems using pass-fail test data analysis," *ACM Transactions on Design Automation of Electronic Systems*, vol. 19, no. 2, pp. 1–23, mar 2014.

TABLE III
BOARD-LEVEL FUNCTIONAL TEST STRATEGIES OBTAINED BY DIFFERENT METHODS

Testing strategy	Number of functional testing items																Total number of testing items
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
$\boldsymbol{S}_{\text{ours}}$	1	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	13
$\boldsymbol{S}_{\text{approx}R}$	1	0	1	0	1	1	0	1	0	1	1	1	1	1	1	1	12
$\boldsymbol{S}_{\min c}$	1	0	1	0	1	0	0	1	0	1	1	0	1	1	1	1	10

TABLE IV
EVALUATION CRITERIA OF DIFFERENT TEST STRATEGIES ON THE TRAINING SET

Testing strategy	Average testing time/sec		FPR		Average cost of testing/sec	
	Value	Relative change	Value	Relative change	Value	Relative change
$\boldsymbol{S}_{\text{ours}}$	74.9424	\	**0.0847**	\	94.9224	\
$\boldsymbol{S}_{\text{approx}R}$	72.7853	-2.88%	**0.0847**	0.00%	92.7653	-2.27%
$\boldsymbol{S}_{\min c}$	**48.2619**	-35.60%	0.2804	19.58%	**88.2219**	-7.06%

TABLE V
EVALUATION CRITERIA OF DIFFERENT TEST STRATEGIES ON THE TEST SET

Testing strategy	Average testing time/sec		FPR		Average cost of testing/sec	
	Value	Relative change	Value	Relative change	Value	Relative change
$\boldsymbol{S}_{\text{ours}}$	74.9406	\	**0.0914**	\	**107.2506**	\
$\boldsymbol{S}_{\text{approx}R}$	72.7834	-2.88%	0.1624	7.11%	116.4334	8.56%
$\boldsymbol{S}_{\min c}$	**48.2603**	-35.60%	0.3299	23.86%	118.6403	10.62%

[4] M. Chen and A. Orailoglu, "Test cost minimization through adaptive test development," in *2008 IEEE International Conference on Computer Design*. IEEE, oct 2008.

[5] W. Jiang and B. Vinnakota, "Defect-oriented test scheduling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 3, pp. 427–438, jun 2001.

[6] R. Pan, Z. Zhang, X. Li, K. Chakrabarty, and X. Gu, "Black-box test-cost reduction based on bayesian network models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 2, pp. 386–399, 2020.

[7] K. Huang, J. Wen, and J. Willmore, "Test-suite-based analog/RF test time reduction using canonical correlation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 12, pp. 2143–2147, 2016.

[8] M. Pradhan, B. B. Bhattacharya, K. Chakrabarty, and B. B. Bhattacharya, "Predicting x-sensitivity of circuit-inputs on test-coverage: A machine-learning approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 12, pp. 2343–2356, dec 2019.

[9] Y. Li, E. Yilmaz, P. Sarson, and S. Ozev, "Adaptive test for RF/analog circuit using higher order correlations among measurements," *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 4, pp. 1–16, jun 2019.

[10] E. Yilmaz, S. Ozev, O. Sinanoglu, and P. Maxwell, "Adaptive testing: Conquering process variations," in *2012 17TH IEEE EUROPEAN TEST SYMPOSIUM (ETS)*. IEEE, May 2012.

[11] S. Eryilmaz, "Reliability analysis of multi-state system with three-state components and its application to wind energy," *Reliability Engineering & System Safety*, vol. 172, pp. 58–63, Apr. 2018.

[12] H.-P. Chen and M. B. Mehrabani, "Reliability analysis and optimum maintenance of coastal flood defences using probabilistic deterioration modelling," *Reliability Engineering & System Safety*, vol. 185, pp. 163–174, May 2019.

[13] P. Ni, J. Li, H. Hao, W. Yan, X. Du, and H. Zhou, "Reliability analysis and design optimization of nonlinear structures," *Reliability Engineering & System Safety*, vol. 198, p. 106860, Jun. 2020.

[14] Y. Li, K. Wang, Y. Kang, Y. Zhao, and P. Bai, "Board-level functional test selection based on fault tree analysis," in *2023 6th International Symposium on Autonomous Systems (ISAS)*. IEEE, Jun. 2023.